UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALEXANDRE GREFF BUAES

A Low Cost One-camera Optical Tracking System for Indoor Wide-area Augmented and Virtual Reality Environments

Porto Alegre, February 2006

ALEXANDRE GREFF BUAES

A Low Cost One-camera Optical Tracking System for Indoor Wide-area Augmented and Virtual Reality Environments

Dissertation presented to the **Post-Graduation Program in Electrical Engineering** of the Federal University of *Rio Grande do Sul* as part of requirements to obtain the Master Degree in Electrical Engineering.

Field of investigation: Automation and Electro-electronical Instrumentation

ADVISER: Prof. Dr. Carlos Eduardo Pereira

Porto Alegre, February 2006

ALEXANDRE GREFF BUAES

A Low Cost One-camera Optical Tracking System for Indoor Wide-area Augmented and Virtual Reality Environments

(Sistema de Rastreamento Ótico Monocular de Baixo Custo para Ambientes Internos Amplos de Realidade Virtual e Aumentada)

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela Universidade de Stuttgart, Alemanha.

Banca Examinadora:

Prof. Dr. Clésio Luis Tozzi, UNICAMP

Doutor pela UNICAMP - Campinas, Brasil

Prof. Dr. Altamiro Amadeu Susin, UFRGS

Doutor pela INPG - Grenoble, França

Prof. Dr. Walter Fetter Lages, UFRGS

Doutor pelo ITA – São José dos Campos, Brasil

Coordenador do PPGEE: _____

Prof. Dr. Carlos Eduardo Pereira

Porto Alegre, Fevereiro de 2006

ACKNOWLEDGEMENTS

I would like to thank Dr. Prof. Carlos Eduardo Pereira, my adviser, for his full support and guidance, especially for intermediating negotiations for my stay in Darmstadt at Fraunhofer IGD.

I would like to thank Dr. Andre Stork, Head of Department Industrial Applications at Fraunhofer IGD for the opportunity of developing most of this work under his supervision.

I would like to thank Gino Brunetti for his support and help during my stay at Fraunhofer IGD and, particularly, I thank Pedro Santos for the unmeasurable amount of technical experience I learned when working with him, as well as for the complete support and help during our teamwork experience by Fraunhofer IGD.

I am also grateful to SENAI-CETA/RS and CAPES for the financial support during my post-graduation study period.

I would like to thank my girlfriend Daniela, especially for her support and understanding during the development of this work.

Last but not least, I would like to thank my family, especially my parents, for their dedication and complete support for my activities, not only during the development of this work but in every important moment of my life.

Alexandre Greff Buaes

Porto Alegre, February 2006

RESUMO

O número de aplicações industriais para ambientes de "Realidade Aumentada" (AR) e "Realidade Virtual" (VR) tem crescido de forma significativa nos últimos anos. Sistemas óticos de rastreamento (*optical tracking systems*) constituem um importante componente dos ambientes de AR/VR. Este trabalho propõe um sistema ótico de rastreamento de baixo custo e com características adequadas para uso profissional. O sistema opera na região espectral do infravermelho para trabalhar com ruído ótico reduzido. Uma câmera de alta velocidade, equipada com filtro para bloqueio da luz visível e com *flash* infravermelho, transfere imagens de escala de cinza não comprimidas para um *PC* usual, onde um *software* de pré-processamento de imagens e o algoritmo *PTrack* de rastreamento reconhecem um conjunto de marcadores retro-refletivos e extraem a sua posição e orientação em 3D. É feita neste trabalho uma pesquisa abrangente sobre algoritmos de pré-processamento de imagens e de rastreamento. Uma bancada de testes foi construída para a realização de testes de acurácia e precisão. Os resultados mostram que o sistema atinge níveis de exatidão levemente piores, mas ainda comparáveis aos de sistemas profissionais. Devido à sua modularidade, o sistema pode ser expandido através do uso de vários módulos monoculares de rastreamento interligados por um algoritmo de fusão de sensores, de modo a atingir um maior alcance operacional. Uma configuração com dois módulos foi montada e testada, tendo alcançado um desempenho semelhante à configuração de um só módulo.

Palavras-chave: Realidade Aumentada. Realidade Virtual. Sistemas Óticos de Rastreamento. Baixo Custo. Fusão de Sensores.

ABSTRACT

In the last years the number of industrial applications for Augmented Reality (AR) and Virtual Reality (VR) environments has significantly increased. Optical tracking systems are an important component of AR/VR environments. In this work, a low cost optical tracking system with adequate attributes for professional use is proposed. The system works in infrared spectral region to reduce optical noise. A high-speed camera, equipped with daylight blocking filter and infrared flash strobes, transfers uncompressed grayscale images to a regular PC, where image pre-processing software and the PTrack tracking algorithm recognize a set of retro-reflective markers and extract its 3D position and orientation. Included in this work is a comprehensive research on image pre-processing and tracking algorithms. A testbed was built to perform accuracy and precision tests. Results show that the system reaches accuracy and precision levels slightly worse than but still comparable to professional systems. Due to its modularity, the system can be expanded by using several one-camera tracking modules linked by a sensor fusion algorithm, in order to obtain a larger working range. A setup with two modules was built and tested, resulting in performance similar to the stand-alone configuration.

Keywords: Augmented Reality. Virtual Reality. Optical Tracking Systems. Low Cost. Sensor Fusion.

TABLE OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

INDEX OF ABBREVIATIONS

1394 Firewire	FireWire interface (also known as IEEE 1394 or i.Link)
2D	Two Dimensional in XY
3D	Three Dimensional in XYZ
AC	Alternate Current
ANN	Artificial Neural Network
API	Application Programming Interface
AR	Augmented Reality
ART	Advanced Real Time Tracking GmbH. – Tracking System Manufacturer
BBN	Bayesian Belief Network
CAD/CAM	Computer-aided Design / Computer-aided Manufacturing
CCD	Charge-Coupled Device
CMOS	Complementary Metal-oxide Semiconductor
COTS	Commercial-Off-The-Shelf
CRT	Cathode Ray Tube
DC	Direct Current
DFS	Depth-First Search
DGPS	Differential GPS
DLT	Direct Linear Transformation
DoF	Degrees of Freedom
DUI	Device Unified Interface
DWARF	Distributed Wearable Augmented Reality Framework
EMI	Electromagnetic Interference
EVaRT	EVa Real-Time Software
FoV	Field of View
FPGA	Field Programmable Gate Array
Fraunhofer IGD	*Fraunhofer Institut für graphische Datenverarbeitung* (Fraunhofer Institute for Computer Vision), Darmstadt, Germany
GPS	Global Positioning System
HMD	Head Mounted Display

ID	Identification
IDS	Imaging Development Systems GmbH. – Camera Manufacturer
IP	Internet Protocol
IR	Infrared
LED	Light Emitter Diode
LEPD	Lateral Effect Photodiodes
ML	Machine Learning
MR	Minimal Reality
OpenGL	Open Graphics Library
PAL	Phase Alternating Line (TV standard)
PC	Personal Computer
PDA	Personal Digital Assistant
PSS	Personal Space Station
RAC	Radial Alignment Constraint
RAM	Random Access Memory
RANSAC	Random Sample Consensus
RGB	Red Green Blue color definition system
RMS	Root Mean Square
RTK-DGPS	Real Time Kinematics DGPS
SCAAT	Single Constraint At A Time
SENAI-CETA/RS	*Centro de Excelência em Tecnologias Avançadas* (Center of Excellence in Advanced Technologies) within SENAI/RS, Porto Alegre, Brazil
SI	*Système International d'unités* – International System of Units
SVD	Singular Value Decomposition
TDM	Time Domain Multiplex
TOF	Time of Flight
UDP	User Datagram Protocol
USB 2.0	Universal Serial Bus, specification version 2.0
UUV	Unmanned Undersea Vehicles
VR	Virtual Reality
WLAN	Wireless Local Area Network
XML	Extensible Markup Language
YIQ	Luminance-Inphase-Quadrature color model

1 INTRODUCTION

In Virtual Reality (VR) applications, the complete environment is computer-generated. In Augmented Reality (AR), real and virtual objects are combined in a real environment. AR systems must run interactively and in real time, because real and virtual objects must be aligned with each other, according to (AZUMA, 1997) and (AZUMA, 2001). AR/VR environments have a variety of possible applications, for example in product design and simulation, factory planning and medical procedures.

A fundamental component in AR/VR environments are tracking systems, which allow users to interact with the mixed or virtual world. Tracking consists in estimating the pose (position and orientation) of objects or artifacts, allowing seamless integration of real and virtual content.

1.1 GOALS AND MOTIVATION

Nowadays, tracking systems with enough precision and accuracy for use in professional Augmented Reality (AR) and Virtual Reality (VR) environments are usually commercial solutions very precise on one hand but very expensive on the other hand. Systems with intermediate attributes, i.e., not so expensive but with reasonable precision and accuracy, are a research field yet to be better explored. This is the main motivation for this work. Many applications, which do not demand very precise systems, could have their requirements fulfilled by a system with intermediate performance.

The goal for this work is the implementation of a new low cost optical tracking system, whose performance allows it to be compared to and even to replace some commercial systems, in applications where performance can be slightly worse.

Optical tracking systems represent the best trade-off between requirements of an ideal tracking system and actual properties, composing the best suitable set of attributes among all existing tracking technologies. Marker-based optical tracking reduces latency time, due to the simplification of image pre-processing tasks, and increases robustness, because the probability of artifact detection is higher. The use of one-camera tracking topology allows reduction of costs, since only one camera is needed instead of two, and is a research topic not so exhaustively explored as stereo topology. The use of one-camera tracking was also chosen in order to continue ongoing research work initiated at Fraunhofer Institute for Computer Vision (Fraunhofer IGD).

Based on the aforementioned reasons, the new system to be developed will be optical, marker-based and have one-camera topology. Also a multiple-camera tracking configuration will be developed and evaluated, whose goal is to increase working volume. Specifications of the new system are explained in more detail in section 2.8.

1.2 BRIEF DESCRIPTION OF RESULTS

This thesis presents a comprehensive survey on existing commercial and research tracking systems, followed by a comparison among these systems. For the implementation of a new optical tracking system, an extensive description of concepts and algorithms for image pre-processing, pose estimation and calibration techniques is presented.

In the scope of this work the author conducted a research on suitable filter and cameras, followed by components selection for the new system. Regarding software modules development as described in chapter 4, the image pre-processing software module was entirely implemented within this work, as well as part of the hardware interface module and adaptions in the PTrack module. Also a complete testbed for system evaluation was built and widely used, including calibration patterns.

Experimental results showed that the new system, called PTrack on IDS (Imaging Development Systems GmbH.) Cameras, performed slightly worse than PTrack running on its original hardware, ART (Advanced Real Time Tracking GmbH.) Cameras. However, results are better than expected from the comparison, since some experiments showed very similar results. In comparison with ARToolKit running on the same camera, PTrack performed between 2 and 3 times better what relates to translation and rotation exactness, in average. However, PTrack revealed much higher performance sensitivity to the number of labels tracked in the scene. The system running in multiple-camera configuration performed slightly worse than in stand-alone setup, except for the translational accuracy, which reached 8.3 mm, somewhat better than 10.4 mm obtained when using a single tracking module.

The score for each attribute of the new tracking system, running in stand-alone configuration, as well as the overall grade, were obtained using the grading method defined in section 2.8. The only specification considered as not met by this version of PTrack, running on the implemented hardware, was translational accuracy, which performed slightly worse than the acceptable range. All other attributes were met. Working range and costs were assigned the rank “Very Good”, standing out as best attributes of the system.

The overall grade of the new implemented system was “Good”, indicating that it achieved results very close to the best performance specified in section 2.8.

In summary, the new system implemented in this work is an interesting contribution to the research field, since it is a modular low cost system with reasonable attributes and sufficient performance for many AR/VR applications. Modularity allows the system to be expanded by use of multiple one-camera tracking modules, resulting in enlarged working range.

1.3 ORGANIZATION OF THIS DOCUMENT

This thesis is organized as follows.

In chapter 2, basic concepts about tracking are presented. Typical application scenarios for tracking systems are described. The available tracking technologies are analyzed and compared, and the reasons for choosing optical tracking

for the new system are presented. Besides, the main existing commercial and research optical tracking systems are described and compared. Based on analysis of comparison's results, specifications for the new system are presented.

In chapter 3, the fundamental concepts and tools related to the new implemented system are analyzed. This chapter contains more information than needed for specifically understanding the implemented solution, providing a basis for comprehension of roughly any optical tracking solution.

In chapter 4, the new tracking system is described. First, the one-camera tracking module is detailed, followed by descriptions of large area tracking topology and manner of operation. At last, calibration patterns and procedures are explained. In chapter 5, the experiments and the testbed used to evaluate the system are described. Results of each test are presented and comparisons between systems are established.

In chapter 6, concluding remarks and overall results of this work are presented and possible future work and enhancements related to the new system are listed.

Electrical schematics of the evaluation testbed are included as Appendix to this work.

2 TRACKING TECHNOLOGIES AND SYSTEMS

In this chapter basic concepts used to evaluate tracking systems are presented. These definitions will be recalled in every section of this work where comparisons and performance measurements are made. Also some application scenarios for tracking systems as well as related projects are presented.

In sequence the existing tracking technologies and their positive and negative aspects are listed, followed by a brief explanation of the reasons why marker-based optical tracking has been chosen for this work. Then a list of the most significant commercial and research marker-based optical tracking systems is presented, followed by a coarse comparison among those systems and finally the requirements of the system to be developed are presented.

2.1 BASICS ABOUT TRACKING

A tracking system is a three-dimensional (3D) input device whose main objective is to acquire the position of a point in the 3D space, which is expressed by the 3 coordinates x, y, z. However, many applications work with objects instead of points, and therefore also require from the tracking system the orientation of the object in relationship to the coordinate system axes. The orientation is expressed by the angles to the axes, known as *pitch* (also known as elevation or attitude), *yaw* (also known as azimuth or heading) and *roll* (also known as bank), which are shown in the Figure 1. The set of position and orientation information is called pose.

Taking into account position and orientation information, tracking an object in 3D space requires the evaluation of 6 degrees of freedom (DoF), task that must be performed by the tracking system.

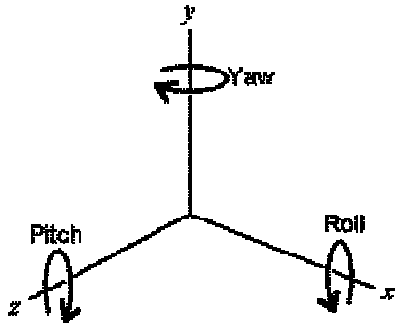

Figure 1 – Angles of Orientation for Objects in 3D Space – Pitch, Yaw, Roll.

In order to obtain this information, many technologies are available, such as electromagnetic, mechanical, inertial, acoustic, optical or radio-based tracking. The most used technologies are dealt with in the section 2.3.

The main properties of a tracking system, typically used for comparison purposes, are:

- Accuracy: is the mean error in the measurement of position or orientation of an object, calculated as the difference between the values (distance or angle of the tracked object) measured by the system (computed pose estimation) and the actual values (physical, real position or orientation) obtained by a reference system, expressed by

$$Accuracy = \mu_{error} = \frac{1}{N}\sum_{N}\left|value_{measured} - value_{real}\right|, \quad (1)$$

where N is the total number of measurements and value is the orientation or position value measured. Theoretically a tracking system has three accuracy values for position (x, y, z) and three for orientation (pitch, yaw, roll). In practice, only one accuracy value for translation and one for orientation are presented. Some vendors present accuracy as the root mean square (RMS) value of error. All experiments executed within this work consider accuracy as mean error.

- Precision: is the degree of mutual agreement among a series of individual measurements, expressed as the standard deviation of computed position or orientation error. Again there are three values for position precision and three for orientation precision, but in practice only one for each is used. Precision is calculated by

$$Precision = \sigma_{error} = \sqrt{\frac{1}{N}\cdot\sum_{N}\left(value_{measured} - \mu_{error}\right)^2}\,. \quad (2)$$

- Exactness: is the RMS value of the error in measurement of position or orientation, which incorporates both mean error and standard deviation, expressed by

$$Exactness = RMS_{error} = \sqrt{{\mu_{error}}^2 + {\sigma_{error}}^2}\,. \quad (3)$$

- Resolution: is the smallest change in the position or orientation of an object which can still be detected by the system.

- Update rate: how many times per unit of time the system provides new position or orientation information.

- Latency: time taken by the system since an actual change in position or orientation of a tracked object until this change is reported by the system's output. In other words, time needed by the system to calculate new position and orientation data. Latency and update rate can be related if the system does not pipeline calculations, which is usually the case. Under this condition, calculations for the next frame can only be initiated when calculations for the last frame have been finished. In this case the period of calculation (latency) determines the maximum update rate of the system.

- Range of operation (working volume): region inside which the system is able to provide its service, even though in border regions accuracy and resolution specifications might not be met any more.

These properties will be considered later in this work in order to evaluate the tracking systems properties and performance.

Regarding accuracy, precision and exactness definitions, each manufacturer of tracking systems has its own specific performance criteria and concepts. For example, see (FRANTZ, 2003) for the definitions used by Northern Digital, Inc. As a rule of thumb the following citation by *Yiding Wang can be used: "Accuracy is telling the truth . . . Precision is telling the same story over and over again."*

2.2 APPLICATION SCENARIOS FOR TRACKING SYSTEMS

Any VR/AR immersive application scenario needs at least some information about instant position and orientation of a subject or object. Here some examples of utilization of tracking system in AR/VR scenarios are given.

2.2.1 Cultural Heritage

Immersive AR/VR applications can be used to preserve cultural heritage information, such as ancient history information. Buildings destroyed a long time ago can be shown standing, old monuments and constructions can be displayed on their original locations, with their projections being mixed with the current existing scene. This is a typical outdoor AR scenario.

One example of project which implements this application is ArcheoGuide (HILDEBRAND, 2000), a system consisting of a mobile multimedia system, which provides visitors to archeological sites with the possibility of receiving multimedia information (images, sounds and text) about points of interest depending on their location on the site. Figure 2 shows the virtual image of an ancient Greek temple projected over the real location of the temple, as a result of using ArcheoGuide.

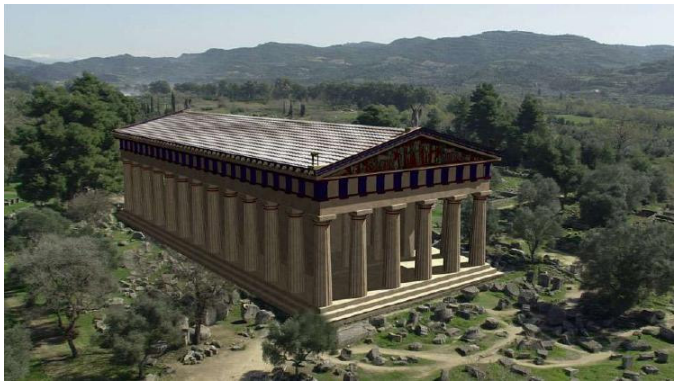

Figure 2 – ArcheoGuide: Virtual Image of an Ancient Greek Temple Projected over its Real Location.

2.2.2 Medicine

As an indoor AR application, the development of mechanisms to help surgeons in minimally invasive and non-invasive surgery approaches has become a major field of research. A key problem in this issue is renouncing the use of equipments which major interfere in the proceedings of a surgery such as e.g. Head-Mounted Displays (HMD).

A project which addresses this issue is MEDARPA (SCHWALD, 2002), which uses semi-transparent displays and a mixture of optical and magnetic tracking for medical instruments and patient tracking. Figure 3 shows this application being used. In this system information is provided to the surgeons through the see-through devices during surgeries, in order to assist them.

Figure 3 – Use of MEDARPA during Medical Procedures.

2.2.3 Industrial Applications

One of the most promising applications of AR/VR systems is in the industry, due to the almost uncountable possibilities of use and enhancements which can be achieved in this area.

Factory Planning was a task developed until few years ago only by ordinary simulation and usual visualization of text results. Currently the use of 3D graphics and animations to show the obtained results is much more intuitive, expanding the benefits of this technique by providing formatted results to a larger audience. The constant integration of planning, simulation, analysis of results, presentation and corrections enhances the whole development cycle, and VR plays a very important role in this matter.

Product design, in every possible scale, can take advantage of AR/VR systems. This is a typical indoor scenario, where former physical models are replaced

by modern virtual ones (VR), or by a mixture of physical and virtual models (AR). An example of this application is the SketchAR software (SANTOS, 2003), developed within the SmartSketches Project, shown in Figure 4.

Simulation of physical phenomena, usually integrated with design, is a major application of AR/VR in the industry. An example of this application is the Visicade (BENÖLKEN, 2004) project, which proposes the integration of Computer-aided Design/Manufacturing (CAD/CAM) tasks into VR systems.

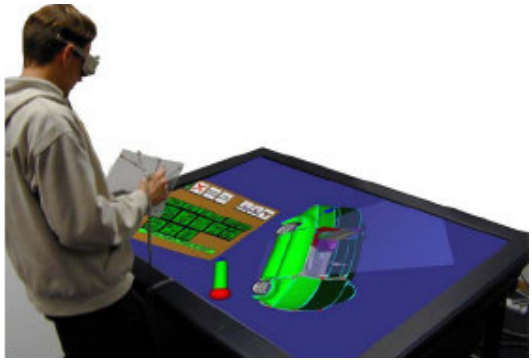

Figure 4 – Immersive Car Styling with SketchAR (SANTOS, 2003).

Another important application field is maintenance, due to increasing difficulties of keeping technical personnel updated with latest training and maintenance topics. In this field, the ARVIKA[1] Consortium in Germany has come up with the AR Browser solution, shown in Figure 5, which is an example of applied AR in maintenance systems.

Figure 5 – AR Browser, developed by ARVIKA

2.3 AVAILABLE TRACKING TECHNOLOGIES

The following sections present the main available technologies used in current tracking systems. Each one has advantages and disadvantages, which are related to the desired application and the trade-offs involved. A valuable source for this chapter was (HAND, 1993). At the time of this work, the newest technology available was

[1] ARVIKA Consortium. Website available at <www.arvika.de>. Last accessed on December 16th, 2005.

radio-based tracking, e.g. Global Positioning System (GPS), which can be usually found in the most modern outdoor tracking systems.

2.3.1 Electromagnetic

An electromagnetic tracker consists of a transmitter and a receiver. An oscillating magnetic field is generated in three orthogonal coils, each one corresponding to a different axis of the 3D coordinate system at the transmitter side. At the receiver side, the magnetic field is sensed by three corresponding coils. The received intensity of the magnetic field in each direction varies with the distance (cubically) between transmitter and receiver as well as with the difference in orientation between them.

Measuring the intensities of the magnetic field induced in the receivers enables calculation of the position and orientation. For a 6 DoF system, measurements in 3 sets of 3 coils each are needed. Usually receivers are attached to the tracked object or subject, transmitters are fixed.

The magnetic field can be generated either with low frequency Alternate Current (AC) or pulses of Direct Current (DC).

The typical accuracy, precision and resolution of electromagnetic trackers are in the millimeter range. Typical update rates are around 100 Hz, with latency times as low as 5 ms.

Main advantages are: the small size of the non-tethered receiver, allowing it to be attached to a subject's hand or head with minimum intrusion; total freedom of movement; and the usually high update rate.

Disadvantages are: small working volume, due to limited reach of transceivers; ferromagnetic interference caused by metal objects; Eddy currents induced by AC electromagnetic field, which generate distortions – DC transmitters overcome this problem; and sensitivity to electromagnetic interference (EMI).

2.3.2 Mechanical

Mechanical tracking uses physical connections between the tracked object and the system, which is built with metal strings or any other type of linkages, in general connected to rotating joints. Usually potentiometers or optical encoders are used to measure the rotation of the joint. Given the angle and the length of the rods or wires, together with the known position of the fixed hardware, it is possible to calculate the position of the object in the space. Figure 6 shows a simple mechanical finger tracking system.

Accuracy, precision and resolution are naturally very high due to the construction of the system. Update rate tends to be high due to simplicity of sensors, typically above 100 Hz.

Advantages are the small latency of the system, due to the simplicity of the sensors, the immunity to environment interference, due to lack of transmitters or receivers, and the usual low cost, also due to the fair simple construction of the sensor.

Most systems also provide the possibility of force-feedback i.e. transmitting movements back to the user or object, which is a unique property among the tracking technologies.

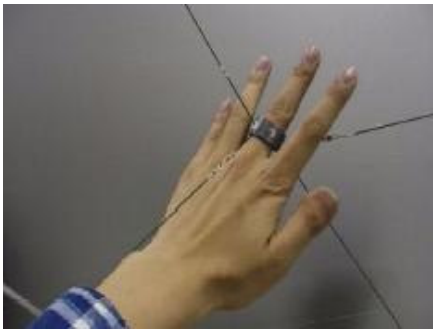

Figure 6 – Mechanical Tracker with Force Feedback Capabilities

Typical disadvantages are the small working volume and freedom of movement, because the user must be tethered to the system, and the usual small life cycle of the contained moving parts, when compared to non-tethered systems.

2.3.3 Inertial

Inertial trackers use gyroscopes to measure changes in rotation around axes and accelerometers to measure changes in the position over the axes. Gyroscopes work based on the conservation of angular momentum, so changes in rotation can be measured as changes in angular speed. Figure 7 shows the symbolic representation of a gyroscope for 3 axes.

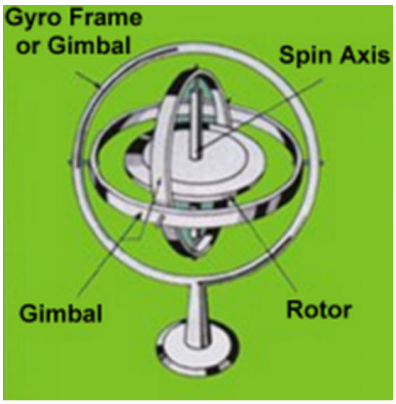

Figure 7 – Gyroscope able to measure Rotation around 3 Axes

Accelerometers work based on measure of the force exerted on a mass, since acceleration cannot be measured directly. Based on Newton's Second Law of Motion, acceleration can be calculated from force and then integrated twice, in order to obtain position. Due to double integration and use of differential distance, measurements tend to have large error growing rates and large errors if periodic synchronization is not executed. This can lead to poor accuracy and precision properties. Latency times are typically low.

Advantages of inertial tracking systems are: no transmitters or receivers are necessary, since the units are self-contained; working volume is theoretically unlimited, so outdoor large area tracking is a suitable application; lightweight equipments, suitable for wearing with low intrusion.

Disadvantages are: error accumulation due to numerical integration, what demands periodic recalibration – typically used here are hybrid systems (BISHOP, 2001); drifting in the axis of rotation of a gyroscope due to remaining friction between mechanical parts – also corrected with sensor fusion by use of hybrid systems.

2.3.4 Acoustic (Ultrasound)

Acoustic tracking uses a well-known technique called time-of-flight (TOF) measurement. One or more emitters send a sound signal, typically in the ultrasonic region (frequency above 40 KHz), which is then received by several sensors (microphones). The time between emitting and receiving the sound pulses is measured and distance between both sides can be calculated based on the speed of sound in the air. Another technique also used is Phase Coherence, which measures phase difference between the received and the sent signals. For a full 6 DoF tracking system, 3 transmitters and 3 receivers are necessary.

Typical accuracy values are in centimeter range.

Advantages are the small size and weight as well as availability and low cost of transceivers.

Disadvantages are: speed of sound in air varies with temperature, pressure and humidity; low update rate; undesired reflected echoes of the sound signal generate interference; sensitive to external acoustic interference; line-of-sight must be maintained between transmitter and receiver.

2.3.5 Optical

Optical tracking systems use a variety of sensors, from cameras to Lateral Effect Photodiodes (LEPDs), to detect the light emitted or reflected by objects and so determine their position in a 3D environment.

An optical tracking system can be marker-based – uses active or passive markers which are considered as points – or markerless – based on matching of the observed two-dimensional (2D) scene with pre-defined forms. In the first case usually triangulation methods are used to obtain one 3D point (position) from two 2D points. Additional 3D points are required to extract orientation information. In the second case numerical iterative or algebraic methods are used to reconstruct the 3D position and orientation of the target (set of markers) based on the 2D projection. See sections 3.4 and 3.5 for details on algorithms for pose estimation.

Marker-less optical tracking systems have as drawback higher latency time (due to more processing steps), higher computer load (due to elevated complexity of the processing tasks) and lower precision (due to inaccurate matching between pattern and real image). Nevertheless they have evolved quickly and are a major research topic, due

to the advantage that markers are not needed and thus there is no interference in the scene.

Typical detectors are LEPDs, Quad Cells, analog or digital – Charge-coupled Device (CCD) or Complementary Metal-oxide-Semiconductor (CMOS) sensors - video cameras. As targets (or markers) passive objects could be used, like reflective materials, or active ones, like LEDs. Infrared spectrum region is typically used to avoid ambient light interference.

Figure 8 shows an example of camera and markers, provided within the AR Tracking GmbH. tracking system.

Figure 8 - Hardware for Infrared Optical Tracking System from AR Tracking GmbH

Optical tracking systems can be classified by:

- Configuration: a system is said to be Inside-out if the camera is able to move and the objects to be tracked are fixed; a system is considered Outside-In if the camera is fixed and the objects to be tracked are able to move, as shown in Figure 9.

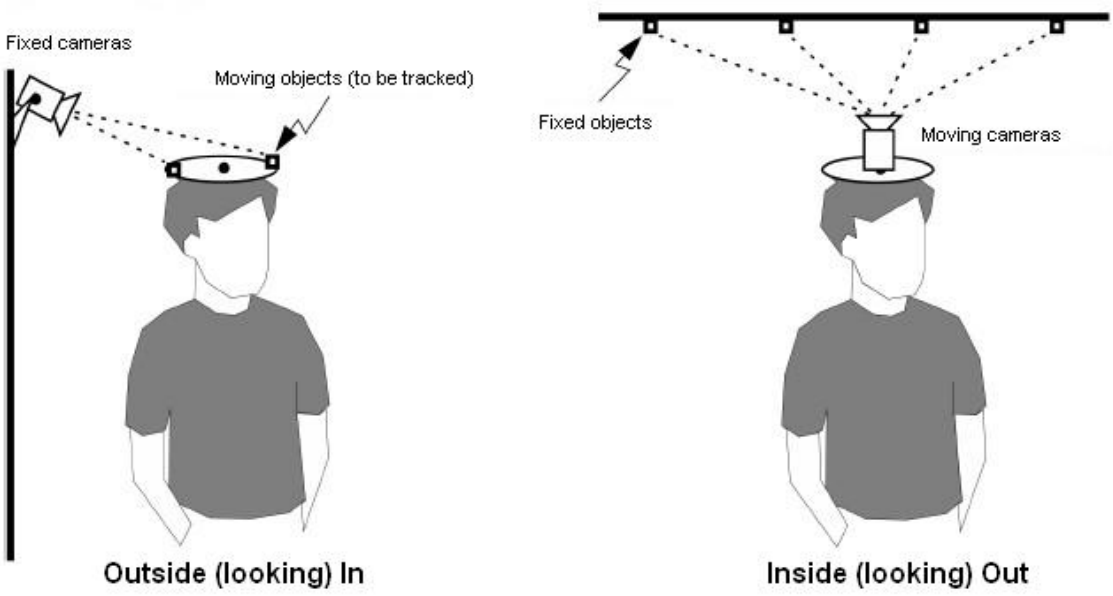

Figure 9 - Outside-In and Inside-Out Configurations

- Real-time capability: a system is considered to be real-time capable if it completely processes tracking information immediately after acquisition, with a minimum acceptable update rate, which is application dependant. Otherwise it is called an offline system.

- Type of markers (if marker-based): the markers (also called targets) are the tracked objects, which can be passive (only reflect light) or active (contain light emitters). Reflective labels are passive markers. LEDs are active markers.

Typical accuracy, precision and resolution values are in the sub-millimeter range. Typical update rates are between 50 and 200 Hz.

Advantages are the good update rate, high accuracy, possibility of large area tracking with multiple sensors.

Disadvantages are: line-of-sight must be maintained (as in acoustic tracking); high costs in comparison with other technologies; need of computer processing power; sensitivity to optical noise and spurious light (minimized if infrared is used).

In section 2.4 the features and drawbacks of optical tracking systems are highlighted. In sections 0 and 2.6 both research and commercial available systems are listed.

2.3.6 Radio-based

Tracking systems based on radio-frequency technologies are basically suitable for large area tracking, using GPS based systems for outdoor tracking and Wireless Local Area Network (WLAN) or radar based systems for indoor tracking.

GPS based systems use the network of geostationary satellites around the earth to provide location information. As working principle the time of flight (TOF) delays between satellites and receivers are measured and compared. By use of a triangulation algorithm the position of the receiver can be tracked, as long as 4 satellites are available. Recently the US Government has switched off the Selective Availability feature, which worsens the accuracy of the system, allowing typical accuracy values to be around 10 m nowadays.

Using enhancements like DGPS (differential GPS) the accuracy of GPS based systems has grown largely in the last developments. DGPS merges the information from a stationary GPS receiver, located near the actual receiver, with the information available in the receiver itself. By comparing the difference between provided position for the fixed receiver and the known real position, the errors in the GPS system can be extracted and so corrections in the provided position for the actual receiver can be made. With this progress, accuracy of 1 m can be obtained.

In further enhancements like RTK-DGPS (Real time kinematics – DGPS), which uses extra information received from the satellites (a second carrier wave in the signal is also used for error estimation), even better accuracy can be obtained. This system is still being developed and researched, but first experiments show that accuracies of 2 cm can be obtained under drawbacks such as very high latency times.

WLAN based systems use existing WLAN interfaces to track mobile equipments such as notebooks or Personal Digital Assistants (PDAs) or any device equipped with a WLAN card. The working principle is measurement of the propagation delays between the base-stations (access points) and the nodes. The greater the number of WLAN access points, the better the accuracy of the system. Typical accuracy is around 1 m.

Radar based systems measure the wave's TOF to determine the distance from the sensors to the tracked object and, by sweeping in different directions, the exact position of the tracked object can be determined. A possible application is real time tracing of vehicles and machines (e.g. cranes, forklifts and trucks) in industrial environments such as factories. Typical accuracy values are at least 10 cm.

Typical latency values for any radio-based tracking system are high, at least 50 ms, typically 100 ms.

The main disadvantage of radio-based tracking systems is the bad accuracy, still not suitable for AR/VR applications. In some cases such as satellite systems latency is also a major problem, preventing the use of these systems for real-time AR/VR applications.

The main advantage relies on the high availability in large areas especially outdoors. No other technology provides worldwide range of operation as satellite based systems.

2.4 DECISION FOR MARKER-BASED OPTICAL TRACKING SYSTEMS

Indoor immersive virtual and augmented reality applications, such as medicine and most of the industrial application scenarios, demand some specific technical properties for the tracking systems used in their implementations, such as:

- *High accuracy*: interaction between virtual and real objects demand exact positioning, so that the user percepts it as only one environment;
- *Low latency and high update rate*: in order to provide real-time capability and e.g. allow overlapping of virtual and real objects if see-through displays are used;
- *High degree of freedom of movement to the user*: the system should not impose obstacles to the user;
- *High robustness against interferences*: for reliability reasons.

After listing the main available tracking system technologies, the main pros and contras of each technology can be compared to those desirable features listed above. Table 1 shows this comparison in a simplified way. Critical values in the properties are marked.

Table 1 – Comparison of Tracking System Technologies against Indoor AR/VR Application Requirements - Marked Cells show Critical Values

Technology	Accuracy	Latency	Update Rate	Freedom of Movement	Robustness to Interferences	Additional Problems
Desired Value	high	low	high	high	high	none
Electro-magnetic	high (sub-millimeter)	very low (typ. 5 ms)	high (100 Hz)	high	low (sensitive to metals and electromagnetic field sources)	small working volume not expandable;
Mechanical	high (sub-millimeter)	low (<10 ms)	high (>100 Hz)	very low (user tethered to the system)	high	small working volume, user cannot interact with other objects due to tethers;
Inertial	medium (high error growing rate)	low (<10 ms)	very high	very high (user non-tethered and without defined working volume)	high	accumulated position error due to integration and differential measurement;
Acoustic	medium (centimeters)	low (<10 ms)	high (>100 Hz)	medium-high (line of sight problem)	medium (sensitive to acoustic interference and echoes)	speed of sound varies with environmental conditions;
Optical	high (sub-millimeter)	low-medium (10 to 40 ms)	medium-high (50 to 200 Hz)	medium-high (line of sight problem)	medium-high (using infrared range)	sensitive to infrared light sources and optical noise;
Radio-based	low (meter or tens of meters)	high (>50 ms)	low (due to high latency)	high	medium-high (GPS demands line of sight to satellites)	none

From the comparison shown in Table 1 it can be seen that optical tracking systems do not have the best specific properties among all systems, but also do not have any critical values. However there are some drawbacks such as:

- *Freedom of movement* - due to the line of sight problem, the user cannot block the way between the markers and the cameras. The simplest solution is using more cameras in order to increase the likelihood that at least one or two (stereo tracking) cameras see the markers;

- *Robustness* – optical tracking systems are sensitive to optical noise like undesirable light sources and reflections. The use of infrared light minimizes this problem, reducing it to sensitivity only to the less frequent infrared optical noise.

After minimizing these problems, optical tracking systems become the best trade-off between requirements and actual properties, composing the best suitable set of attributes among all mentioned technologies. Marker-based optical tracking reduces latency time, due to the simplification of image pre-processing tasks, and increases robustness, because the probability of artifact detection is higher.

For the reasons aforementioned a marker-based optical tracking system will be developed in this work. In the following sections this type of system is investigated. Commercial as well as research systems which fulfill this attribute are listed and examined.

2.5 COMMERCIALLY AVAILABLE MARKER-BASED OPTICAL SYSTEMS

In this section the most representative commercially available marker-based optical tracking systems, at the time of this work, are presented in order to investigate existing systems which match the properties defined in section 2.4. In each system the main attributes are presented, especially the properties listed in section 2.1. Many vendors do not provide complete information about their systems. The formats and procedures used by vendors in measurements to obtain the technical information are diverse, yet the highest standardization level has always been sought in the presented information. An extended list of commercial systems can be found in (BUAES, 2005). The information for this section was also collected from (RIBO, 2001b) and (SANTOS, 2005).

Many of the following systems have been created for medical use (gait analysis, ergonomics) or for the animation industry (motion capture for movies, basically). Nowadays most of these systems can be or already were adapted also to industrial applications as well as to almost any indoor VR/AR scenario.

2.5.1 ARTtrack and DTrack (Advanced Realtime Tracking GmbH.)

ARTtrack[1] is an outside-in, infrared-based optical tracking system which uses two or more cameras per scene in order to obtain tracking information (position and orientation, so 6 DoF) by means of stereo vision algorithms.

The cameras are equipped with CCD sensors, daylight blocking filter and infrared enhancing filter, working with a wavelength of approximately 880 nm. Additionally they have infrared flash strobes (250 high-power infrared LEDs) to illuminate the scene. The use of infrared light enhances robustness against undesired optical noise, as explained in section 2.4. Flashes must be synchronized through an external sync signal, which is provided to each camera. Figure 10 shows the ARTtrack1 cameras, as this model is called. The camera has a maximum Field of View (FoV) of 60 deg in horizontal direction and 45 deg in vertical direction.

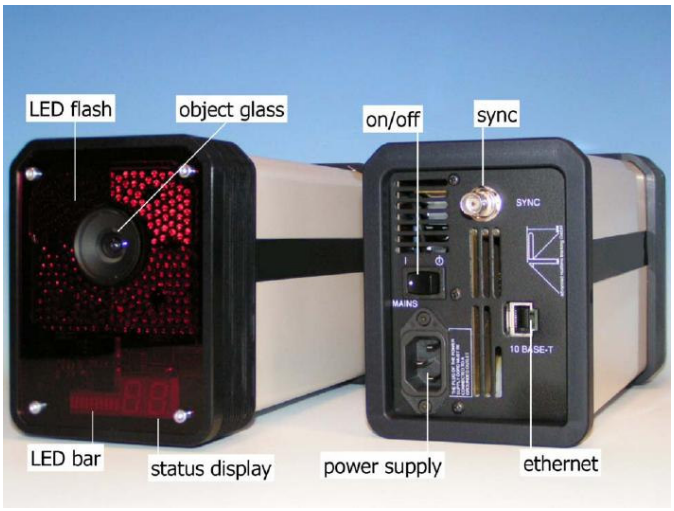

Figure 10 – Front and Rear View of ARTtrack1 Camera

The cameras are equipped with embedded processors running real-time Linux and thus have on-board image pre-processing to correct lens deformation in the image as well as to apply simple algorithms to the acquired image as e.g. Threshold Filtering - to remove non-IR (infrared) components present in the picture.

Markers of the system are small spheres or circles covered with a light reflective material. Each camera is capable of tracking up to 20 markers. Each object to be tracked (also called target) must contain at least 4 markers.

Each camera sends the processed images (2 DoF, in image coordinates data) to a central computer (tracking server) by means of an Ethernet interface, using User Datagram Protocol / Internet Protocol (UDP/IP) datagrams (small packets, no error detection/correction, thus faster if communication channel is error-free). The central server runs the DTrack software, which processes the received data and generates complete 6 DoF tracking information by means of triangulation (stereo vision)

[1] Advanced Real Time Tracking GmbH. ARTtrack System: Product Information. Available at <http://www.ar-tracking.de>. Last accessed on December 16th, 2005.

algorithms. The cameras' maximum pixel resolution is 658 x 496, nearly the Phase Alternating Line (PAL) system resolution (720 x 486).

The accuracy of the system has been tested, according to the company, in following conditions: working area of 3 x 3 m, 4 cameras, and all markers fully visible (no blocking of line of sight). Results are 0.4 mm of accuracy in position estimation and 0.12 deg in orientation estimation. Precision, here calculated as the standard deviation from mean value in a whole series of tracking data (repeatability), is 0.06 mm for position and 0.03 deg for orientation.

The system maximal range of operation is up to 10 m distance from the camera, but the area of 3 x 3 m used for accuracy measurements suggests this should be a usual working volume for a system with minimal configuration. The maximum working volume with multiple cameras is 300 m³.

The costs of the system start around EUR 30,000 including 2 cameras and one tracking server running DTrack software, which is the basic configuration. Table 2 summarizes the properties of the system.

Table 2 – Summary of Main Properties of ARTtrack System

Maximum Update Rate (at given camera resolution)		60 Hz @ 658 x 496
Accuracy	**Positional (mm)**	0.4
	Angular (deg)	0.12
Precision	**Positional (mm)**	-
	Angular (deg)	-
Resolution	**Positional (mm)**	0.06
	Angular (deg)	0.03
Maximum Latency (ms)		20 - 40
Maximum Working Volume (m³)		300
Price – minimal configuration		EUR 30,000

2.5.2 HiBall-3100 (3rdTech, Inc.)

This is an inside-out optical tracking system developed initially as a research project in the Department of Computer Science of the University of North Carolina at Chapel Hill[2], USA (WELCH, 2001). Later a joint-venture between the University and the company 3rdTech was created in order to commercialize the system.

The working principle is based on lateral effects photodiodes (LEPDs) - components that generate a signal proportional to the position of the incoming light on one specific direction - which are mounted on a portable device (HMD) attached to the object to be tracked, and which look upwards to view infrared LEDs mounted on the ceiling of the area to be tracked. The device containing LEPDs is called HiBall Optical Sensor and the set of infrared LEDs on the ceiling is called HiBall Ceiling Beacon Arrays. Figure 11 shows a HiBall Optical Sensor and part of a ceiling equipped with the arrays.

[2] Tracker Project Website at the University of North Carolina at Chapel Hill. Available at <http://www.cs.unc.edu/~tracker/>. Last accessed on December 16th, 2005.

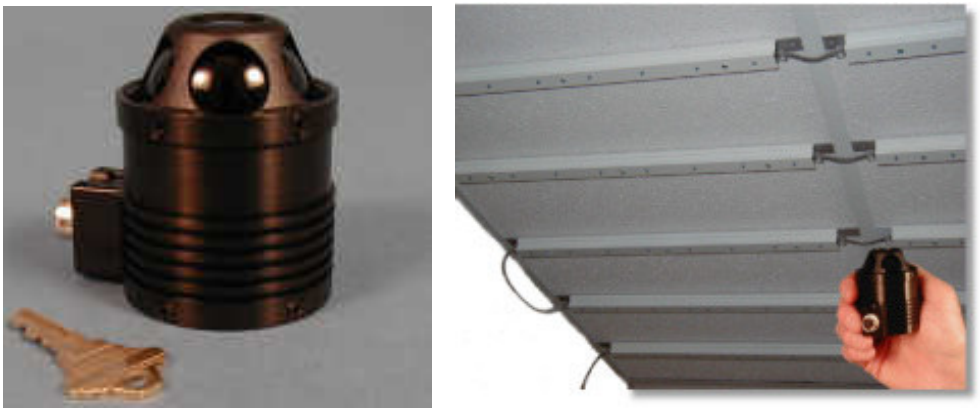

Figure 11 – HiBall Sensor (left) and Ceiling Equipped with Beacon Array Modules of HiBall 3100 System (right)

Each Optical Sensor (simply known as HiBall) is a cluster of 6 lenses and 6 LEPDs arranged so that each diode can view the ceiling LEDs through several of the 6 lenses, providing 26 viewing possibilities in a single HiBall, which improves visibility of the ceiling LEDs.

Based on the known location of the LEDs on the ceiling, the known geometry of the user mounted optical sensors and the use of the SCAAT (Single Constraint At A Time) algorithm, the system is able to extract complete position and orientation information from the scene. Currently the HiBall Sensors are tethered to the central personal computer (PC) where tracking calculations are performed. The substitution for a wireless interface is though foreseen and easily executable, according to (WELCH, 2001).

The SCAAT algorithm makes use of an Extended Kalman Filter to provide position and orientation updates after each individual Light Emitter Diode (LED) sighting, what allows a very high update rate for the system.

Since the LEDs on the ceiling are passive in the sense that they are just observed by the optical sensor, the expandability of the system is natural and almost unlimited. Just adding more Beacon Arrays increases the range of operation. Each Beacon Array Module contains 6 strips of LEDs, 8 LEDs per strip. By composing with these modules, systems can be built in configurations as little as 3.6 x 3.6 m as well as in areas with 12 x 12 m.

The system performs autocalibration which allows installation of not exactly aligned ceiling beacon arrays. After some samples the system automatically tunes the actual LED positions, applying the resulting corrections to the pose information. This feature makes installation very easy and allows the commercialization of the system.

The absolute accuracy of the system, using one HiBall sensor in a single 3.6 x 3.6 m configuration, reaches 0.4 mm in position and 0.02 deg in orientation. Precision values are not provided by the vendor.

The system's update rate is maximum 2,000 Hz using only one HiBall Sensor and decreases linearly with increasing number of sensors – by 4 HiBall Sensor, update rate is 500 Hz each sensor. The working volume is theoretically unlimited. Tests

have already been successfully performed with 200 Beacon Array modules in a volume of 12 x 12 x 3 m.

Costs for a configuration with area 3.6 x 3.6 m and only one HiBall Sensor start at US$ 31,000. Table 3 summarizes the main properties of the system.

Table 3 – Summary of Main Properties of HiBall-3100 Tracking System

Maximum Update Rate		2,000 Hz
Accuracy	**Positional (mm)**	0.4
	Angular (deg)	0.02
Precision	**Positional (mm)**	-
	Angular (deg)	-
Resolution	**Positional (mm)**	0.2
	Angular (deg)	0.01
Maximum Latency (ms)		< 1
Maximum Working Volume (m^3)		Unlimited
Price – minimal configuration		US$ 31,000

2.5.3 Eagle/Hawk Digital System (Motion Analysis Corp.)

At the time of this work, Motion Analysis Corporation is probably the world leader in providing motion capture optical systems, particularly regarding animation production in the entertainment business.

The Eagle Digital System[3] is the enhanced version, the Hawk Digital System the simplified one. As the ARTtrack, the Vicon Tracker and the Qualisys Motion Capture systems, this is a passive optical outside-in tracking system based on retro-reflective markers.

In the maximum performance configuration the system is built with Eagle-4 CMOS Digital cameras, recently released and shown in Figure 12, capable of 2,352 x 1,728 pixel resolution at a full frame rate of 166 Hz. At reduced resolutions the cameras reach 10,000 Hz frame rate. These cameras are equipped with 237 infrared LEDs around the lens, which provide infrared flash function. The cameras have on-board processing through microprocessors and Field Programmable Gate Arrays (FPGA). An Ethernet 100 Mbps interface is available to send the processed data to an Ethernet Switch (EagleHub, also provided within the system's package), which also has analog video inputs and framegrabbers, allowing analog cameras to be integrated in the same systems. Power supply to the cameras is also provided through the Ethernet interface. The EagleHub is connected to a central computer where the EVaRT (EVa Real-Time software) finally processes received data and provides the user with the tracking data.

Other cameras are much cheaper and still have plausible properties, as the Eagle Digital Camera, capable of processing 480 frames per second at a full resolution of 1,280 x 1,024 pixels.

[3] MOTION ANALYSIS CORPORATION. Eagle-4 Digital System. Specifications available at <http:// www.motionanalysis.com/pdf/systemeagle4.pdf>. Last accessed on December 16th, 2005.

The system uses triangulation (stereo vision) algorithms to calculate tracking data based on the information from at least 2 cameras.

Accuracy values are not provided by the vendor. However, values obtained from third-party comparison tests[4] show 2.77 mm of mean absolute error in position and 0.52 deg of mean absolute error in orientation measurements. The same source presents results of 0.36 mm for the positional precision as well as 0.13 deg for the angular precision. Resolution was not provided by the vendor.

Figure 12 – Eagle-4 Digital Camera

Configurations can be built with up to 64 cameras in order to expand the working volume. The vendor reports having built systems as large as 16.5 x 16.5 x 3 m or 816 m³ with 50 cameras. From the aforementioned comparison tests it can be also concluded that a reasonable working volume where the system keeps the above shown attributes is 7 x 7 x 3 m or 63 m³.

The costs for a basic configuration of 4 Hawk cameras with maximal resolution of 640 x 480 start at EUR 40,000. A system with 4 Eagle Digital cameras with maximal resolution 1,280 x 1,024 has prices starting at EUR 65,000. Table 4 shows a summary of the properties of the Eagle Digital System.

[4] COMPARISON MEETING OF MOTION ANALYSIS SYSTEMS 2002, 27-29th July, held at Japan Technology College, Tokyo, Japan; Results available at <http://www.ne.jp/asahi/gait/analysis/comparison2002/Result/basic/basic_eng.html>. Last accessed on December 16th, 2005.

Table 4 – Summary of Main Properties of Eagle Digital System

Maximum Update Rate (at given resolution)		480 Hz @ 1,280 x 1,024 166 Hz @ 2,352 x 1,728
Accuracy	**Positional (mm)**	2.77
	Angular (deg)	0.52
Precision	**Positional (mm)**	0.36
	Angular (deg)	0.13
Resolution	**Positional (mm)**	-
	Angular (deg)	-
Maximum Latency (ms)		6 (1 frame)
Maximum Working Volume (m³)		816
Price – minimal configuration		EUR 65,000

2.5.4 Vicon MX (ViconPeak)

On Februar 10th, 2005 Vicon officially announced the acquisition of Peak Performance Technologies, Inc., the new company to be called ViconPeak. Peak Performance was also a vendor of tracking systems, having as main product the well known Peak Motus 3D Optical Capture Sysem.

The Vicon Tracker MX is an optical, infrared-based, passive marker tracking system, very similar to the ARTtrack system. Retro-reflective markers are positioned on the objects to be tracked. At least 4 markers are needed to provide 6 DoF information, as well as 3 cameras – the additional camera is needed for calibration purposes. System is able to track up to 50 objects.

The cameras, like the MX3[5] model shown in Figure 13, use CMOS sensors and can work in visible light (623 nm), near infrared (780 nm) or infrared (875 nm) regions of the spectrum. They have onboard processing to filter undesired noise in images as well as to pre-process the data, executing, for instance, threshold filtering and other possible corrections in the images before they are sent to the central tracking server. Camera's frame rate at full resolution varies from 166 Hz (MX40) up to 484 Hz (MX13), and resolution varies from 659 x 493 pixels (SVCam), nearly PAL system resolution, up to 2,352 x 1,728 pixels (MX40).

The accuracy for a system using MX40 cameras is declared by the vendor as 0.1 mm for position and 0.15 degree for orientation, measured independently "through a large 3D space". Resolution reaches 0.2 mm for translation and 0.2 deg for rotation. Precision values are not provided.

The working volume is limited by the maximum distance in which a marker can still be seen by the cameras, which is 5 m with the SVCam camera and e.g. 50 m with the M2 camera. With multiple cameras the maximum range of operation is as large as 1,000 m³. The company claims to have built the world's largest tracking system for motion capture purposes with 200 cameras. A typical working volume in which accuracy and resolution features are still valid and were also calculated is 4 x 4 x 3 m.

[5] VICONPEAK. MX3 Camera Brochure, available at <http://www.viconpeak.com/downloads/Vicon_MX3_hr.pdf>. Last accessed on December 16th, 2005.

Costs for a basic configuration with 3 MX-3 cameras begin at EUR 35,000. Table 5 shows a summary of the main properties of the Vicon MX System.

Figure 13 – Vicon's MX3 Camera

Table 5 – Summary of Main Properties of Vicon MX System

Maximum Update Rate (at given camera resolution)		160 Hz @ 2,352 x 1,728
Accuracy	**Positional (mm)**	0.1
	Angular (deg)	0.15
Precision	**Positional (mm)**	-
	Angular (deg)	-
Resolution	**Positional (mm)**	0.2
	Angular (deg)	0.2
Maximum Latency (ms)		< 10
Maximum Working Volume (m^3)		1,000
Price – minimal configuration		EUR 35,000

2.5.5 SMART (BTS spa.)

The SMART[6] Motion Capture System from BTS, which has emerged from a fusion between BTS SpA and eMotion Srl, is an optical infrared-based tracking system, just as the ARTtrack and the Vicon MX. The working principle is the same as those systems i.e. small retro-reflective markers are illuminated by infrared flashes (infrared LED) built on the cameras. Their position is then tracked by the system, calculated by use of triangulation algorithms, so at least 2 cameras are needed. The vendor however recommends at least 3 cameras for better accuracy results.

Tracked markers are spheres which can have from 2 mm up to 30 mm diameter. The cameras are equipped with CCD sensors and as usual daylight-blocking

[6] SMART. SMART Motion Capture System. Information. Available at <http://www.bts.it/proser/elisma.htm>. Last accessed on December 16th, 2005.

filters, allowing only infrared light (880 nm) through. The possible frame rates for the cameras are 50, 60, 120 and 240 Hz. The maximal camera resolution is 640 x 480 pixels at 120 Hz.

The accuracy of the system has been tested by the company in a few experiments under following conditions: 6 cameras, with 50 Hz frame rate, in a calibrated working volume of 3 x 2 x 2 m. As test tool a 500 mm rigid wand with one marker at each end is used.

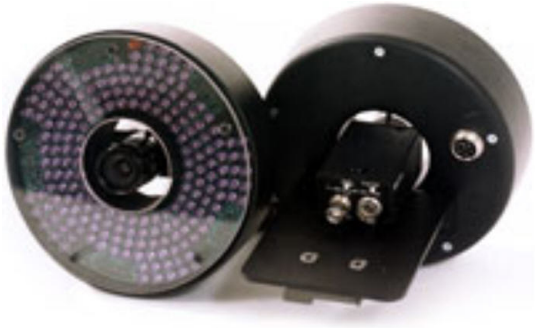

Figure 14 – Cameras used in SMART, with Infrared Flash Strobes (LEDs) around Lenses

In a first experiment the wand is placed on the floor, aligned with the X axis of the calibrated volume reference frame, and is kept stationary during the whole duration of the test (10 s). Results are shown in Figure 15. For the error, calculated as the difference between the position measured by the system and the real position, a standard deviation of 0.06 mm was obtained with 0.38 mm peak-to-peak error.

In a second experiment the wand is moving around the working volume always in a nearly vertical position for 20 s. Results are shown in Figure 16. A standard deviation of 0.5 mm is obtained with 4.04 mm peak-to-peak error.

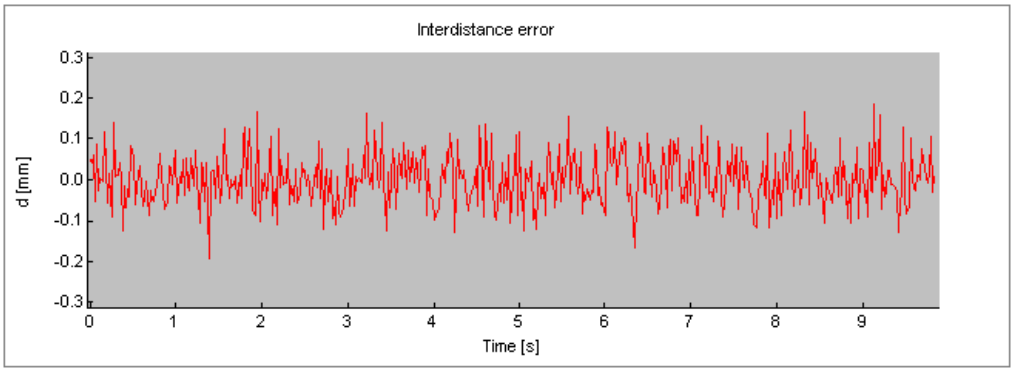

Figure 15 – Results of Translation Error in a 10 s Duration Test run on SMART

The results show that the overall system's precision is 0.5 mm for translation. Positional accuracy is less than 1.0 mm, considered as the maximum mean value of the absolute error in the experiments, measured in a working volume of 4 x 3 x 2 m. Orientational accuracy, precision and resolution values were not provided by the vendor.

The maximum working volume is 10 x 10 x 3 m or 300 m³. Up to 9 cameras can be used to expand working volume. Table 6 summarizes the properties of the SMART Motion Capture System.

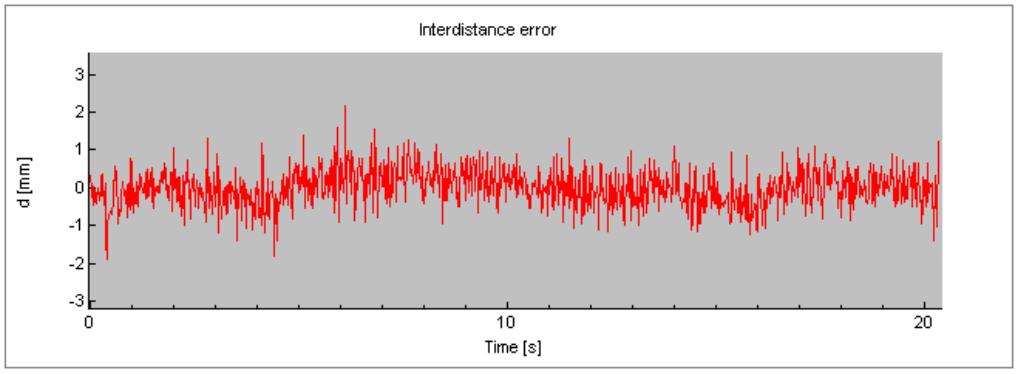

Figure 16 - Results of Translation Error in a 20 s Duration Test run on SMART

Table 6 – Summary of Main Properties of SMART Motion Capture System

Maximum Update Rate (at given resolution)		120 Hz @ 640x480
Accuracy	**Positional (mm)**	1.0
	Angular (deg)	-
Precision	**Positional (mm)**	0.5
	Angular (deg)	-
Resolution	**Positional (mm)**	-
	Angular (deg)	-
Maximum Latency (ms)		10
Maximum Working Volume (m³)		300
Price – minimal configuration		EUR 45,000

2.5.6 Visualeyez VZ4000 (Phoenix Technologies Inc.)

Phoenix Technologies has developed a complete family of optical tracking systems using active markers. VZ4000[7] is the system to be analyzed here, since it was the latest model for the time this work has been developed. The working principle is the use of active LEDs positioned on the objects to be tracked, which are turned on, one at a time, sequentially. The system uses therefore Time Domain Multiplex (TDM) method, since each marker uses only one time slot. The maximum number of tracked LEDs is 512. This feature reduces almost completely the correspondence problem, because every marker is uniquely identified.

The vendor does not provide detailed information about the sensors, but it can be inferred that very high resolution cameras are used, 3 pieces in a set. The

[7] PHOENIX TECHNOLOGIES INCORPORATED. Visualeyez VZ4000 Tracking System. Product Information available at <http://ptiphoenix.com/products/visualeyez/ vz4000.php>. Last accessed on December 16th, 2005.

cameras must work synchronized with the control modules of the markers and must have on-board processing in order to reduce the computation load in a central server.

Figure 17 shows the main component – the 3 sensors' set – which, together with the control modules of the markers and a central server connected to these modules, composes the system.

The vendor offers tethered active markers in the basic configuration, but there is also the possibility of using a wireless module for communication between tracked objects (e.g. a person's body), composed of many markers, and the system's control module in order to turn the system into a non-tethered one.

Figure 17 – Visualeyez VZ4000 System, composed of 3 Sensors rigid coupled

A feature of the system is the high update rate around 4,350 samples per second, each sample composed of the complete tracking (position) information of one marker. In a typical application of 50 markers, the overall update rate of the system would be 87 Hz. The drawback is that system's update rate depends on the number of markers used.

The system's positional accuracy is 0.6 mm, using as measurement volume the following: 0.6 to 2 m distance; ± 40 deg in horizontal direction (yaw); ± 30 deg in vertical direction (pitch). Positional precision is not provided by the vendor. Angular accuracy is about 0.007 deg. Angular precision values are not provided by the vendor. Position resolution is 0.015 mm, measured at a distance of 1.2 m between marker and sensors. Angular resolution is not provided.

The working volume for just one tracker (composed of a rigid set of 3 sensors) is determined by the maximum field of view of the sensors in both directions. Figure 18 shows the field of view in both directions and a 3D representation of the whole working volume. Considering the maximum radius the maximum working volume is around 190 m³ for one tracker. The minimum distance between tracker and objects is 0.5 m. The vendor offers the possibility of expanding the system up to 24 trackers.

Prices start at US$ 61,000 for the simplest configuration with one single tracker system (VZ4000) which can be used to capture simple motions with little rotation e.g. upper body motion with facial captures or full body front/side view capture.

A complete four tracker system (VZ4000M4), suitable for complete angular coverage of a subject, enables capture of complex motions and full rotations with regular marker placements as well as multiple people, hand and facial captures. A system like this costs US$ 215,000. Table 7 shows a list of properties of the VZ4000 system.

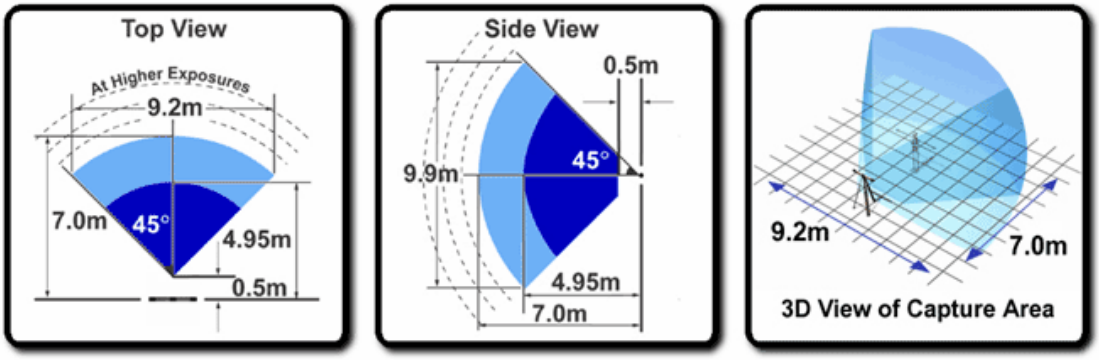

Figure 18 – Top and Side Views showing FoV Angles in VZ4000 (left); Maximum Working Volume with One Tracker (right)

Table 7 - Summary of Main Properties of Visualeyez VZ4000 System

Maximum Update Rate		4,350 markers / s (~87 Hz @ 50 markers)
Accuracy	**Positional (mm)**	0.6
	Angular (degree)	0.007
Precision	**Positional (mm)**	-
	Angular (degree)	-
Resolution	**Positional (mm)**	0.015
	Angular (degree)	-
Maximum marker Latency (ms)		<0.5
Maximum Working Volume (m³)		190 (with 1 tracker)
Price – minimal configuration		US$ 61,000

2.5.7 PhaseSpace Optical Motion Capture (PhaseSpace)

The PhaseSpace[8] Optical Motion Capture system with the recently released *Impulse* technology (2005), uses active LED markers with an innovative technique: each marker has its own digital identification (ID), which modulates its brightness.

[8] PHASESPACE. PhaseSpace Optical Motion Capture: datasheet – Available at <http://www.phasespace.com/ Documentation/PhaseSpaceOpticalMotionCapture.pdf>. Last accessed on December 16th, 2005.

Figure 19 shows the utilization of a PhaseSpace system for tracking full body motions in a typical application in the entertainment industry.

CCD cameras with a resolution of 3,600 x 3,600 pixels are used. Using interpolation algorithms, 30,000 x 30,000 sub-pixel resolution is obtained. The cameras work at 480 Hz frame rate at full resolution and have a maximum lateral field of view of 60 deg.

Each LED marker is modulated by a microcontroller between 80% bright and 100% bright in a pattern from 4 to 16 bits with the 100% brightness representing a binary one and the 80% brightness representing a binary zero. The vendor offers as default solution a wireless module for the markers to allow untethered operation.

This pattern varies for each marker, requiring the hardware and software to track the marker over several frames to establish the ID of the marker. This is both an advantage and a requirement of the high frame rate, in that the marker will not move much during the 2 ms between frames. After an ID has been established, it is validated in each frame. This way, just the next frame is enough to validate it, instead of a full 16 frame set.

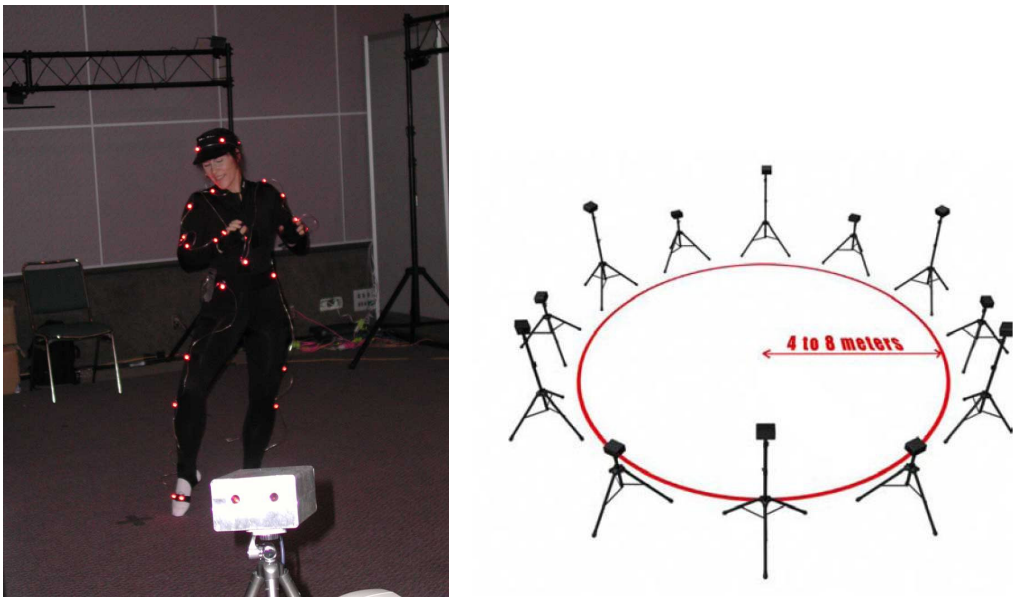

Figure 19 – Use of PhaseSpace System for Tracking Full Body Movements (left); Typical Setup with 12 Cameras capable of tracking within 600 m³ (right).

This feature solves a huge problem in optical tracking systems: how to deal with occlusions. Since every marker has its own ID, even if one or more markers are not visible, the system knows instantly which markers are not being seen. Of course correspondence problem is automatically solved as well. This strategy is very similar to the one used by the Visualeyez VZ4000 system. In optical tracking systems with passive markers complex algorithms must be employed in order to solve occlusion and correspondence problems.

A predictor-corrector algorithm is also used to predict the next expected position and validate that a marker indeed appeared where expected with the right ID value. This feature accelerates the processing.

On-board processing is present in the cameras, correcting lens distortion and performing tasks, e.g. threshold filtering, and also in the LEDs modules, where the ID's modulation is performed. Cameras have a custom communication interface and are connected through custom hubs to a central server, where information from all cameras is gathered together, processed and delivered to the VR/AR application.

A great advantage of this system is the robustness to light conditions due to the modulated LED IDs, because differential calculation can be used to extract the position of a marker even under high intensity ambient lighting. The system uses an adjustable shutter whose opening time depends on ambient light intensity. This intensity is periodically measured every 2,080 µs for a duration of 80 µs. Due to this feature the system is able to track active markers outdoors including in direct sunlight.

The system's overall update rate varies from 30 Hz (maximum 128 markers) up to 480 Hz (maximum 20 markers), depending on the marker configuration.

Positional accuracy provided by the vendor is 0.5 mm (mean absolute error) at a 5 m or 0.2 mm at a 1 m distance between tracked object and cameras. Precision data are not provided by the vendor. The positional resolution is given in Table 8 depending on the distance and the angle between tracked object and one camera, where it can be observed that the worst positional resolution of the system is 1.3 mm. Angular resolution is not provided.

Table 8 – Positional Resolution (mm) of PhaseSpace System according to Distance and Angle to One Camera

Distance from Camera (m)	Angle to Center of one Camera (deg)			
	5	**10**	**15**	**60**
0.1	0.0011	0.0022	0.0033	0.0131
1	0.0109	0.0218	0.0327	0.1309
10	0.1091	0.2182	0.3272	1.3090

With multiple cameras it is possible to expand the working volume up to several hundred cubic meters, as informed by the vendor. Maximum tracking distance is about 10m between camera and tracked object.

A typical configuration with 12 cameras, able to track within a circle with radius up to 8 m (total area around 200 m²), is sketched in Figure 19. Considering a maximum height of 3 m, the maximum working volume reaches approximately 600 m³. The price for a complete system in a typical configuration, composed of 16 cameras, able to track a maximum of 128 markers at 30 Hz, is US$ 104,000. Each camera itself costs US$ 5,000. Table 9 summarizes the properties of the PhaseSpace system.

Table 9 – Summary of Main Properties of PhaseSpace System

Maximum Update Rate (at given resolution)		480 Hz @ 3,600 x 3,600 (max. 20 markers) 30 Hz @ 3,600 x 3,600 (max. 128 markers)
Accuracy	**Positional (mm)**	0.5
	Angular (deg)	-
Precision	**Positional (mm)**	-
	Angular (deg)	-
Resolution	**Positional (mm)**	1.3 (worst-case)
	Angular (deg)	-
Maximum Latency (ms)		<10
Maximum Working Volume (m^3)		600 (at least)
Price – typical configuration		US$ 104,000

2.5.8 Other Systems

There are other systems which are not listed in this work but that have similar properties and working principles to the ones presented here. For a detailed description see (BUAES, 2005).

For instance, the Optotrak[9] Certus system, from Northern Digital Inc., the CodaMotion system, from Charnwood Dynamics Ltd.[10], and the ReActor 2 system, from Ascension Technology Corporation[11], use active infrared markers and are thus very similar to the Visualeyez VZ4000 and PhaseSpace systems (sections 2.5.6 and 2.5.7). The Qualisys[12] Motion Capture System uses passive infrared markers just as the ARTtrack, Eagle/Hawk, Vicon MX and SMART systems (sections 2.5.1, 2.5.3, 2.5.4 and 2.5.5).

2.6 RESEARCH, NON-COMMERCIAL MARKER-BASED OPTICAL SYSTEMS

In this section the marker-based optical tracking systems developed within research institutions and that have not yet been commercially explored are listed. Information for this section has been collected, among other sources, from (SANTOS, 2005) and (ZHOU, 2004).

---

[9] Optotrak Certus Tracking System Website by Northern Digital Inc. Available at <http://www.ndigital.com/ certus.php>. Last accessed on December 16th, 2005.

[10] CHARNWOOD DYNAMICS LTD. Codamotion system. Product information available at <http://www.charndyn.com/Products_Intro.html>. Last accessed on December 16th, 2005.

[11] ASCENSION TECHNOLOGY CORPORATION. ReActor 2: Digital Active-Optical MoCap System. Product information available at <http://www.ascension-tech.com/ products/reactor.php>. Last accessed on December 16th, 2005.

[12] QUALISYS AB. Qualisys ProReflex Motion Capture Unit: Product Information Brochure, available at <http://www.qualisys.com/images/ ProReflex.pdf>. Last accessed on December 16th, 2005.

2.6.1 ARToolKit

ARToolKit (KATO, 2000) is actually not a complete tracking system but a free software library that can be used to calculate camera position and orientation relative to physical markers in real time. This enables an easy development of a wide range of augmented reality applications. Currently only one camera is supported for the tracking application (one camera tracking). ARToolKit can only track square targets whose patterns (including dimensions) are previously known. Thus, ARToolKit is a marker-based optical tracking system.

Although ARToolKit has been originally developed as a non-commercial solution, commercial exploration can be made and for that reason it was included in this survey.

A typical application can be built with a regular web camera and a PC running ARToolKit, interconnected for instance by a Universal Serial Bus (USB) 2.0 or 1394 Firewire interface.

The working principle is as follows. First the live video image (Figure 20) is transformed into a binary image (black or white) based on a lighting threshold value. This image is then searched for square regions. ARToolKit finds all the squares in the binary image, many of which are not the tracking markers. For each square, the pattern inside the square is captured and matched against pre-trained pattern templates. If there is a match then ARToolKit has found one of the AR tracking markers.

ARToolKit then uses the known square size and pattern orientation to calculate the real position of the video camera relative to the physical marker. A 3 x 4 matrix is filled with the video camera real world coordinates relative to the card (marker). This matrix is then used to set the position of the virtual camera coordinates. Since the virtual and real camera coordinates are the same, the generated computer graphics precisely overlay the real marker (Figure 20). A detailed description of tracking information calculation and 3D reconstruction algorithms is given in section 3.4.1.1. The Open Graphics Library (OpenGL) Application Programming Interface (API) is used for setting virtual camera coordinates and drawing virtual images.

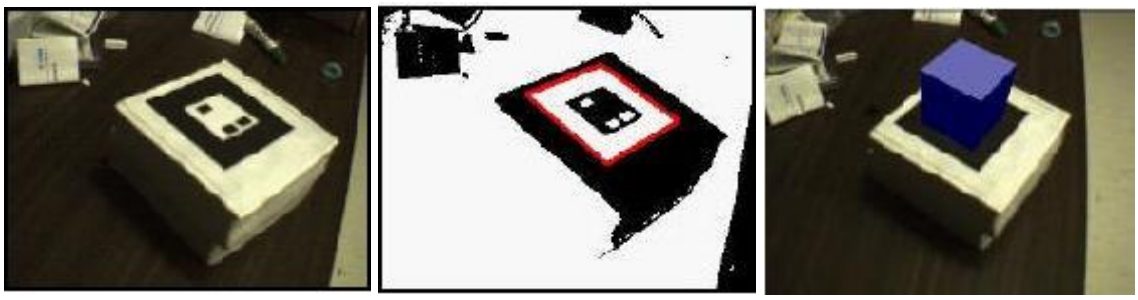

Figure 20 –Processing Sequence of ARToolKit, from Left to Right: Input Video, Thresholded Video, Virtual Overlay

The performance of the ARToolKit algorithms depends drastically on the hardware used. In (KATO, 1999) the creators of the algorithm carried out some performance tests, providing only preliminary results.

In (MALBEZIN, 2002) experiments were conducted with a 1394 Firewire Pyro WebCam, which has the following features: 640 x 480 pixel resolution, 30 Hz, and uncompressed video data transmission over IEEE 1394 Firewire interface. The setup for those experiments was a black and white 20 x 20 cm marker pattern fixed at central position, around which a camera performs circular movements with progressive increasing radius. Error measurements are made only in X and Y directions. The results show that ARToolKit works in an acceptable way within a maximum distance of 2.5 m between marker and camera. Figure 21 shows mean error values in position estimates in X and Y directions.

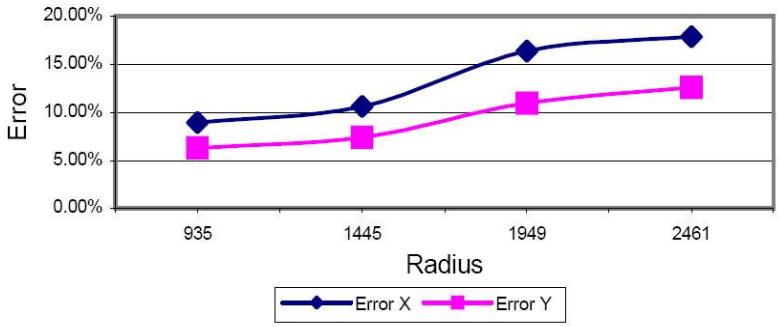

Figure 21 – ARToolKit: Mean Error Values (Accuracy) in Position Estimation in X and Y Directions (MALBEZIN, 2002)

The system's accuracy, here considered as maximum error values obtained during experiments, is shown in Table 10. It has been detected that system's accuracy depends on the angle between camera and pattern alignment, as can be seen in Figure 22.

Table 10 – ARToolKit: Maximum Error Values up to 2.5 m between Camera and Marker (MALBEZIN, 2002)

Radius (m)	1.0	1.5	2.0	2.5
Error (mm)	±14	±18	±22	±27

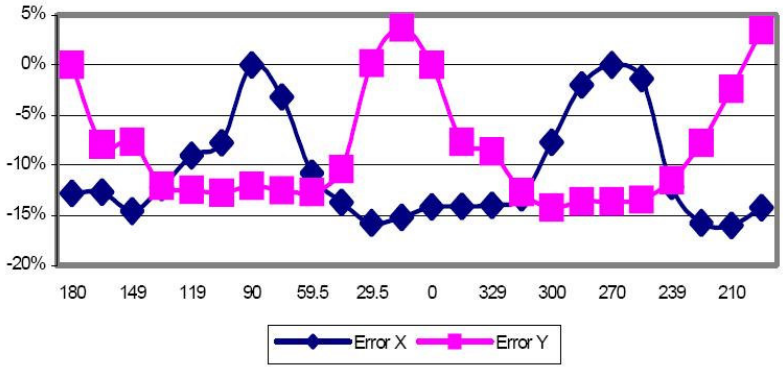

Figure 22 – ARToolKit: Maximum Error Values with Variation in Angle between Camera and Marker during Rotation (MALBEZIN, 2002)

The update rate depends on the hardware (camera and computer) used. For instance in (ABAWI, 2004) experiments were conducted with a Philips PCVC750K Webcam, using a resolution of 640 x 480 at a frame rate of 15 Hz which can be considered a typical setup for ARToolKit. Figure 23 shows the resulting pictures with overlaid virtual objects after processed by ARToolKit.

Latency times depend on the hardware used. Since ARToolKit is not based on on-board processing but on PC processing, it is expected that latency is larger than in systems with embedded processing on camera, which only transmit already filtered information to the computer where the application runs.

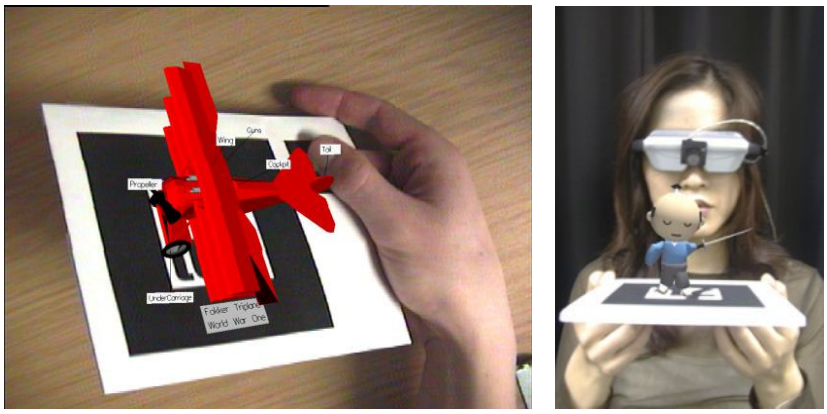

Figure 23 – ARToolKit: Virtual Objects over Real Table-tops (KATO, 2000)

The working range depends on the configuration used (hardware, operating conditions and system parameters). Considering the experiments made in (MALBEZIN, 2002), the maximum acceptable working range would be a circle with radius 2.5 m around the marker to be tracked. Considering a maximum height of 3 m these dimensions result in a maximum working range of 59 m³. This is rather small in comparison with other systems previously presented.

ARToolKit has no costs for research purposes, which is one of the reasons why it is widely used in many AR projects around the world. For commercial use the authors[13] should be contacted.

2.6.2 PTrack

PTrack (SANTOS, 2005) is an outside-in single-camera marker-based optical tracking system, based on a one-camera tracking algorithm. It was initially developed to work with the ART cameras used on the ARTtrack system. The cameras provide the algorithm with the 2D position of the markers detected on the camera sensor. Figure 10 shows the cameras used with PTrack. The algorithm processes the data and estimates the 3D position of a label – a set of 6 markers, 4 of them composing a square, one placed on an edge of the square and another one disposed within the area

[13] ARToolkit. Website available at <http:// www.hitl.washington.edu/artoolkit/>. Last accessed on December 16th, 2005.

defined by the square. Figure 24 shows a typical label design as well as the definition data for this label, containing the 2D coordinates of each marker.

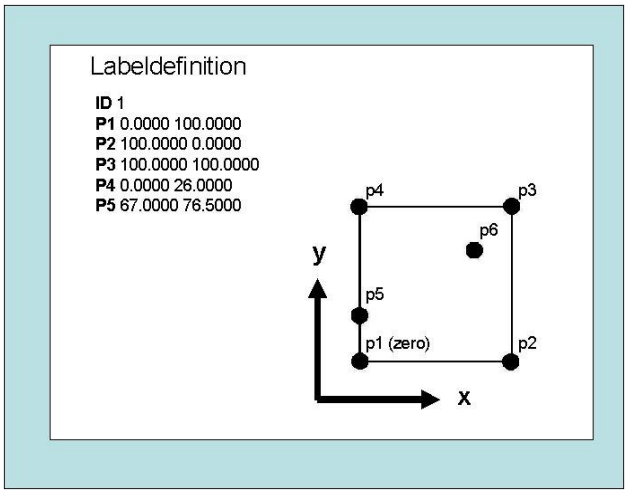

Figure 24 – PTrack's Label Definition Data and Typical Design (SANTOS, 2005)

The algorithm uses a "divide-and-conquer" strategy during 2D processing of image space to detect potential projections of labels. Figure 25 shows the processing pipeline of PTrack. First the 2D image space is segmented using a quad-tree approach, which divides the image frame in smaller regions. Then in each region a radar sweep algorithm around the center of gravity of the set of 2D points is used to detect a potential label. The innovation of the algorithm lies in the reconstruction from 2D image space feature points to 3D camera space feature points. In this approach the projection of a potential label converges rapidly to the correct rotational pose and then is scaled to the correct translational position in camera space with high translational and rotational accuracy. Figure 26 shows PTrack stand-alone application plotting 2 detected labels.

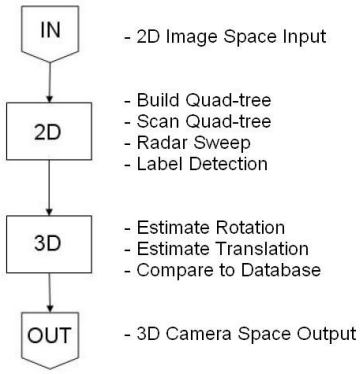

Figure 25 – Description of PTrack's Processing Pipeline (SANTOS, 2005)

Due to the importance of PTrack for this work and the later references to the algorithm, a detailed description of each step in the processing pipeline of PTrack is given in section 3.4.1.2.

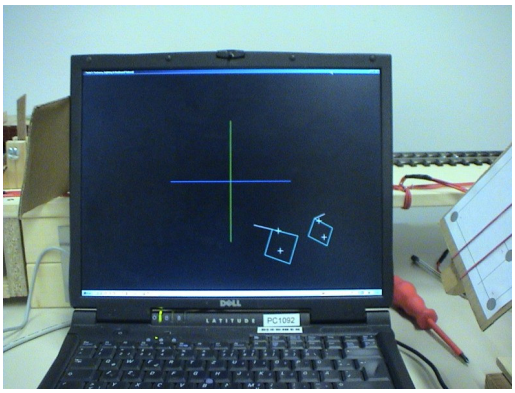

Figure 26 – PTrack Stand-alone Application detecting 2 Labels (SANTOS, 2005)

PTrack has been designed and developed in a modular way, so that it can be adapted to work with cameras other than the ART ones. The software module that interfaces with the cameras must be changed. Figure 27 shows PTrack's system architecture.

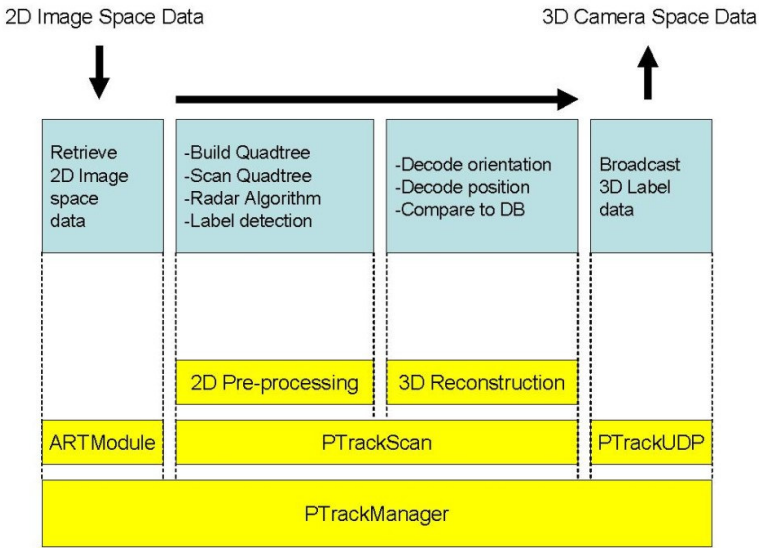

Figure 27 – PTrack System Architecture (SANTOS, 2005)

PTrack was originally written to work with ART cameras. In that setup the ART module connects the processing pipeline with the cameras and is responsible for synchronized acquisition of a list of 2D marker positions, as well as delivery of that list to the algorithm, which runs as an independent thread. Both 2D processing and 3D reconstruction take place in PTrackScan module. Broadcasting of tracking information to external modules, through the OpenTracker[14] framework, is executed in PTrackUDP module. PTrackManager module coordinates and synchronizes all parts and runs as another thread.

[14] OpenTracker, Unified Abstract Tracking Layer, website, 2005. Available at <http://www.studierstube.org/ opentracker>. Last accessed on December 16th, 2005.

PTrack has been tested for accuracy and precision using the same testbed and test sequences used for the new tracking system presented in this work, as explained in chapter 5. In translation experiments PTrack has obtained, under full illumination conditions, accuracy (average error) of 1.7 mm and precision (standard deviation) of 1.5 mm. Figure 28 shows measured and real distances. In rotation experiments, under full illumination, PTrack's overall accuracy reaches 2.9 deg. Precision has been tested under specific conditions, namely with different angles of attack. For 40 deg angle of attack (defined as the angle at which the target or label is tilted from the position in which it is facing the camera) the measured overall precision under full illumination is 1.3 deg. Figure 29 shows an overview of accuracy results for PTrack.

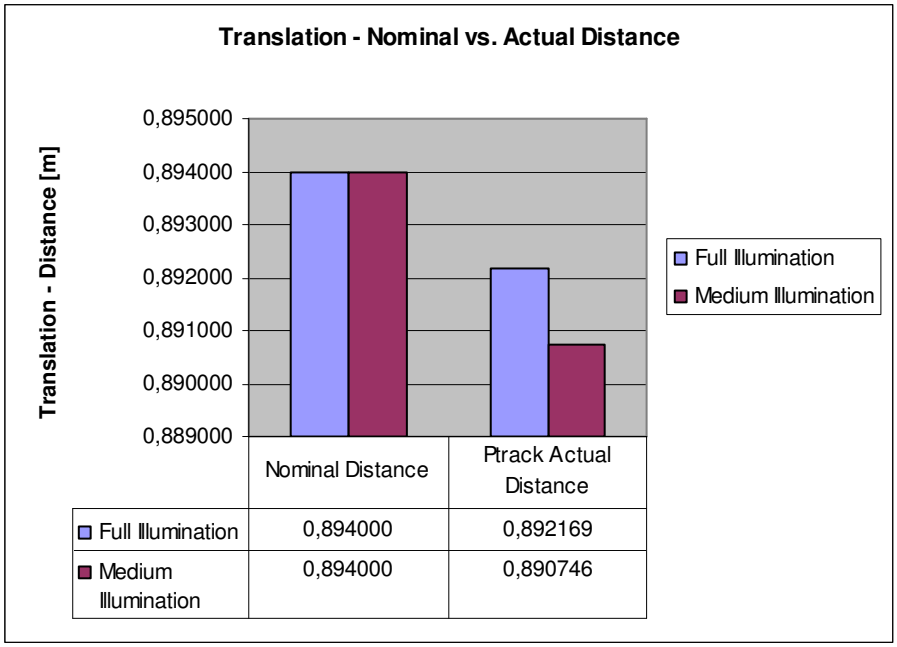

Figure 28 – PTrack Translation Test Results – Nominal *versus* Actual Distance (SANTOS, 2005)

For more details on accuracy and precision tests, see (SANTOS, 2005).

The working volume of PTrack using ART cameras is defined by the minimum and maximum distances from markers to the camera, where the marker can still be correctly detected. Minimum distance is always around 0.5 m. Maximum distance ranges from 4 m (small cameras) to 10 m (bigger cameras). Considering an horizontal FoV of 57.9 deg and a vertical one of 45 deg, the maximum theoretical working volume using 1 camera varies from 4.25 to 11.5 m³, with varying camera range from 4 to 10 m.

PTrack's update rate depends on the computer where it runs and on the number of markers visible in the scene. For instance running on a Pentium Centrino 2 GHz 1 GB Random Access Memory (RAM) and with only 1 label (6 markers) in the scene, update rate reached 57 Hz, almost the full frame rate of the camera (60 Hz). Latency is directly related to the update rate, since next frame can only be processed after finishing the last one.

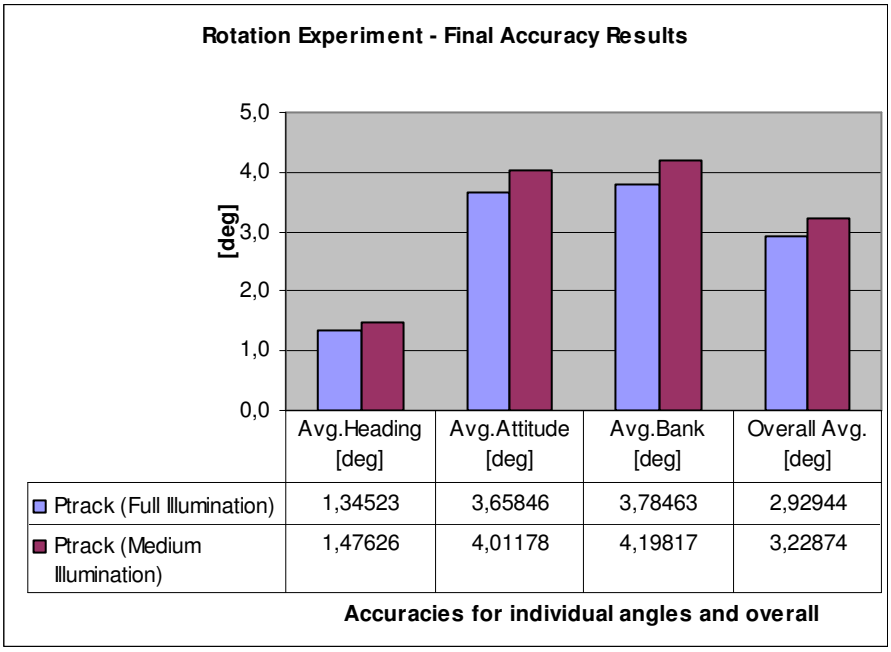

Figure 29 – PTrack Rotation Test Results – Overall Accuracy (SANTOS, 2005)

2.6.3 Stereo Tracker from VRVis

Ribo *et al.* (RIBO, 2001a) presented an optical stereo tracking system for AR/VR applications. Although VRVis GmbH. is a private research institution, this system is yet considered a research system, due to inexistence of commercial exploration.

The system implements a two-camera stereo configuration and also uses infrared light. The used cameras allow up to 30 Hz update rate. Retro-reflective markers are illuminated by infrared light strobes. Their 2D position is detected by both cameras and then epipolar geometry (to solve correspondence problem) and 3D reconstruction algorithms (see section 3.5) are used to extract the 3D position from each marker. Using workspace constraints like distances and angles between a set of markers, the system is capable of identifying artifacts and also calculating their orientation. At least 3 markers are needed to obtain 6 DoF information.

The authors claim that the system has around 1 mm accuracy in translation and 0.1 deg in orientation, although no detailed information was provided regarding tests sequences or used testbeds. Nevertheless a demonstration video which can be downloaded in the system's website shows it is fast and stable enough (low jitter, precision at least reasonable) to be used with AR/VR applications.

An interesting aspect of this system is the use of linear prediction algorithms to avoid excessive jitter in the measurements, based on the knowledge of movements of the markers along a sequence of image frames.

Figure 30 shows the camera used in the system and a user with a typical set of artifacts: shutter-glasses for active stereo projection, pen and transparent pad. Each artifact is equipped with markers to allow detection by the system.

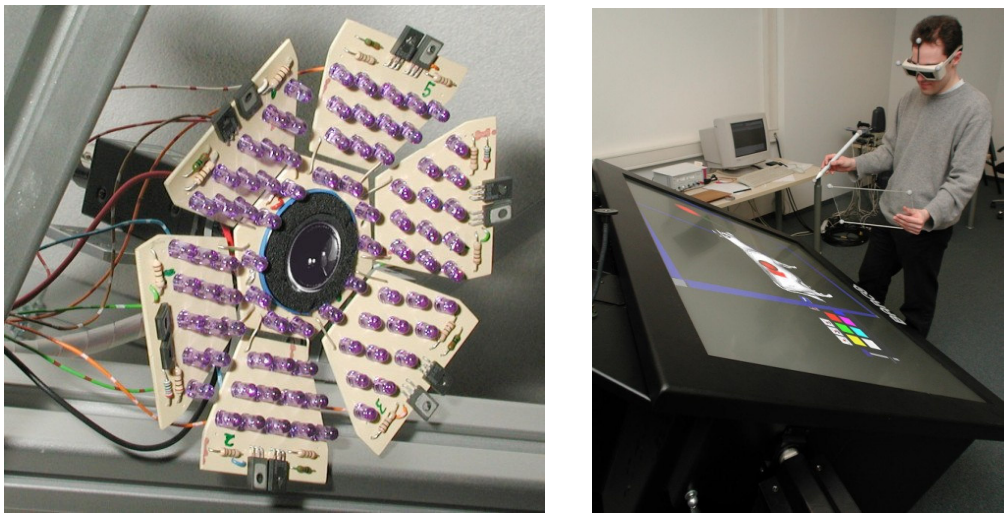

Figure 30 – Camera of VRVis Stereo Tracker (left); Typical Usage Scenario (right)

Working volume information is also not provided by the authors, but based on demo videos and pictures it can be estimated that it is at least 4 m³, the maximum limit being defined by the reach of the cameras for marker detection.

No reference to the costs of the used hardware (cameras, flash strobes, additional components) has been made by the authors.

2.6.4 Other Systems

Chung *et al.* (CHUNG, 2001) present POSTRACK, a low cost motion capture tracker based on detection of large infrared markers. The system achieves 15 Hz update rate and requires 4 affordable grayscale cameras in order to track up to 5 markers, but has high jittery in position estimation. POSTRACK uses standard stereo algorithms and has translational accuracy and precision around a few centimeters. OpenCV calibration routines are used.

Tao and Hu (TAO, 2003) present a mixed marker-based and markerless solution to analyze human movements for rehabilitation treatment. Three Qualisys[15] cameras are used to acquire images, which build three different combinations of two cameras each. Stereo relations are then used to extract 3D position of retro-reflective markers attached to a limb, like arm or leg. Using model-based tracking algorithms, the trajectory described by the limb can be calculated.

Dorfmüller (DORFMÜLLER, 1999) presents a system very similar to the VRVis tracker. Two infrared cameras in stereo configuration are used, as well as spherical retro-reflective markers. The novelty in this case is the use of a very simple

[15] QUALISYS AB. Qualisys ProReflex Motion Capture Unit: Product Information Brochure, available at <http://www.qualisys.com/images/ ProReflex.pdf>. Last accessed on December 16th, 2005.

but efficient adaptive calibration algorithm which requires only one moving point visible by both cameras. The system achieved 6 mm translational accuracy with update rate of 25 Hz.

Mulder and van Liere (MULDER, 2002) developed the Personal Space Station (PSS), equipped with a marker-based stereo head tracker (MULDER, 2003), where even new correspondence and recognition algorithms have been developed for stereo tracking (VAN LIERE, 2003). The PSS also uses infrared retro-reflective markers and stereo configuration. It is a mirror-based environment, where the user sees, through a mirror, reflected images originally generated by a Cathode Ray Tube (CRT) monitor. The PSS system differs from other approaches in that, due to the goal of tracking within a so called near-field virtual environment, it is intended to have only a limited small working range, where translational accuracy of 6 mm is achieved, yield accurate tracking information. The complete hardware setup of PSS costs around EUR 13,000.

2.7 COARSE COMPARISON AMONG SYSTEMS

All systems listed in sections 0 and 2.6, with exception of the HiBall-3100, have outside-in topologies. This configuration is more suitable for indoor usage, because the use of fixed sensors instead of moving ones makes image processing tasks easier.

The systems can be categorized according to the type of markers used: passive and active. The systems with passive markers usually use infrared light with retro-reflective markers, e.g. ARTtrack, Vicon MX and PTrack. ARToolKit is an exception, using labels in daylight region of spectrum. Systems with active markers – like HiBall-3100, Visualeyez VZ4000 and PhaseSpace – also usually work with infrared light. The hardware of active markers' systems is more complex, because it demands high frequency synchronization between sensors (cameras) and emitters (markers). Systems with active markers are more frequently used for motion capture purposes. Systems with passive markers are more frequent, although in the last years use of active markers has grown largely.

The reason for using infrared is the higher robustness obtained against optical noise, without disturbing system users (due to the fact that infrared is invisible to the human eye). The update rate usually depends on application requirements: commercial systems have at least 60 Hz of overall update rate, and some maximum rates can reach 2,000 Hz under special conditions; academic systems have lower update rates, with the maximum rates reaching 60 Hz (PTrack using ART Cameras running on a high performance PC). Higher update rates of commercial systems are needed to meet quality requirements, allowing systems to be used professionally.

With exception of ARToolKit, which has worse values, all systems have accuracy and precision values below 1 mm for position and 1 degree for orientation. Resolution values are at most also 1 mm and 1 degree. These values seem to be sufficient for practical professional use.

Working range highly depends on the number of sensors (cameras) used. Commercial systems have higher working ranges, basically due to high modularity of

the systems, which can be scaled up as much as needed. Obviously prices can only be analyzed and compared among commercial systems.

Table 11 shows a coarse comparison - because the conditions under which the values have been measured are not exactly the same - only among commercial systems, with their main properties.

Table 11 – Coarse Comparison among Commercial Systems

System	System Update Rate (Hz)	Accuracy		Precision		Resolution		Max. Latency (ms)	Max. Working Volume (m^3)	Price (EUR)
		Pos (mm)	Ang (deg)	Pos (mm)	Ang (deg)	Pos (mm)	Ang (deg)			
ARTtrack	60	0.4	0.12	-	-	0.06	0.03	40	300	30,000
HiBall-3100	2,000	0.4	0.02	-	-	0.2	0.01	1	unlimited	24,000
Vicon Tracker	160	0.1	0.15	-	-	0.2	0.2	10	1,000	35,000
Eagle/Hawk	166	2.77	0.52	0.36	0.13	-	-	6	816	65,000
SMART	120	1	-	0.5	-	-	-	10	300	45,000
Visualeyez VZ4000	87	0.6	0.007	-	-	0.015	-	0.5	190	47,000
PhaseSpace	480	0.5	-	-	-	1.3	-	10	600 at least	80,000

It can be seen that commercial systems are usually very expensive, what precludes a wider dissemination of these systems.

2.8 System Specification for a New Tracking System

As seen in the previous section, typical marker-based optical tracking systems use cameras in infrared region of spectrum and with passive markers; have update rate of at least 60 Hz (and corresponding latency); precision and accuracy values around 1 mm for position and 1 deg for orientation; working range scalable by using more modules of the system; and very high costs (more than EUR 20,000 for a basic configuration).

Looking back at the disadvantages mentioned in Section 2.4 about general optical systems, it becomes clear that the line-of-sight problem is solved by including additional sensors (cameras) to the system configuration. The sensitivity to presence of noise is drastically reduced by using infrared, so robustness is increased. The main problem still remaining are the high costs of such systems.

However options such as ARToolKit are very cheap – it can work with webcams and common PCs –, but lack of precision and speed. The stereo tracker from VRVis has enough precision, but the update rate does not meet the requirements for professional use.

This scenario was the main motivation for this work: to build a low cost, marker-based optical tracking system which fulfills the basic requirements to be used professionally, such as the commercial systems in section 0, but has a lower cost than these systems, collaborating to elevate dissemination of this type of tracking systems.

The proposed target update rate for the new system is 60 Hz, with a minimum of 25 Hz as acceptable value, in order to be at least as fast as already existing solutions. This minimum update rate is sufficient for most of proposed utilizations for the system, i.e. indoor AR/VR applications.

Latency is related to update rate and to the use of pipelined calculations, if any. Target values for precision and accuracy are 1 mm for translation and 1 deg for orientation.

For indoor use in AR/VR applications, excluding utilization for motion capture, a working range able to track within a small room is considered sufficient. For instance, the system from VRVis has a reasonable 4 m³ working range, which is also set as target working range for the new system. The costs of the new system should be at least 10 times lower than the smallest price among all commercial systems. With that in mind, EUR 2,000 was established as maximum target cost for the new system.

Based on target requirements and on tracking systems surveyed in sections 0 and 2.6, a grading method can be established in order to evaluate the new system. This method is presented in Table 12. Target values are considered “very good”, intermediate values are considered “good”, minimum acceptable values are considered “satisfactory”, and deficient values are considered “insufficient”.

The one-camera tracking topology has been chosen for this system because it is a research topic not so exhaustively explored as the stereo tracking system and, in addition, it complements the implementation of the PTrack algorithm, already developed within Fraunhofer IGD. That is also the reason why that algorithm is chosen as basis for this work.

Table 12 – Grading Method for Performance Evaluation of a New System

Classification		Very Good	Good	Satisfactory	Insufficient
Update Rate (Hz)		≥ 60	≥ 45	≥ 25	< 25
Precision and Accuracy	Translational (mm)	≤ 1.0	≤ 5.0	≤ 10.0	> 10.0
	Rotational (deg)	≤ 1.0	≤ 5.0	≤ 10.0	> 10.0
Working Range (m³)		≥ 4	≥ 3	≥ 2	< 2
Costs (EUR)		≤ 2,000	≤ 3,000	≤ 4,000	> 4,000

As an extended function of the system and simultaneously a case study, a multiple camera tracking system should be implemented. This specific solution has been chosen because it is a field of research still partially unexplored in comparison to others. This would show one of many possible applications for the new modules to be developed as e.g. hardware, image pre-processing, calibration and the tracking algorithm itself.

A new tracking system which meets these requirements can be a valuable contribution for the dissemination of this type of trackers. Other benefits of such system are the adaption of PTrack, transforming it in a tracking algorithm compatible with theoretically all cameras which can have images transferred to a computer, and the implementation of a scalable multiple-camera tracking system based on one-camera modules.

3 THEORETICAL PRINCIPLES

In this chapter basic definitions and background concepts needed for development of the system implemented in this work are presented and explained. Initially, camera basic concepts are illustrated. As stated in the section 2.8, the new system to be developed will be a marker-based optical one-camera tracking system. Image pre-processing algorithms are needed in order to extract the position of markers. Calibration procedures are used to obtain camera parameters. A one-camera tracking algorithm is needed to obtain the pose information of labels, which must be formatted and forwarded to AR/VR interaction frameworks. For the case study sensor fusion techniques are required.

Additional techniques, which are not necessarily used in the implemented system, are also explained as a supplement to help users to understand the entire system.

3.1 CAMERA BASIC CONCEPTS

This section describes the basic concepts of cameras related to optical tracking systems: camera model, calibration algorithms, coordinate systems and transformations. (TRUCCO, 1998) and (SHAH, 1997) contain clear explanations regarding these issues and are the major sources for this section.

3.1.1 Basic Optics

Image focusing is a main issue among the basic optics concepts. A simple image capturing device to be used can be a pin-hole camera (Figure 31), i.e. camera's aperture reduced to only one point, what results in very sharp images but gives no flexibility regarding exposure time or focal length adjust, or an optical system composed of lenses.

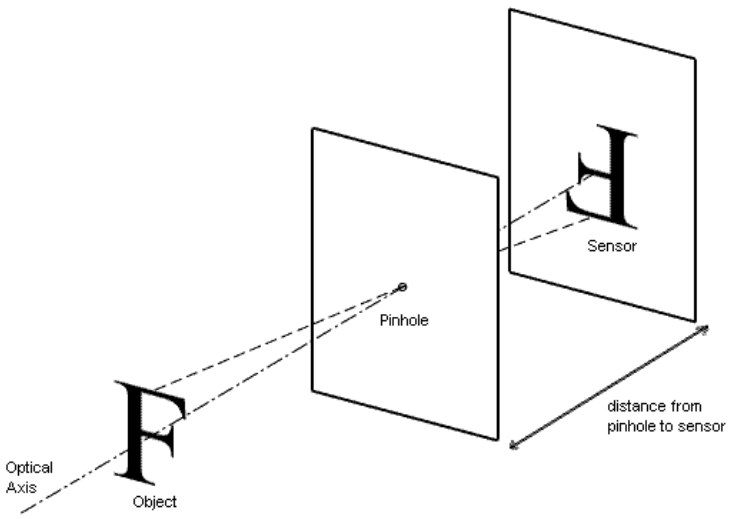

Figure 31 – Pinhole Camera Example

Thin lenses are a very basic though important concept for this work. Thin lenses have their properties defined by the fundamental equation

$$\frac{1}{S_1}+\frac{1}{S_2}=\frac{1}{f} \tag{4}$$

where f is the focal length of the lens and the measures S_1 and S_2 are described in Figure 32. In general, a real lens may have two different focal lengths, because the curvatures of its two surfaces may be different, but for simplification purposes only one focal length measure is considered. Equation (4) implies that scene points at different distances from the lens come in focus at different image distances. The optical lens systems of real cameras are designed in such a way that all points within a given range of distances have their images formed on or close to the image plane, and therefore acceptably in focus. This range is called the *depth of field* of the camera.

Due to construction defects the aperture of a lens may prevent light rays from reaching the lens peripheral points, so the effective diameter of a lens is different from the physical diameter. The lens effective diameter and the focal length determine the field of view of the lens, which is the angular measure of the portion of 3D space actually seen by the camera. The field of view of lenses is defined by the equation

$$\tan(\omega)=\frac{d}{2f} \tag{5}$$

where ω is the field of view angle, d is the effective lens diameter and f is the focal length.

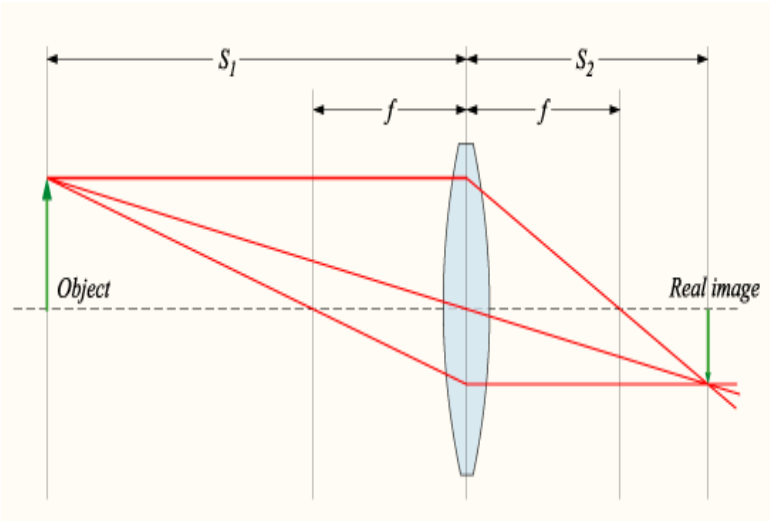

Figure 32 – Image Formation of a Thin Lens

3.1.2 The Perspective Camera

In order to establish a relation between scene points with their corresponding image points, the geometric projection of the camera must be modelled. The most common geometric model of an intensity camera is the *perspective* or *pinhole* camera model, shown in Figure 33. The model consists of a plane π, the *image plane*, and a 3D point **O**, which is the *center of projection*. The distance between π and **O** is the focal length *f*. The line through **O** and perpendicular to π is the optical axis, and **o**, which is the intersection between π and the optical axis, is called *principal point* or *image center*. The projection **p**, on the image plane, of the point **P** is obtained by the intersection of the straight line through **P** and **O** with the image plane π. The 3D reference frame in which **O** is the origin and the plane π is orthogonal to the Z axis is called the *camera frame*, and represents the camera's coordinate system, where the points **p**[x, y, z] and **P**[X, Y, Z] are located.

The relation between the coordinates of points **p** and **P** is given by the equations

$$\begin{aligned} x &= f\frac{X}{Z}, \\ y &= f\frac{Y}{Z} \end{aligned} \tag{6}$$

which are the fundamental equations of the perspective projections. These equations are nonlinear because of the factor 1/Z, and do not preserve either distances between points (not even common scaling factors) or angles between lines, although they do map lines into lines and preserve parallelism between lines perpendicular to the optical axis.

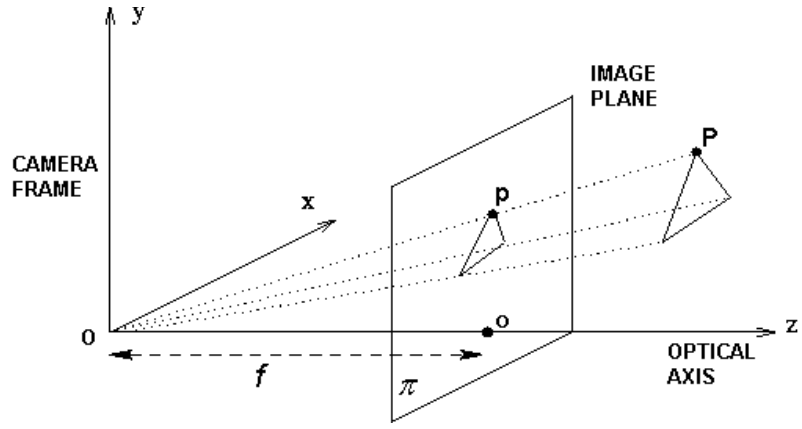

Figure 33 – The Pinhole or Perspective Camera Model

Note that, in the camera frame, any image point has the third component always equal to the focal length, because the equation of the plane π is $z = f$, which leads to the representation **p**[x, y] instead of **p**[x, y, z].

3.1.3 Camera Intrinsic Parameters

Any computer vision system intending to reconstruct the 3D position of objects or points in space needs expressions which relate the coordinates of image points with the *pixel coordinates*, which are the only one directly available from the image, and the camera reference frame with respect to some other coordinate system – known as *world reference frame*. The first relationship is given by the camera intrinsic parameters, the latter by the camera extrinsic parameters.

In order to link pixel coordinates of an image with the corresponding coordinates in the camera reference frame, the camera intrinsic parameters need to characterize the optical, geometric and digital properties of the camera. Considering the pinhole camera model, these parameters are:

- the focal length *f*, which defines the perspective projection;
- the ones which define the transformation between camera frame and pixel coordinates; and
- the geometric distortion introduced by the lens.

The focal length *f* is already explicit in (6). To find the second set of intrinsic parameters, we must link the coordinates (x_{im}, y_{im}) of an image point in pixel units, the coordinates in the *image reference frame*, with the coordinates (x, y) of the same point in the camera reference frame. This relation is established by (7), considering that the sensor array consists of rectangular photosensitive elements and ignoring optics distortion, where (o_x, o_y) are the coordinates in pixel units of the image center, also called *principal point*, and (S_x, S_y) express the effective size of the pixel, in millimeters, in the horizontal and vertical directions, respectively. The image and camera reference frames have opposite orientation, therefore the sign changes in (7).

$$\begin{aligned} x &= -\left(x_{im} - o_x\right) S_x \\ y &= -\left(y_{im} - o_y\right) S_y \end{aligned} \tag{7}$$

Equation (6) can be rewritten as matrices multiplication using homogeneous coordinates, resulting in

$$\begin{bmatrix} s \cdot x \\ s \cdot y \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{8}$$

where $s \neq 0$ is a scale factor. Using (7) and (8) and considering $f_x = f / S_x$ and $f_y = f / S_y$, the following expression can be obtained,

$$\begin{bmatrix} s \cdot x_{im} \\ s \cdot y_{im} \\ s \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{9}$$

where the scale factor s has value Z. In short notation (9) can be rewritten as

$$\tilde{u} = M_{\text{int}} \cdot \tilde{W} , \tag{10}$$

where $\tilde{u}$ is the vector of image pixel coordinates and $\tilde{W}$ is the vector of world coordinates, both in homogeneous notation. M_{int} is the perspective projection matrix which takes into account camera intrinsic parameters. Thus, a camera can be considered as a system that performs a linear projective transformation from a 3D projective space into a 2D projective space.

There are five camera intrinsic parameters, when not considering distortion introduced by lenses: f, o_x, o_y, S_x, S_y. However, only four separable parameters can be determined during calibration process (see section 3.3 for details), since there is an arbitrary scale factor involved in f and in the pixel size when using this matrix notation. Thus, it can only be solved for the relations f_x and f_y. The focal length f is usually defined based on the pixel size, which is a parameter provided by camera or sensor manufacturers.

The optics introduces image distortions, both tangential and radial, which become visible especially at the image's periphery. Figure 34 shows an example of an image with radial lens distortion and its undistorted corrected version. (TSAI, 1986) and other studies show that for practical use tangential distortions can be neglected and radial distortions must be considered only up to the second coefficient at most.

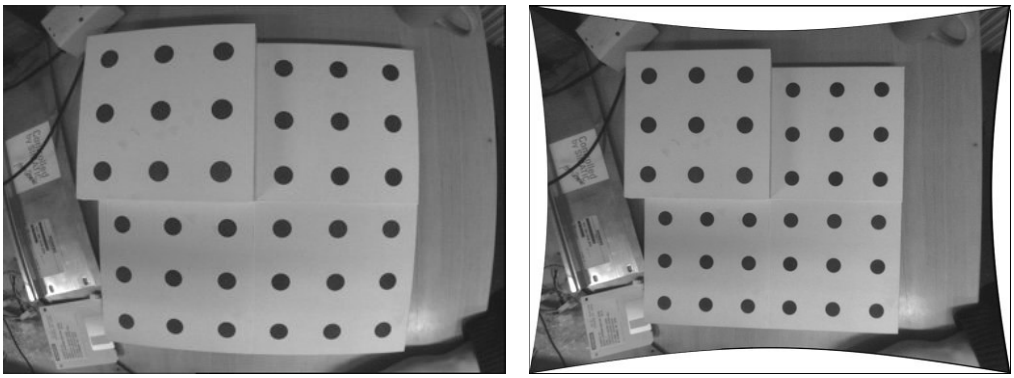

Figure 34 – Radial Lens Distortion: Original Image (left); Corrected Version (right)

Radial distortions can be modelled according to

$$\begin{aligned} x &= x_d \left(1 + k_1 \cdot r^2 + k_2 \cdot r^4\right) \\ y &= y_d \left(1 + k_1 \cdot r^2 + k_2 \cdot r^4\right) \end{aligned} , \tag{11}$$

where (x_d, y_d) are the coordinates of the distorted points, $r^2 = x_d^2 + y_d^2$ and k_1, k_2 are the first 2 coefficients for radial distortion. This distortion is a radial displacement of the image points, which is null at the image center and increases with the distance of the point from the image center. k_1 and k_2 are camera intrinsic parameters. Because they are usually very small, radial distortion is ignored whenever high accuracy is not required in

all regions of the image, which is not the case of a high precision optical tracking system.

3.1.4 Camera Extrinsic Parameters

The camera reference frame is often unknown or undesired, and a common problem is determining the location and orientation of the unknown camera coordinate system with respect to some known reference frame, using only information extracted from the image. Camera extrinsic parameters are defined as a set of geometric parameters which identify uniquely the transformation between the camera reference frame and the world reference frame.

This transformation is typically defined as a translation followed by a rotation. The translation is described by the 3D translation vector T, which specifies the relative positions of the origins of the two reference frames. The rotation is described by an orthogonal 3 x 3 matrix R which brings corresponding axes of the two coordinate systems onto each other.

Translation and rotation, combined, result in a 6 DoF transformation depicted in Figure 35 and shown in the expression

$$P_{cam} = R \cdot \left(P_w - T \right), \tag{12}$$

being P_{cam} a point P in the camera reference frame, P_w the same point in the world coordinate system,

$$T = \begin{bmatrix} T_1 & T_2 & T_3 \end{bmatrix}^T \tag{13}$$

and

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{14}$$

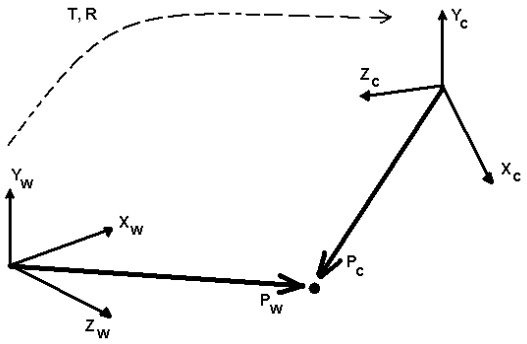

Figure 35 – Relation between Camera and World Coordinate Systems

Using homogeneous coordinates, equations (13) and (14) can be combined in only one matrix in order to represent the transformation of (12), resulting in

$$P_{cam} = M_{ext} \cdot P_w , \tag{15}$$

where

$$M_{ext} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -R_1^T \cdot T \\ r_{21} & r_{22} & r_{23} & -R_2^T \cdot T \\ r_{31} & r_{32} & r_{33} & -R_3^T \cdot T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{16}$$

being R_i, i=1, 2, 3, a 3D vector formed by the i-th row of the matrix R. M_{ext} contains the complete description of the transformation between the world and the camera coordinate systems.

3.1.5 Camera Projection Matrix

Combining the effects of intrinsic and extrinsic camera parameters is nothing more than at first transforming from 3D world coordinates to 3D camera coordinates and then to 2D image pixel coordinates. This can be done by applying not only M_{int} , as in (10), but also M_{ext}, from (16), to the 3D point in world coordinates, what can be written as

$$\tilde{u} = M_{\text{int}} \cdot M_{ext} \cdot \tilde{W} \tag{17}$$

or, making the homogeneous coordinates explicit,

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = M_{\text{int}} \cdot M_{ext} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}, \tag{18}$$

where x_{im}=x1/x3 and y_{im}=x2/x3 are the pixel coordinates. If the ratios (x1/x3) and (x2/x3) are incorporated as part of the transformation matrix, the following expression can be written,

$$\begin{pmatrix} x_{im} \\ y_{im} \\ 1 \end{pmatrix} = P \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}, \tag{19}$$

where P is the *camera projection matrix*, which includes transformations due to both intrinsic and extrinsic camera parameters.

3.2 IMAGE PRE-PROCESSING FOR TARGET RECOGNITION AND LOCATION

Image pre-processing steps are necessary in order to obtain, with subpixel precision, the 2D coordinates of the center of targets, like retro-reflective ones. The tasks to be performed involve target recognition – unambiguous detection of targets within a scene – and location – precise and accurate determination of the target image center within the digital image frame. (SHORTIS, 1994) compares several methods to perform those tasks. According to the general image conditions (noise, lighting, target shape and size, exactness requirements, and others), the most adequate techniques must be chosen.

3.2.1 Grayscaling

If the original image is unnecessarily colored, conversion to grayscale values should be performed in order to reduce image information. A possible method to convert from Red-Green-Blue (RGB) color values to grayscale values consists of calculating the Luminance component (Y) of the Luminance-Inphase-Quadrature (YIQ) color model and considering Y as the desired grayscale value, using the expression

$$GRAY\_val(i) = 0.299 \cdot RED\_val(i) + 0.587 \cdot GREEN\_val(i) + 0.114 \cdot BLUE\_val(i), \quad (20)$$

where $i = 1..width \cdot height$ is the index of every pixel in the image. See (GONZALEZ, 1992) for more details.

3.2.2 Thresholding

Image thresholding filters are commonly used to decrease the complexity of image data so that reduced computational load is attained for successive pre-processing steps. The output of a thresholding filter is a binary image whose one state will indicate the foreground objects, that is, printed text, a legend, a target, defective part of a material, etc., while the complementary state will correspond to the background. The simplest variation of the thresholding technique consists of using a static threshold, whose value is chosen, for example, based on a simple analysis of global histogram, as shown in Figure 36.

A frequently adopted strategy in industrial metrology is the use, only under controlled lighting conditions, of retro-reflective markers together with static threshold methods, which normally leads to good performance. But when lighting intensity varies, automatic methods must be used in order to dynamically adapt image threshold values under different illumination conditions within the scene. For example, the average lighting value of a grayscale image could be calculated for each frame and used as global thresholding, as in

$$GLOBAL\_THRES = \frac{1}{width \cdot height} \cdot \left(\sum_{i=1}^{width \cdot height} GRAY\_val(i) \right) \cdot \quad (21)$$

In addition, even empirically adjusted constants could be used to scale this value in order to obtain best performance. This is a simplified computation and analysis of the actual histogram of pixel intensities within the image window, which is not usually done in order to reduce computational overhead. With this same goal it can be assumed that the target is centered in the image and therefore the edge pixels are representative of the background noise, which are then the only ones considered for (21).

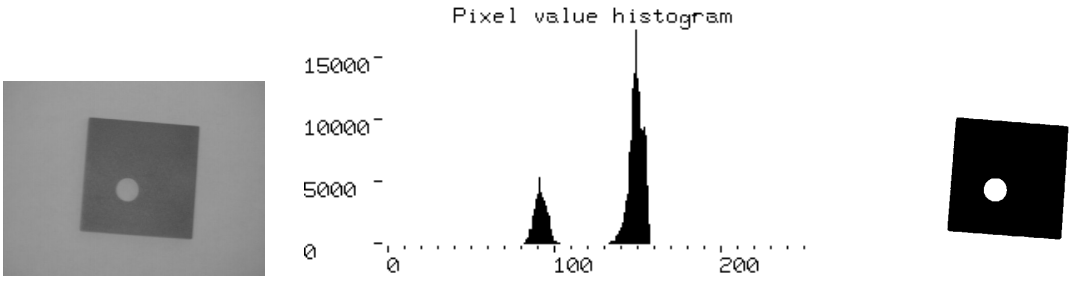

Figure 36 – Thresholding: Original Grayscale Image, Histogram of 8-bit Grayscale Values and Output Binary Image using Static Threshold Value 120

In (SEZGIN, 2004) a very detailed taxonomy study of thresholding methods is presented, where more than 40 techniques are analyzed. Entropy based methods exploit the entropy of the distribution of gray levels in a scene, either interpreting the maximization of the entropy of the thresholded image as indicative of maximum information transfer (PUN, 1980) (YEN, 1995) or trying to minimize the cross-entropy between the input grayscale image and the output binary image (BRINK, 1996) (LI, 1998) in order to avoid information loss. Some methods explore attributes similarities between original and thresholded binary images, e.g. edge matching (HERTZ, 1988) similarity. Old and yet efficient, locally adaptive thresholding techniques (NAKAGAWA, 1979) (SAUVOLA, 2000) consider local statistics of the image, like gray values range or variance, to calculate the threshold value. These methods have main application in thresholding for images with high lighting variation.

3.2.3 Blob Coloring

To achieve target recognition, an automated method must be used to scan the image window and search for potential targets. There are many operators which can be applied to identify and list regions of pixels having intensity values within specific ranges, e.g. above a certain grayscale threshold or even white pixels in the case of a binary image. The process of image segmentation into regions of adjacent pixels is called *Labeling* or *Blob Coloring*.

The *eight-connect operator* or *eight-way connectivity*, so called because it tests and connects all 8 neighboring pixels as just one region, is the most used technique to identify regions. In this method the whole image, from top to bottom on a line basis, must be scanned twice for complete segmentation: one time to enumerate the regions and one time to merge adjacent regions. In fact only 5 adjacent pixels must be tested in order to cover all 8 possible neighbors. Figure 37 shows this operator (a), a sample image (b), over which the operator is run, the result after one pass of the operator (c),

where all possible regions are identified, and the final result after second pass (d), where regions are already merged.

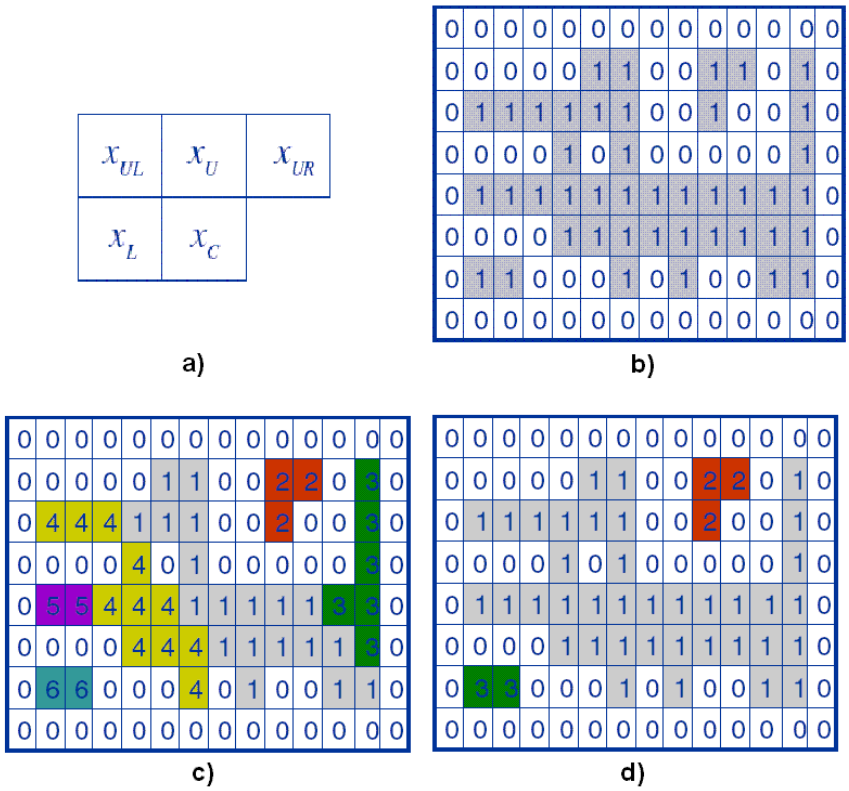

Figure 37 – Eight-connect Operator: a) 5-element Test Operator; b) Original Binary Image; c) Resulting Image after 1st Pass — 6 Regions Identified; d) Resulting Image after 2nd Pass — after Merging, only 3 Regions Identified

Some automated methods (SHORTIS, 1994) (CLARKE, 1993) only consider grayscale values above a certain threshold, keeping 8-bit information and using this to calculate target centers, after identifying them by using a simple eight-way search like the one previously shown. An advantage of this operator is that a list with maximum X and maximum Y dimensions, in addition to perimeter length, is automatically available for each region after the line-scan process is finished.

3.2.4 Edge Detection

Instead of performing threshold algorithms and then looking for specific image regions, edge detection algorithms could be used to directly obtain the regions' contour, which leads next also to a search for their boundaries. Edge points are pixels at or around which the image values undergo a sharp variation. Based on this fact, edge detectors typically try to maximize intensity value gradients. The Canny algorithm (CANNY, 1986) is probably the most used edge detector in the machine vision community. Other methods that are simpler but still very disseminated were proposed by Roberts (ROBERTS, 1965) and Sobel (DUDA, 1973).

3.2.5 Hough Transform

The Hough transform (HOUGH, 1959) was introduced to detect complex patterns of points in binary images and became quickly a very popular algorithm to detect lines and other geometric shapes. The key idea behind the algorithm is to map a complex pattern detection problem in the image space into a simple peak detection problem in the space of line parameters.

The input for the Hough transform is an image with contour points such as the output of an edge detector algorithm. Any line with equation $y = a \cdot x + b$ is identified by a unique parameter pair (a, b). Thus, the line is represented by a point in the parameter space (a plane built by varying parameters *a* and *b*). Correspondingly, any point (x, y) in the image space can be represented as a line $b = x \cdot (-a) + y$ in the parameter space, which, as *a* and *b* vary, represents all possible image lines through the point (x, y). Therefore, a line defined by N collinear image points is identified in parameter space by the intersection of lines associated with each of the image points. This is illustrated in Figure 38 for the case of 2 points, *p* and *q*, where the intersection of their representations in the line parameter space corresponds to the parameters of the line between the points, in the image space.

In order to obtain the parameters for the most likely line to pass through several points, the line intersection problem is transformed in a peak detection problem in the line parameter space. Accumulators are associated to each element of a grid drawn on the line parameter plane, the resolution of which depending on the desired accuracy for determination of parameters. Then, for each line passing through the grid element area, the counter of this element is incremented. In the end, the grid element with the highest count defines the parameters for the line connecting the points.

A variation of the Hough transform, which uses polar representation $p = x \cdot \cos\theta + y \cdot \sin\theta$ for lines, where *p* represents the distance between the image origin and the line, and θ the angle or orientation of the line, is a more complete solution, which can represent lines as x=k, where k is a constant, and has finite maximal and minimal parameter values.

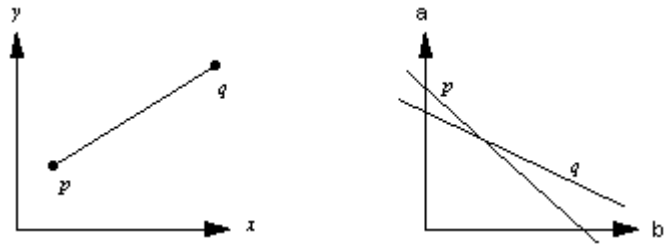

Figure 38 – Hough Transform: Points p and q in Image Space (left) and Representations in Line Parameter Space (right)

3.2.6 Corner detection

For systems where corner features are to be identified, corner detectors are usually more suitable than edge or line detectors followed by additional processing steps. Moravec (MORAVEC, 1979), Harris and Stephens (HARRIS, 1988), and Kitchen and Rosenfeld (KITCHEN, 1982) proposed direct solutions for the corner detection problem. The point feature detector proposed by Tomasi and Kanade (TOMASI, 1992) is widely used at the time of this work.

3.2.7 Filling

After being thresholded and blob colored, identified image regions which can correspond to potential targets may consist mainly of their peripheral pixels, while within the region some pixels remain not filled or marked. Depending on the successive target center location method, these inner gaps have influence on the final calculated center coordinates. In order to avoid this undesired effect, filling methods are used to fill the gaps. Interpolation or constant values are preferred to obtain the intensity values for the filled pixels when not using binary images.

3.2.8 Size and Geometrical Shape Constraints

Subsequently, image regions can be tested for correct shape using a series of straightforward geometry tests, which are based on prior knowledge of the expected size and shape of target images.

The simplest test is a size range criterion, which will reject targets below a minimum size and above a maximum size. Also the ratio of the target boundaries in two perpendicular directions — usually image window width and height directions — can be tested. Previous knowledge of the expected size of actual targets as well as image scaling and expected perspective distortion are needed in order to establish adequate criteria against which to test the size range and extent ratio. These tests generally eliminate spurious targets such as background light sources and unwanted reflections.

The so called *black-white ratio* test can be used to reject targets whose shape is not circular or elliptical. The ratio between the number of non-zero intensity pixels and the area (in pixels) of the smallest possible enclosing rectangular window of the target image is calculated. In both circular and elliptical cases, the expected ratio is one quarter of Pi, as shown in Figure 39 and in

$$\exp ected\_ratio_{BW} = \frac{area_{ellipse}}{area_{rectangle}} = \frac{\frac{\pi \cdot w \cdot h}{4}}{w \cdot h} = \frac{\pi}{4}. \qquad (22)$$

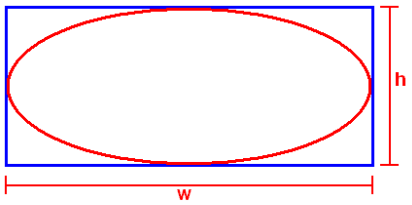

Figure 39 – Black-white Ratio Test: Elliptical Target and Smallest Possible Enclosing Rectangular Window

Similar tests, such as verifying the ratio of the perimeter to the area of the target image blob, can be utilized to ignore targets with incorrect shape.

3.2.9 Marker Center Location

There are several methods which can be used to find the subpixel location of a 2D target image. The simplest method is to consider the center of the smallest rectangle surrounding the target region as the center of the region itself, as in

$$\begin{aligned} x_c &= \frac{w}{2}, \\ y_c &= \frac{h}{2} \end{aligned} \tag{23}$$

where w and h are width and height of the region, respectively, and (x_c, y_c) are the center coordinates.

Averaging the coordinates of the perimeter of the marker, called *Average of Perimeter*, is also a simple method and can be described as

$$\begin{aligned} x_c &= \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \\ y_c &= \frac{1}{n} \cdot \sum_{i=1}^{n} y_i \end{aligned}, \tag{24}$$

where (x_i, y_i) are the coordinates of the i-th pixel belonging to the perimeter of the target region, n is the total number of pixels comprised in the perimeter of the region and (x_c, y_c) are the coordinates of the subpixel location of the marker center.

The *Binary Centroid* method averages the coordinates of every pixel inside the target region whose intensity value is greater than a threshold value. Equations of this technique are given by

$$x_c = \frac{\sum_{j=1}^{w}\sum_{i=1}^{h} I_{i,j} \cdot x_{i,j}}{\sum_{j=1}^{w}\sum_{i=1}^{h} I_{i,j}}, \qquad y_c = \frac{\sum_{j=1}^{w}\sum_{i=1}^{h} I_{i,j} \cdot y_{i,j}}{\sum_{j=1}^{w}\sum_{i=1}^{h} I_{i,j}} \tag{25}$$

where $(x_{i,j}, y_{i,j})$ are coordinates of the i-th pixel in height and of the j-th pixel in width within the target region, w and h are width and height of the region, $I_{i,j}$ is one or zero depending on the threshold and the intensity of the i,j-th pixel and (x_c, y_c) are the coordinates of the calculated center. This technique can be expanded to a *Grayscale Centroid* method if no thresholding is applied to the pixel intensity values, which are then considered within (25).

The *Ellipse Fitting* method tries to find least-squares fit of an arbitrary ellipse to a set of points considered to lie on the perimeter of the target region. Other similar fitting methods, as the *Gaussian Distribution Fitting*, provide more complex ways of finding marker's center coordinates.

(WANG, 2004) and (WEST, 1990) present interesting surveys on subpixel location methods.

3.2.10 Top-hat operator

Morphological operators are an interesting alternative to recognize feature points in labels. Among these techniques is the Top-hat operator (MEYER, 1977), well known for good results when applied to extract very bright dots on a dark background, emphasizing image features with high contrast variations. The Top-hat operator consists of an *opening* operation followed by a subtraction of the resulting image from the original image. The opening operator is composed of an *erosion* operation followed by a *dilation* operation – see (GONZALEZ, 1992) for details about these operators. Figure 40 shows each step of the Top-hat operator applied on a grayscale image with high contrast dots.

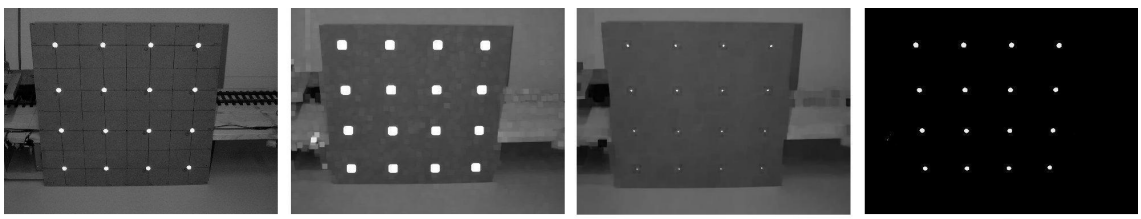

Figure 40 – Top-hat Operator: a) Original Grayscale Image with Small Bright Dots; b) Erosion applied on (a); c) Dilation applied on (b); d) Resulting Image: (c) Subtracted from (a)

3.3 CALIBRATION

Camera calibration is essentially the estimation of intrinsic and extrinsic parameters of the camera, as defined in sections 3.1.3 and 3.1.4, including estimation of lens distortion coefficients. Usually intrinsic parameters must be calibrated, while extrinsic parameters must only be included if there is an external reference (in translation and orientation) for the coordinates system.

The main idea of calibration procedure is to write projection equations which link known coordinates of a set of 3D points in space and their 2D projections, and then solve those for the camera parameters. In order to know the coordinates of some 3D points, a calibration pattern can be built, which has known geometry and allows extraction of some image features so that accurate 3D positions of these features can be located. When using retro-reflective markers, the same pre-processing used for tracking can also be used to extract 2D position of markers, and only the geometric relations between markers of the pattern must be known.

The exactness of the calibration procedure is directly related to the exactness of measurements of the calibration pattern as well as to the precision used to build the pattern, which must be one or two orders of magnitude higher than the precision required from the calibration. It is also possible to do calibration without calibration patterns, extracting 3D information directly from the environment. The use of calibration patterns facilitates however the procedure.

There are many calibration algorithms. Some of them estimate the complete camera projection matrix (see section 3.1.5), such as the technique proposed by Longuet-Higgins (LONGUET-HIGGINS, 1981), called the *eight-point algorithm*, because at least 8 3D points are necessary to estimate the 3D relation between cameras of two different views. Intrinsic and extrinsic parameters can then be extracted from the matrix. Others algorithms estimate directly the parameters. In the subsequent sections some well known algorithms are presented.

3.3.1 Direct Linear Transformation (DLT)

The Direct Linear Transformation (DLT) algorithm was first proposed by Abdel-Aziz and Karara (ABDEL-AZIZ, 1971). It has the advantage of estimating the camera parameters based on the solution of a system of linear equations. In this section this technique is briefly explained.

The DLT method assumes two different coordinate system references: the object-space and the image-plane reference frames, as shown in Figure 41. The optics of the camera maps the point O with coordinates (x, y, z) in the object space to the point I with coordinates (u, v) in the image plane, based on the central projection point N. Points O, I and N are aligned, defining a collinearity condition which is the basis of the DLT method.

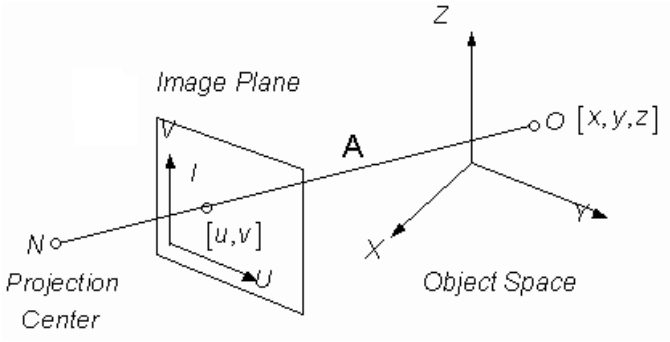

Figure 41 – Object-space and Image-plane Reference Frames used in DLT

Considering the projection center N to be positioned at (x_o, y_o, z_o) in the object-space reference frame, the vector A from N to O can be expressed as being ($x-x_o$, $y-y_o$, $z-z_o$). The addition of a third axis W to the image plane reference frame, which then becomes three-dimensional, changes coordinates of point I to (u, v, 0). The principal point P, referred in section 3.1.3, is shown in Figure 42 with coordinates (u_o, v_o, 0). The distance d between points P and N is called *principal distance*, and is equivalent to the camera's focal length (see section 3.1.2). The point N in the image-plane reference frame has coordinates (u_o, v_o, d) and the vector B drawn from point N to I becomes ($u-u_o$, $v-v_o$, -d).

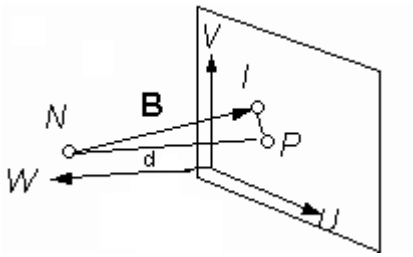

Figure 42 – Vector and Points in Image-plane Reference Frame used in DLT

Taking into account that points O, I and N are collinear, vectors A (Figure 41) and B (Figure 42) form a single line, and this condition can be written as

$$B_{image} = s \cdot A_{object}\,, \tag{26}$$

where s is a scale factor, B_{image} is the B vector in the image-plane reference frame and A_{object} is the A vector originally described in the object-space reference frame. In order to use (26) it is necessary to convert vector A to the image-plane reference frame, introducing the transformation matrix

$$T_{image\_object} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{27}$$

which includes only rotation, since translation is given by the position of point N (x_o, y_o, z_o) , and applying it to A_{object}, obtaining

$$A_{image} = T_{image\_object} \cdot A_{object} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot A_{object}, \tag{28}$$

where A_{image} is the A vector in image-plane coordinates. Combination of (26) and the vectors A and B yields

$$\begin{bmatrix} u-u_o \\ v-v_o \\ -d \end{bmatrix} = s \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} x-x_o \\ y-y_o \\ z-z_o \\ 1 \end{bmatrix} = \begin{bmatrix} s \cdot \left[r_{11} \cdot (x-x_o) + r_{12} \cdot (y-y_o) + r_{13} \cdot (z-z_o) \right] \\ s \cdot \left[r_{21} \cdot (x-x_o) + r_{22} \cdot (y-y_o) + r_{23} \cdot (z-z_o) \right] \\ s \cdot \left[r_{31} \cdot (x-x_o) + r_{32} \cdot (y-y_o) + r_{33} \cdot (z-z_o) \right] \end{bmatrix}. \tag{29}$$

From (29), third row in the matrices, the expression

$$s = \frac{-d}{r_{31} \cdot (x-x_o) + r_{32} \cdot (y-y_o) + r_{33} \cdot (z-z_o)} \tag{30}$$

is obtained. Coordinates u, v, u_o and v_o use physical units (such as mm or μm). The transformation between these and discrete units of acquisition systems, like pixels, is carried out by the conversion factors s_u and s_v, which generate discrete coordinates u_{pix}, v_{pix}, u_{o_pix} and v_{o_pix} (in pixels), expressed as

$$\begin{aligned} u - u_o &= s_u \cdot \left(u_{pix} - u_{o\_pix} \right) \\ v - v_o &= s_v \cdot \left(v_{pix} - v_{o\_pix} \right) \end{aligned}. \tag{31}$$

Applying (30) and (31) to the first two rows of (29) results in

$$\begin{aligned} u_{pix} - u_{o\_pix} &= \frac{-d}{s_u} \cdot \frac{r_{11} \cdot (x-x_o) + r_{12} \cdot (y-y_o) + r_{13} \cdot (z-z_o)}{r_{31} \cdot (x-x_o) + r_{32} \cdot (y-y_o) + r_{33} \cdot (z-z_o)} \\ v_{pix} - v_{o\_pix} &= \frac{-d}{s_v} \cdot \frac{r_{21} \cdot (x-x_o) + r_{22} \cdot (y-y_o) + r_{23} \cdot (z-z_o)}{r_{31} \cdot (x-x_o) + r_{32} \cdot (y-y_o) + r_{33} \cdot (z-z_o)} \end{aligned}. \tag{32}$$

Rearranging (32) for variables x, y and z, and considering the naming conventions defined in section 3.1.3 for the intrinsic parameters (d=f, $s_u=s_x$, $s_v=s_y$, $u_{o\_pix}=o_x$, $v_{o\_pix}=o_y$, $u_{pix}=x_c$, $v_{pix}=y_c$), the expression

$$\begin{aligned} x_c - o_x &= \frac{-f}{s_x} \cdot \frac{r_{11} \cdot (x-x_o) + r_{12} \cdot (y-y_o) + r_{13} \cdot (z-z_o)}{r_{31} \cdot (x-x_o) + r_{32} \cdot (y-y_o) + r_{33} \cdot (z-z_o)} \\ y_c - o_y &= \frac{-f}{s_y} \cdot \frac{r_{21} \cdot (x-x_o) + r_{22} \cdot (y-y_o) + r_{23} \cdot (z-z_o)}{r_{31} \cdot (x-x_o) + r_{32} \cdot (y-y_o) + r_{33} \cdot (z-z_o)} \end{aligned} \tag{33}$$

is obtained. Rewriting (33) in order to emphasize the relationships between camera-plane and object-space reference frames coordinates results in

$$\begin{aligned} x_c &= \frac{L_1 \cdot x + L_2 \cdot y + L_3 \cdot z + L_4}{L_9 \cdot x + L_{10} \cdot y + L_{11} \cdot z + 1} \\ y_c &= \frac{L_5 \cdot x + L_6 \cdot y + L_7 \cdot z + L_8}{L_9 \cdot x + L_{10} \cdot y + L_{11} \cdot z + 1} \end{aligned}, \tag{34}$$

being coefficients L_1 to L_{11} the *DLT parameters*, whose expressions are

$$\begin{aligned} L_1 &= \frac{o_x \cdot r_{31} - f_x \cdot r_{11}}{D} \\ L_2 &= \frac{o_x \cdot r_{32} - f_x \cdot r_{12}}{D} \\ L_3 &= \frac{o_x \cdot r_{33} - f_x \cdot r_{13}}{D} \\ L_4 &= \frac{\left(f_x \cdot r_{11} - o_x \cdot r_{31}\right) \cdot x_o + \left(f_x \cdot r_{12} - o_x \cdot r_{32}\right) \cdot y_o + \left(f_x \cdot r_{13} - o_x \cdot r_{33}\right) \cdot z_o}{D} \\ L_5 &= \frac{o_y \cdot r_{31} - f_y \cdot r_{21}}{D} \\ L_6 &= \frac{o_y \cdot r_{32} - f_y \cdot r_{22}}{D} \\ L_7 &= \frac{o_y \cdot r_{33} - f_y \cdot r_{23}}{D} \\ L_8 &= \frac{\left(f_y \cdot r_{21} - o_y \cdot r_{31}\right) \cdot x_o + \left(f_y \cdot r_{22} - o_y \cdot r_{32}\right) \cdot y_o + \left(f_y \cdot r_{23} - o_y \cdot r_{33}\right) \cdot z_o}{D} \\ L_9 &= \frac{r_{31}}{D} \\ L_{10} &= \frac{r_{32}}{D} \\ L_{11} &= \frac{r_{33}}{D} \end{aligned}, \tag{35}$$

where

$$\begin{aligned} \left(f_x, f_y\right) &= \left(\frac{f}{s_x}, \frac{f}{s_y}\right) \\ D &= -\left(x_o \cdot r_{31} + y_o \cdot r_{32} + z_o \cdot r_{33}\right) \end{aligned}. \tag{36}$$

Expressions (35) include the 5 intrinsic parameters f, s_x, s_y, o_x and o_y, but do not include lens radial distortion coefficients. In standard DLT method, which does not include optical errors, only linear equations need to be solved. If optical distortion modelling is requested, three additional parameters L_{12} to L_{14} can be added, representing 3rd, 5th and 7th order distortion terms of radial distortion modelling coefficients. In this case, a system of non-linear equations must be solved. Details about the different approach methods for calculation of DLT solution can be found in (WONG, 1975).

In (35) complete calculations for camera's extrinsic parameters are included, with rotation (3 angles yaw, pitch and roll, described as a transformation matrix) and translation (3 coordinates x_o, y_o, z_o) between the coordinates system.

Rearranging (34) results in

$$\begin{aligned} \frac{1}{R}\cdot x_c &= \frac{1}{R}\cdot\left(L_1\cdot x + L_2\cdot y + L_3\cdot z + L_4 - L_9\cdot x_c\cdot x - L_{10}\cdot x_c\cdot y - L_{11}\cdot x_c\cdot z\right) \\ \frac{1}{R}\cdot y_c &= \frac{1}{R}\cdot\left(L_5\cdot x + L_6\cdot y + L_7\cdot z + L_8 - L_9\cdot y_c\cdot x - L_{10}\cdot y_c\cdot y - L_{11}\cdot y_c\cdot z\right) \end{aligned}, \tag{37}$$

where

$$R = L_9\cdot x + L_{10}\cdot y + L_{11}\cdot z + 1. \tag{38}$$

Equation (37) is equivalent to

$$\begin{bmatrix} \frac{x}{R} & \frac{y}{R} & \frac{z}{R} & \frac{1}{R} & 0 & 0 & 0 & 0 & \frac{-x_c\cdot x}{R} & \frac{-x_c\cdot y}{R} & \frac{-x_c\cdot z}{R} \\ 0 & 0 & 0 & 0 & \frac{x}{R} & \frac{y}{R} & \frac{z}{R} & \frac{1}{R} & \frac{-y_c\cdot x}{R} & \frac{-y_c\cdot y}{R} & \frac{-y_c\cdot z}{R} \end{bmatrix}\cdot\begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_{11} \end{bmatrix} = \begin{bmatrix} \frac{x_c}{R} \\ \frac{y_c}{R} \end{bmatrix}, \tag{39}$$

which is the expression relating image-plane and object-space coordinates system, for only one *control point*, which is usually fixed to a calibration frame or pattern and belongs to a group of control points. Those must not be coplanar, i.e., they must form a control volume.

Each control point, with known object-space and image-plane coordinates, provides two equations. Considering the usual number of 11 parameters, at least 6 control points are needed to solve for the 11 unknowns. For the calibration procedure, even though the minimum number of control points has already been reached, additional points are always recommended in order to obtain more precise calculations.

Equation (39), after expanded for more control points, can be written as

$$X = L\cdot Y\,. \tag{40}$$

Solving for L in (40) by use, for instance, of a least squares method yields

$$\begin{aligned} &\left(X^t\cdot X\right)\cdot L = X^t\cdot Y \\ &\left(X^t\cdot X\right)^{-1}\cdot\left(X^t\cdot X\right)\cdot L = \left(X^t\cdot X\right)^{-1}\cdot\left(X^t\cdot Y\right)\cdot \\ &L = \left(X^t\cdot X\right)^{-1}\cdot\left(X^t\cdot Y\right) \end{aligned} \tag{41}$$

Once the parameters L_1-L_{11} are known, (35) can be rearranged to calculate the intrinsic and extrinsic camera parameters.

3.3.2 Tsai's Method

The calibration technique described by Tsai (TSAI, 1986) (TSAI, 1987) is probably the most popular calibration procedure in computer vision systems nowadays, basically due to simplicity and separated calibration of intrinsic and extrinsic parameters. It is a two-stage technique, which computes first extrinsic parameters (translation and rotation) and then intrinsic ones. The method can deal with both coplanar and non-coplanar control points.

Tsai's technique is based on the *radial alignment constraint* (RAC), which is only a function of the relative translation and rotation – except for the z component – between the camera and the calibration or control points. Other calibration parameters are computed with normal projective equations. As a restriction, when single-plane control points are used, the plane must not be exactly parallel to the image plane.

The camera model used, depicted in Figure 43, is similar to the DLT camera model, described in Figure 41, with some minor differences. (x_w, y_w, z_w) are the 3D coordinates of the object point P in the 3D world coordinates system. (x, y, z) are the 3D coordinates of the same object point P in the 3D camera coordinates system, which is centered at point O, the optical center, with the z axis the same as the optical axis. (X_i, Y_i) is the image coordinates system centered at O_i and parallel to x and y axis. The focal length, f, is the distance between front image plane and the optical center. (X_u, Y_u) are the image coordinates of (x, y, z) if a perfect pinhole camera model is used. (X_d, Y_d) are the actual image coordinates which differ from (X_u, Y_u) due to optics distortion. (X_f, Y_f) are the coordinates used in the computer, given in number of pixels for the discrete image in the frame memory.

The calibration technique subdivides in 4 steps the transformation from 3D world coordinates to computer image coordinates. Initially, the rigid body transformation from the object world coordinates system (x_w, y_w, z_w) to the camera 3D coordinates system (x, y, z) is accomplished through rotation followed by translation, as opposed to the sequence used in the DLT method. This inverted order is a key feature for the development and success of the technique. The transformation is represented by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \cdot \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \,, \tag{42}$$

where

$$R = \begin{bmatrix} r_1 & r_2 & r3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \text{ and } T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{43}$$

are the rotation matrix and the translation vector, respectively. R and T are extrinsic parameters to be calibrated.

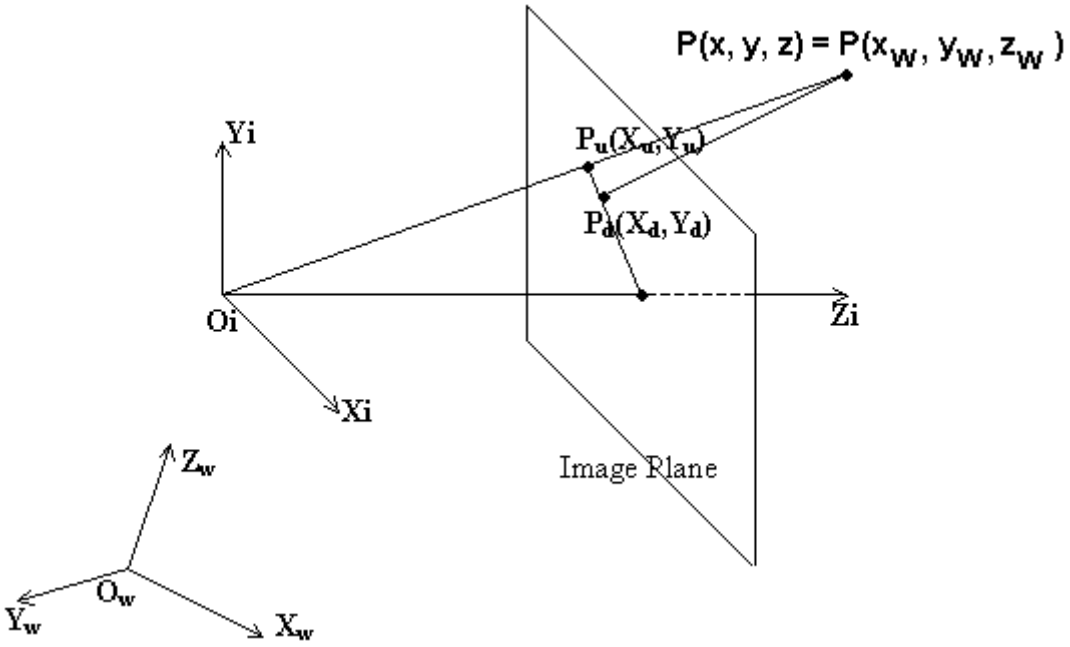

Figure 43 – Camera Model for Tsai's Calibration Technique

The second step is the transformation from 3D camera coordinates (x, y, z) to ideal, undistorted, image coordinates (X_u, Y_u) using perspective projection considering the pinhole camera geometry as in (6), yielding

$$\begin{aligned} X_u &= f\frac{x}{z} \\ Y_u &= f\frac{y}{z} \end{aligned}. \tag{44}$$

The parameter to be calibrated is the effective focal length f.

The third step is the correction of lens distortion, represented by

$$\begin{aligned} X_d + D_x &= X_u \\ Y_d + D_y &= Y_u \end{aligned}, \tag{45}$$

where (X_d, Y_d) are the distorted or real image coordinates on the image plane, and

$$\begin{aligned} D_x &= X_d \cdot \left(k_1 \cdot r^2 + k_2 \cdot r^4 + \ldots\right) \\ D_y &= Y_d \cdot \left(k_1 \cdot r^2 + k_2 \cdot r^4 + \ldots\right) \\ r &= \sqrt{X_d{}^2 + Y_d{}^2} \end{aligned}. \tag{46}$$

Here only radial distortion is taken into account. Practical experiments show that only the first term needs to be considered, thus the distortion coefficient k_1 is the only parameter to be calibrated.

The last step is the transformation from real image coordinates (X_d, Y_d) to computer image coordinates (X_f, Y_f), represented by

$$\begin{aligned} X_f &= s_x \cdot d_x^{'-1} \cdot X_d + C_x \\ Y_f &= d_y^{-1} \cdot Y_d + C_y \end{aligned}, \tag{47}$$

where (X_f, Y_f) are the row and column numbers of image pixel in computer frame memory, (C_x, C_y) are the row and column numbers of the center of computer frame memory,

$$d_x^{'} = d_x \cdot \frac{N_{cx}}{N_{fx}}, \tag{48}$$

(d_x, d_y) are the center to center distances between adjacent sensor elements in X and Y direction, respectively, N_{cx} is the number of sensor elements in the X direction, N_{fx} is the number of pixels in a line as sampled by the computer and s_x is the uncertainty image scale factor, the only parameter to be calibrated.

The combination of the last 3 steps leads to the expressions

$$\begin{aligned} s_x^{-1} d_x^{'} X + s_x^{-1} d_x^{'} X k_1 r^2 &= f \frac{x}{z} \\ d_y^{'} Y + d_y Y k_1 r^2 &= f \frac{y}{z} \end{aligned}, \tag{49}$$

where

$$r = \sqrt{\left(s_x^{-1} d_x^{'} X\right)^2 + \left(d_y Y\right)^2}, \tag{50}$$

as the relations between the computer coordinates (X_f, Y_f) and the 3D coordinates of the object points in camera coordinates system (x, y, z). Substituting (42) into (49) gives

$$\begin{aligned} s_x^{-1} d_x^{'} X + s_x^{-1} d_x^{'} X k_1 r^2 &= f \frac{r_1 x_w + r_2 y_w + r_3 z_w + T_x}{r_7 x_w + r_8 y_w + r_9 z_w + T_z} \\ d_y^{'} Y + d_y Y k_1 r^2 &= f \frac{r_4 x_w + r_5 y_w + r_6 z_w + T_y}{r_7 x_w + r_8 y_w + r_9 z_w + T_z} \end{aligned}, \tag{51}$$

where r is given by (50). Equations (50) and (51) are used in both steps of parameters' determination.

In summary, the intrinsic parameters to be calibrated are the effective focal length f, the sole lens distortion coefficient k_1, the uncertainty scale factor s_x and the origin computer image coordinates (C_x, C_y). The extrinsic parameters to be calibrated are the three components for the translation, vector T, and the three Euler angles yaw (θ), pitch (ϕ) and roll (ψ). The rotation matrix R can be expressed as function of these angles as

$$R = \begin{bmatrix} \cos\psi \cdot \cos\theta & \sin\psi \cdot \cos\theta & -\sin\theta \\ -\sin\psi \cdot \cos\phi + \cos\psi \cdot \sin\theta \cdot \cos\phi & \cos\psi \cdot \cos\phi + \sin\psi \cdot \sin\theta \cdot \sin\phi & \cos\theta \cdot \sin\phi \\ \sin\psi \cdot \sin\phi + \cos\psi \cdot \sin\theta \cdot \cos\phi & -\cos\psi \cdot \sin\phi + \sin\psi \cdot \sin\theta \cdot \cos\phi & \cos\theta \cdot \cos\phi \end{bmatrix} \cdot \quad (52)$$

The base of this calibration technique is the radial alignment constraint, illustrated in Figure 44, which asserts that the radial distortion does not alter direction of vectors from origin to image point, which leads to the conclusion that vectors $\overline{O_i P_d}$ and $\overline{O_i P_u}$, from the origin O_i of the image plane to the real image point (X_d, Y_d) and to the undistorted image point (X_u, Y_u), respectively, as well as the vector $\overline{P_{oz} P}$, extending from the optical axis to the object point (x, y, z), are parallel.

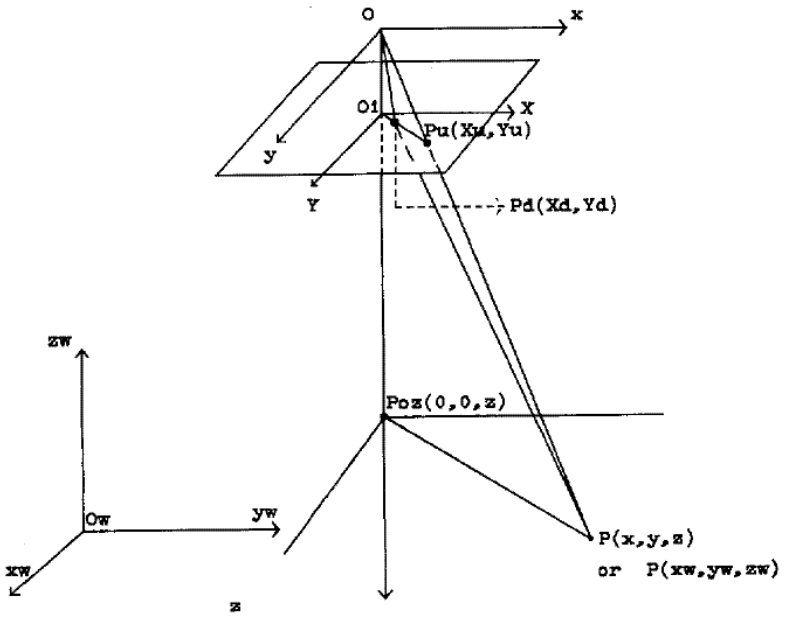

Figure 44 – Illustration of the Radial Alignment Constraint (RAC) (TSAI, 1987)

In order to use the technique to calibrate parameters based on a monoview coplanar set of control points, first the 3D complete orientation as well as the x and y translation information are computed. Then the effective focal length, distortion coefficients and the z information for translation are computed. In the case of coplanar points, the s_x factor is not calibrated, thus it must be previously known. If the s_x scale factor is not known, a non-coplanar set of control points must be used, and the steps are the same as for coplanar points. Details of the mathematical calculations used in Tsai's technique, as well as accuracy and precision test results, are available in (TSAI, 1986) and (TSAI, 1987).

3.3.3 Zhang's Method

Zhang implemented a flexible camera calibration technique (ZHANG, 2000) (ZHANG, 1999) that requires multiple views of a planar pattern shown at a few

different orientations. Either the camera or the pattern can be freely moved, without previous knowledge of this movement. The method is based on a closed-form solution followed by a non-linear optimization based on the maximum likelihood criterion. The advantage of this technique is the simplicity, since an ordinary planar pattern (a grid or chessboard-like pattern) can be printed out with high accuracy and attached to a rigid body, thus not demanding expensive calibration frames. Zhang's technique is implemented in the camera calibration routines included in OpenCV[1], an open-source library for computer vision algorithms, provided by Intel.

The technique explores the constraints on the camera's intrinsic parameters provided by observing a single plane. A 2D point is defined by $m = [u, v]^T$ and a 3D point is defined by $M = [X, Y, Z]^T$. In preparation for use of homogeneous coordinates, both points are rewritten as $\tilde{m} = [u, v, 1]^T$ and $\tilde{M} = [X, Y, Z, 1]^T$. Considering the pinhole camera model as in Figure 33, the relationship between a 3D point $\tilde{M}$ and its 2D projection on image plane $\tilde{m}$ is given by

$$s \cdot \tilde{m} = A \cdot [R \quad t] \cdot \tilde{M}, \tag{53}$$

where

$$A = \begin{bmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}; \tag{54}$$

s is an arbitrary scale factor; R and t are the extrinsic parameters of the camera, which consist of a rotation followed by a translation, as in Tsai's method; A is the matrix containing the intrinsic parameters, where (u_0, v_0) are the coordinates of the principal point, α and β are scale factors in image x and y axes, and c describes the skewness of the two image axes, that is, a measure of the angle alignment between actual and theoretical z axes in camera coordinates system.

Considering points on the pattern plane as having the Z coordinate equal to zero, and denoting the i-th column of the rotation matrix R by r_i, results in

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \cdot [r_1 \quad r_2 \quad r_3 \quad t] \cdot \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A \cdot [r_1 \quad r_2 \quad t] \cdot \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \tag{55}$$

which can be rewritten as

[1] OpenCV from Intel Corporation. Open-source Computer Vision Library. Available at <http://www.intel.com/ technology/computing/opencv>. Last accessed on December 16th, 2005.

$$s \cdot \tilde{m} = H \cdot \tilde{M} \text{ , with } H = A \cdot \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}, \tag{56}$$

where H is defined by Zhang as the homography relating a 3D point $\tilde{M}$ on the pattern and its image point $\tilde{m}$. H is defined only up to a scale factor (*s*). Given a view of the pattern's plane, its homography can be estimated. For details see (ZHANG, 1999). Rewriting H using column vector h_i yields

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} H = k \cdot A \cdot \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}, \tag{57}$$

where k is an arbitrary scale factor. Being r_1 and r_2 orthonormal, the scalar product between them is zero and their length equals the unity, resulting in

$$\begin{aligned} & h_1^T \cdot \left(A^T\right)^{-1} \cdot A^{-1} \cdot h_2 = 0 \\ & h_1^T \cdot \left(A^T\right)^{-1} \cdot A^{-1} \cdot h_1 = h_2^T \cdot \left(A^T\right)^{-1} \cdot A^{-1} \cdot h_2 \end{aligned}, \tag{58}$$

which are the 2 constraints of the intrinsic parameters, given one homography. Since a homography has 8 degrees of freedom and there are 6 extrinsic parameters (3 rotation angles and 3 translation coordinates), only these 2 constraints relating intrinsic parameters can be obtained.

The solving procedure using Zhang's technique starts with a closed-form solution for the intrinsic parameters. Consider $B = \left(A^T\right)^{-1} \cdot A^{-1}$, where

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{21} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{c}{\alpha^2 \cdot \beta} & \frac{c \cdot v_0 - u_0 \cdot \beta}{\alpha^2 \cdot \beta} \\ -\frac{c}{\alpha^2 \cdot \beta} & \frac{c^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{c \cdot (c \cdot v_0 - u_0 \cdot \beta)}{\alpha^2 \cdot \beta^2} - \frac{v_0}{\beta^2} \\ \frac{c \cdot v_0 - u_0 \cdot \beta}{\alpha^2 \cdot \beta} & -\frac{c \cdot (c \cdot v_0 - u_0 \cdot \beta)}{\alpha^2 \cdot \beta^2} - \frac{v_0}{\beta^2} & \frac{(c \cdot v_0 - u_0 \cdot \beta)^2}{\alpha^2 \cdot \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \tag{59}$$

is a symmetric matrix, defined by a vector with 6 elements

$$b = \begin{bmatrix} B_{11}, & B_{12}, & B_{22}, & B_{13}, & B_{23}, & B_{33} \end{bmatrix}^T. \tag{60}$$

Considering the i-th column vector of H as $h_i = \begin{bmatrix} h_{i1} & h_{i2} & hi_3 \end{bmatrix}^T$, the expression

$$h_i^T \cdot B \cdot h_j = v_{ij}^T \cdot b \tag{61}$$

is obtained, where

$$v_{ij} = \left[h_{i1} \cdot h_{j1},\ \ h_{i1} \cdot h_{j2} + h_{i2} \cdot h_{j1},\ \ h_{i2} \cdot h_{j2},\ \ h_{i3} \cdot h_{j1} + h_{i1} \cdot h_{j3},\ \ h_{i3} \cdot h_{j2} + h_{i2} \cdot h_{j3},\ \ h_{i3} \cdot h_{j3} \right]^T . \quad (62)$$

The two constraints defined in (58), for a given homography, can be rewritten as 2 homogeneous equations that have b as vector of unknowns

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} \cdot b = 0 \cdot \quad (63)$$

When n views of the same pattern plane are observed, n equations as (63) can be written, yielding

$$V \cdot b = 0 , \quad (64)$$

where V is a $(2 \cdot n) \cdot 6$ matrix. With $n \geq 3$ there is a unique solution b defined up to a scale factor. If $n = 2$ one can impose that the skewness constraint equals zero and then solve for the remaining intrinsic parameters. The solution of (64) is the eigenvector of $V^T \cdot V$ associated with the smallest eigenvalue, which can be obtained by using the well-known technique called Singular Value Decomposition (SVD). For details on this method see (PRESS, 1992).

Once b is estimated, the camera intrinsic parameters in matrix A can be computer. Once A is known, the computation of the camera extrinsic parameters for each view is achieved by

$$\left[r_1,\ \ r_2,\ \ r_3,\ \ t \right] = \left[k \cdot A^{-1} \cdot h_1,\ \ k \cdot A^{-1} \cdot h_2,\ \ r_1 \times r_2,\ \ k \cdot A^{-1} \cdot h_3 \right], \quad (65)$$

with

$$k = \frac{1}{\left\| A^{-1} \cdot h_1 \right\|} = \frac{1}{\left\| A^{-1} \cdot h_2 \right\|} \cdot \quad (66)$$

Next step is the calculation of radial distortion coefficients, achieved by solving a system of linear equations using least-squares methods, and afterwards the whole solution can be refined by maximum likelihood calculations. Details about these procedures as well as practical results are described in (ZHANG, 1999).

Due to the simplicity and exactness, Zhang's technique is widely used in computer vision desktop applications or wherever affordable solutions are sought.

3.4 3D POSE ESTIMATION FROM ONE VIEW

Estimation of 3D pose based on a single view or single camera is the ability of extracting depth information from a single image of a projected 3D target, in order to obtain its location and orientation in space. Basically the existing methods can be divided in marker-based and marker-less techniques.

3.4.1 Marker-based Pose Estimation

Marker-based methods are based on the extraction and processing of well-defined image features – markers – to obtain 3D pose estimation from a single view. The techniques can be either analytical or numerical.

Analytical methods calculate the camera pose from a set of usually non-linear equations making use of known model features extracted from the image, such as angles between lines or edge lengths. One example of analytical method is the one presented by Fischler and Bolles (FISCHLER, 1981), also called Random Sample Consensus (RANSAC), which uses point correspondences between object and image spaces. The algorithm created by Dhome *et al.* (DHOME, 1989) uses line correspondences between object and image spaces. Abidi and Chandra (ABIDI, 1995) proposed a pose estimation algorithm based on the volume measurement of tetrahedra composed of feature-point triplets extracted from an arbitrary quadrangular target and the lens center of the vision system. Hung, Yeh and Harwood (HUNG, 1985) presented an algorithm capable of computing 3D coordinates of the vertices of a quadrangle relative to the camera, given the image of the quadrangle. Horaud *et al.* (HORAUD, 1989) developed an analytical solution for the general perspective 4-point problem, formulated as a biquadratic polynomial equation with one unknown, where further constraints are applied so that the solution does not result in impossible geometric configurations. Yuan (YUAN, 1989) presents a method for computation of 3D pose estimation based on a pure algebraic solution.

Numerical methods try iteratively to minimize an error function which usually relates image features of objects with their projections on the camera's plane, based on an initial estimation. A classic approach by Lowe (LOWE, 1987) implements a least squares minimization of an error function, used iteratively to optimize the projection transformation from object coordinates system to image coordinates system, and compares the results to measured image space coordinates. Appel and Navab (APPEL, 2000) present a numerical algorithm for one-camera tracking system to be used in industrial applications. Cybercodes (REKIMOTO, 1999) implements a tracking algorithm integrated into a seamless hybrid workspace based on augmented reality. ARToolKit (KATO, 1999) (KATO, 2000) and PTrack (SANTOS, 2005), the algorithms used and compared in this work, are both numerical methods, which are described in detail in the following sections.

Zhang, Fronz and Navab (ZHANG, 2002) presented an interesting comparative study of one-camera tracking systems which use planar square coded visual markers, including ARToolKit.

3.4.1.1 ARToolKit Algorithm

The ARToolKit tracking algorithm has been introduced in section 2.6.1 and will be here explained in more detail. Although the algorithm uses an analytical calculation as part of the solution procedure, a refinement of the coefficients values is achieved by an iterative process, thereby the method is considered numerical.

The problem of pose estimation based on a single view is solved in ARToolKit basically exploring the properties of size-known square markers. The

markers are used as primary reference frame. The transformation matrix from marker coordinates to camera coordinates is called T_{cm}, and the operation is represented by

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & W_x \\ V_{21} & V_{22} & V_{23} & W_y \\ V_{31} & V_{32} & V_{33} & W_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \begin{bmatrix} & V_{3x3} & & W_{3x1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = T_{cm} \cdot \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}, \quad (67)$$

where V_{3x3} is the rotation component and W_{3x1} is the translation component of the transformation matrix, (X_c, Y_c, Z_c) are the coordinates in the camera reference frame and (X_m, Y_m, Z_m) are the coordinates in the marker reference frame. Figure 45 shows the coordinates systems involved in these operations.

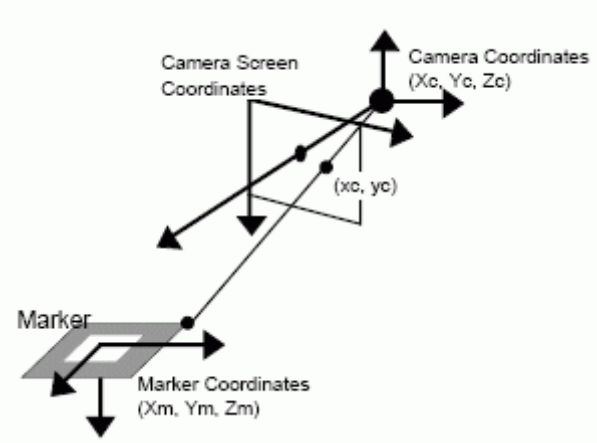

Figure 45 – Coordinates Systems used by ARToolKit (KATO, 1999)

Matrix T_{cm} is estimated by image analysis. Initially a thresholding filter is applied to the input image, then a line-fitting technique, e.g. the Hough Transform (described in section 3.2.5), is used in order to identify regions whose outline contour can be fitted by four line segments. The regions are normalized and the sub-image within the region is compared by template matching with patterns that were previously registered in the system.

The region normalization process, which includes pose estimation and 3D reconstruction, is the key feature of the tracking algorithm. Considering the perspective projection matrix P, obtained by camera calibration and including the camera intrinsic parameters, the perspective transformation can be represented by

$$\begin{bmatrix} h \cdot x_c \\ h \cdot y_c \\ h \\ 1 \end{bmatrix} = P \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}, \text{ where } P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & 0 \\ 0 & P_{22} & P_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (68)$$

(X_c, Y_c, Z_c) are coordinates in the camera reference system and (x_c, y_c) are screen coordinates.

When two parallel sides of a square marker are projected on the camera's screen, the equations of those line segments in camera screen coordinates are given by

$$\begin{aligned} a_1 \cdot x_c + b_1 \cdot y_c + c_1 = 0 \\ a_2 \cdot x_c + b_2 \cdot y_c + c_2 = 0 \end{aligned} . \tag{69}$$

The values of parameters a_1, b_1, a_2 and b_2 have already been obtained in the line-fitting process. Using (68) and (69) the equations of the planes in camera coordinates can be written as

$$\begin{aligned} a_1 \cdot P_{11} \cdot X_c + (a_1 \cdot P_{12} + b_1 \cdot P_{22}) \cdot Y_c + (a_1 \cdot P_{13} + b_1 \cdot P_{23} + c_1) \cdot Z_c = 0 \\ a_2 \cdot P_{11} \cdot X_c + (a_2 \cdot P_{12} + b_2 \cdot P_{22}) \cdot Y_c + (a_2 \cdot P_{13} + b_2 \cdot P_{23} + c_2) \cdot Z_c = 0 \end{aligned} , \tag{70}$$

where all parameters have known values.

Considering the normal vectors of these planes as n_1 and n_2, respectively, it is assumed that the direction vector of two parallel sides of the square is given by the cross product $n_1 \times n_2$. The two unit direction vectors u_1 and u_2, obtained from two different sets of two parallel sides of the square, should be perpendicular. However, due to image processing errors the orthogonality is not guaranteed. In order to correct this problem, two additional orthogonal unit direction vectors v_1 and v_2 are defined on the same plane that includes u_1 and u_2. Given the unit direction vector v_3 perpendicular to both v_1 and v_2, the rotation component V_{3x3} in the transformation matrix T_{cm} is defined as $\begin{bmatrix} V_1^T & V_2^T & V_3^T \end{bmatrix}$.

Since V_{3x3} is given, the remaining unknown translation component W_{3x1} in T_{cm} can be calculated. Using (67) and (68), the following expression can be written,

$$P^{-1} \cdot \begin{bmatrix} h \cdot x_c \\ h \cdot y_c \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} & V_{3x3} & & W_{3x1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} , \tag{71}$$

which can be evaluated for each marker corner, totalizing 8 equations containing W_{3x1} and the corner coordinates in the marker reference frame as unknowns. Given that square marker's dimensions as well as the position of its corners are known, the corner coordinates can be written as function of an origin point on the marker, reducing the unknowns to $\begin{bmatrix} W_x & W_y & W_z \end{bmatrix}^T$ - the translation component of T_{cm}-, which can be evaluated, since there are 3 unknowns to 8 linear dependent (or simultaneous) equations.

The transformation matrix T_{cm} obtained from the method described above may contain errors due to measurement issues, and coefficients can be refined through the following process. Markers' vertex coordinates in the marker coordinates system can be transformed to screen coordinates applying T_{cm}. By iteration, minimization of the sum of differences between these transformed coordinates and the coordinates measured

from the image leads to optimized coefficients in the transformation matrix T_{cm}, which contains the complete pose estimation – rotation and translation - of the square marker.

3.4.1.2 PTrack Algorithm

The PTrack algorithm (SANTOS, 2005) is superficially described in section 2.6.2. A detailed description is provided here, following the sequence shown in Figure 25.

The 2D processing algorithm has as input a set of 2D feature points which represent the projected markers' 2D coordinates for each label in image space.

The identification of possible labels is done by building a quad-tree which segments the image space. The root node of the quad-tree represents the top most quad-tree layer. Each quad-tree node has up to four siblings. Image space points are projections of camera space markers belonging to labels. Initially the image space corresponding to a video frame is sub-divided in four regions and each region is again sub-divided in another four regions and so on. The main objective of using a quad-tree structure is to benefit from the fact that all markers belonging to a label are always nearby. If two labels are visible in image space, it is very likely that they will lie in different quad-tree segments. Thus, for recognition of a label in one segment no other markers belonging to another segment need to be considered, increasing processing speed.

After segmenting the image space (Figure 46) the quad-tree is scanned for possible projections of labels. Once the quad-tree is initialized, a depth-first-search (DFS) is performed on the quad-tree:

- For each node the algorithm checks if more than 5 unrecognized points are in the unrecognized points list of this node. If this is the case, it means that the region could host a potential projection of a label.

- If this is not the case, then all points in this node are transferred one layer up to the father node and added to the list of unrecognized points of the father-node, which covers a larger region.

- If more than 5 unrecognized points are found in the node, then a "radar sweep" algorithm is executed. If a label is then recognized, it is added to the list of recognized labels of the father node.

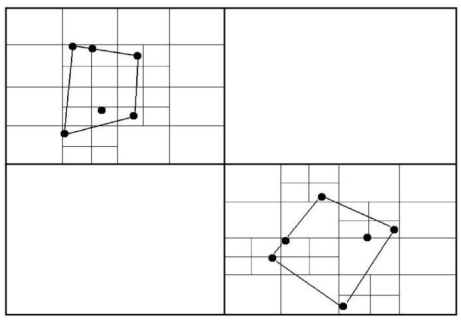

Figure 46 – PTrack: Quad-tree Segmentation of the Image Plane (SANTOS, 2005)

At the end of the scanning process, the root node contains a list of recognized labels and a list of still unrecognized points, which were transferred to the root node from the lowest layers because they were never associated to a label.

When more than 5 unrecognized points are found in a node, then it is possible that the region hosts the projection of a label. A radar sweep (Figure 47) of the region around the center of gravity of all unrecognized points in the region identifies possible top-edges of potential labels containing three points on a line including the top marker. The main idea is to calculate the center of gravity of all unrecognized points in the region defined by the node. Then an imaginary line passing through the center of gravity is rotated up to 180 deg around the center of gravity. For each step of 1 degree, the distances of all points to the line are calculated. All points with the same distance and prefix sign must lie on a line parallel to the radar sweep line (due to measurement errors a certain tolerance difference is applied to define when distances are interpreted as being equal).

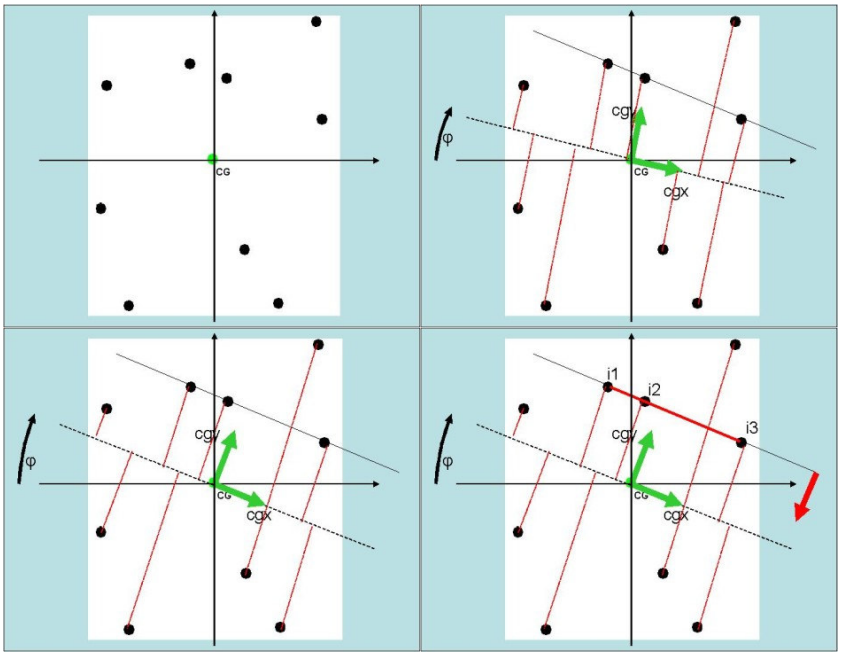

Figure 47 – PTrack's Radar Sweep Algorithm (SANTOS, 2005)

If the algorithm is successful and finds more than two points on a line, then those points are sorted according to their distance from the center of gravity. After sorting, each set of three points is then analyzed to see whether it yields geometric conditions for being a top-edge of a possible label and whether the remaining points of this label are to be found above or below the possible top edge, depending on the location of the top-marker.

Once a possible set of three points has been found, label detection begins. The zero marker of a label is defined as the corner marker closest to the top edge marker. Therefore, by comparing the distances between the second element and the first and last elements of the set, an indication can be obtained, related to which side of the possible top edge the rest of the label has to be searched for.

As a next step, the algorithm tries to find additional corners of the potential label, what is shown in Figure 48. This is done by rotating a copy of the top edge around the first and the second corner which it connects, so both copies are perpendicular to the top-edge, connected to the respective corners of the top-edge. Then the projection of additional corners of a potential label must be close to the ends of both copies and can be found by calculating the distances of all points to be considered from those ends. Once the nearest points are found and interpreted as additional corners, the algorithm attempts to find a coding marker, searching within the boundaries of the four corner markers.

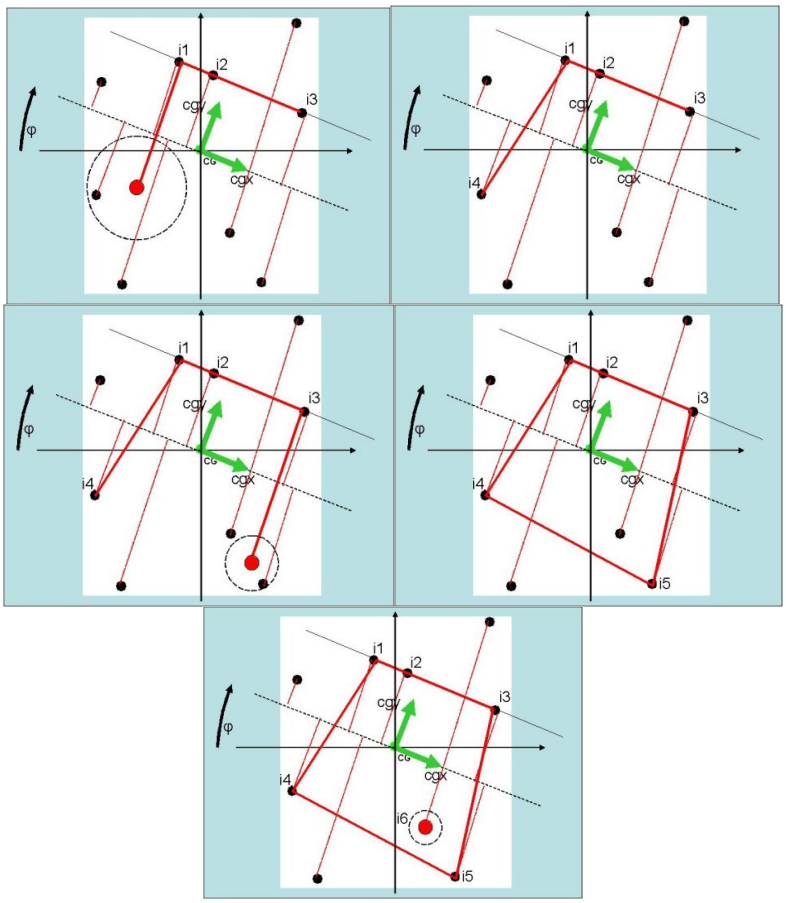

Figure 48 – 2D Label Detection in PTrack Algorithm (SANTOS, 2005)

If a possible projection of a label is found, the label is handed over to the 3D pose estimation algorithm which will be detailed in the next section. The 2D to 3D geometric interpolation algorithm first accurately iterates the rotation of the label in camera space and then scales the label to find the exact translation matching the pre-defined edge lengths. If a fail occurs during reconstruction attempt, the 2D image space coordinates representing a potential label are considered not valid and thus cannot represent a valid label. A reconstructed potential label is also not a valid label if after reconstruction no correspondence is found with a label registered in database.

The translation of 2D image space to 3D camera space coordinates of the potential label, on the image space plane, is possible due to knowledge of where the focal point (origin of the camera space) is in relation to the image plane in 3D.

Projection lines are lines which start in the camera's focal point, go through each of the corners of the projected potential label in the image space plane and extend towards infinite.

If reconstruction is possible, then the resulting corners of the reconstructed label in 3D are on the same projection lines that cross the projection of those corners in the image space plane. Therefore a mapping (Figure 49) must exist which rotates and translates the projection from the image space plane to the original orientation and position in 3D.

This mapping is done by a fast geometric iteration. The basic principle of the algorithm is a round robin scheme largely applied in multitasking operating systems. The particularity in this case is the iterative analysis, in clockwise direction, of the corners of the potential projected label and their associated edges.

Let each corner be clockwise associated to a departing edge of the label. Let a corner be tested at a given moment. If the projection of the edge parallel to its associated edge appears larger than the projection of the associated edge, then the label must be rotated counter-clockwise around the associated edge. If it appears smaller, then the label must be rotated clockwise around the associated edge.

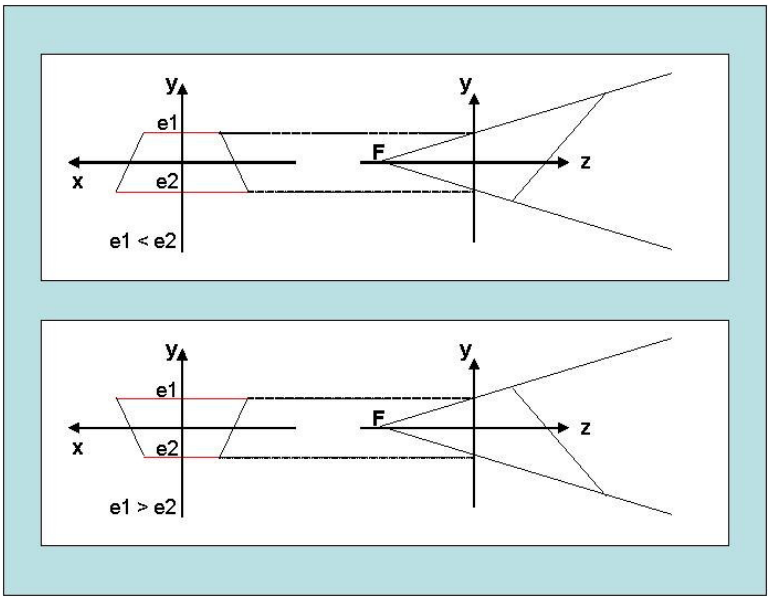

Figure 49 – PTrack: 2D Edge Relations and Correspondence in 3D (SANTOS, 2005)

Figure 50 shows the iteration process. At the beginning of the iteration an estimation of the angle around which to rotate must be defined. Using a Newton's iteration approach, this angle is cut by half after this procedure has been applied clockwise to four corners in a row. The initial guess may be based on the knowledge that in practice retro-reflective markers used do not reflect light at an angle greater than 60 deg. The difference of lengths between both parallel edges is compared before and after applying a rotation. In case the rotation causes the difference of lengths to increase, it is rolled back, and the algorithm proceeds to the next corner. The algorithm stops if the lengths of all edges are identical for a certain accuracy which can be defined at initialization.

Once the rotation reconstruction has been finished, the translation is estimated. The translation in camera space is calculated by scaling the intermediary label along the projection lines until its edge length matches the standard edge length of all registered labels.

After estimating the orientation and location of the projected potential label in camera space coordinates, a coordinate system transformation from camera space coordinates to object space coordinates of the label is applied and the label is then compared to the registered labels in the database. If a match is found, the algorithm has found a label and is finished. At this point, complete pose estimation – rotation and translation – of the label has been estimated by the algorithm.

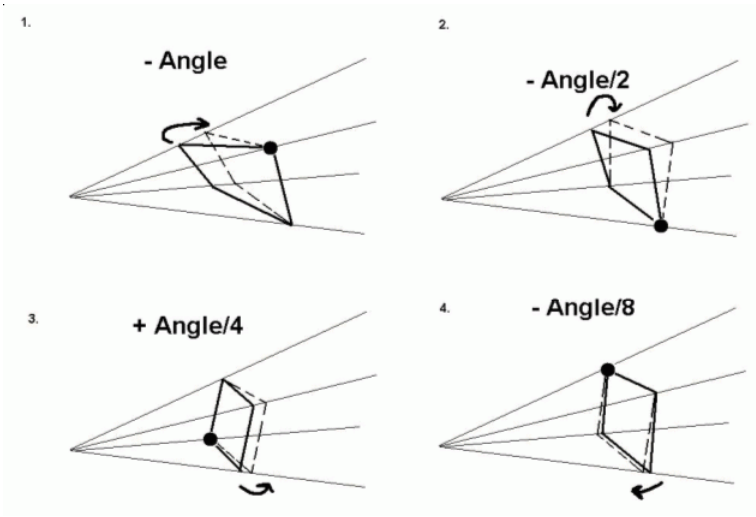

Figure 50 – PTrack: Round Robin Scheme for Orientation Reconstruction (SANTOS, 2005)

3.4.2 Marker-less Pose Estimation

Some techniques for 3D pose estimation do not use markers to help finding reference points in images, what usually causes marker-less pose estimation to demand higher computational power, in comparison with marker-based methods. The techniques classified within this topic can be divided up into two main groups: image-based and model-based methods.

3.4.2.1 Image-based

Image-based methods use a reference image as comparison basis for all other images. The main application for such methods is, besides tracking systems, image merging to build mosaics. The reference image is previously registered and fully calibrated, so that rotation and translation from camera to the scene are completely known. Every frame is directly compared to the reference image using matching criteria, resulting in the estimation of that frame's pose related to the reference. For these methods, the model of transformation between actual frame and reference image usually comprises only linear camera movements, like rotation around a single axis or

simple rotation and translation combinations. According to the matching criteria used for measurement and comparison, the methods can be classified as intensity-based, feature-based or frequency analysis methods.

In intensity-based techniques grayscale images are directly compared using pixel values. These methods usually achieve higher accuracy values although requiring also higher computational power. Jurie and Dhome (JURIE, 2001) proposed a general framework for object tracking in video images, where the intensity difference between pixels belonging to the region being analyzed and pixels of the selected target, which is previously learnt during an offline stage, allows a straightforward prediction of the region position in the current frame. Shum and Szeliski (SHUM, 1998) implemented a technique for constructing full view panoramic mosaics from sequences of images, where a rotation matrix and a focal length are associated with every frame and matching errors across all possible overlapping image pairs are minimized using least-squares techniques.

In feature-based methods various image features are extracted, such as corners, edges or lines. Extracted features are compared against the set of features of a specific reference image, and correspondences are sought. Koch *et al.* (KOCH, 2005) describe a marker-less real-time tracking system for AR applications where sets of salient 2D intensity corners are detected in each image and then compared to a reference using the local feature matching operator implemented in the Shi-Tomasi-Kanade tracker (SHI, 1994), known as feature dissimilarity measure. Nielsen, Kramp and Grønbæk (NIELSEN, 2004) presented a mobile AR system, called SitePack, to support architects in visualizing on site 3D models in real-time. The system has a tracking algorithm which uses feature points correspondences. Zoghlami, Faugeras and Deriche (ZOGHLAMI, 1997) use geometric corners as features to merge sets of images and build 2D mosaics based on the computation of several homographies between sets of 2 images with around 50% overlapping areas. The algorithm can deal with arbitrary rotation and translation, given that the overlap is preserved. Li, Manjunath and Sanjit (LI, 1995) developed 2 contour-based methods to enable comparison and fusion of information from different image sensors. Ansar and Daniilidis (ANSAR, 2003) presented a general framework which allows for a set of linear solutions to the pose estimation problem, based on previous knowledge of points or lines correspondences in the image.

Frequency analysis methods use images' frequency spectrum as matching criterion. Stricker (STRICKER, 2001) proposed a hybrid algorithm using intensity-based and frequency analysis methods, which was implemented as the optical tracking solution of the ArcheoGuide project (HILDEBRAND, 2000). The image registration techniques used are the computation of a transformation minimizing intensity differences in the images and a Fourier-Mellin Transform (CHEN, 1994) (CASASENT, 1976), which is translation, rotation and scale invariant system. Reddy and Chatterji (REDDY, 1996) developed an intermediary technique between using only Fourier Transform and using the Fourier-Mellin version.

3.4.2.2 Model-based

Unlike image-based methods, model-based pose estimation algorithms try to obtain the current camera pose in 6 DoF without constraints on the camera movements

or the scene, which is analyzed for a match to known 3D models. 3D reconstruction is usually based on minimization of cost functions, which describe geometric relations between feature correspondences. Initially, 2D image features such as lines or points are identified in the scene, then features of known 3D model are projected on the 2D plane and matching points are searched for. If a match is found, the correct 3D pose is estimated. Since reconstruction is based on identification of several 3D feature points, robustness against occlusions and changes in pose is higher than in image-based methods.

Simon *et al.* (SIMON, 2000) (SIMON, 2002) described pose estimation algorithms given that one or more planes are visible. No prior knowledge of camera parameters is required. In these methods, planes in 3D space are projected in image space, and pose information for the scene can be extracted by using equations representing these planes. For initialization user must provide an initial guess for planes location in the image. The algorithm uses the Harris corner detector (HARRIS, 1988) and the RANSAC method (FISCHLER, 1981).

The algorithm proposed by Chia, Cheok and Prince (CHIA, 2002) uses two or more reference images with known camera pose. The camera pose to be estimated is calculated by matching natural features in the current frame to spatially separated reference frames and performing a minimization of two-view and three-view constraints between the frames, based on the camera position parameters. The calculation is stabilized using a recursive form of temporal regularization, similar to the Kalman filter. This method has the advantage of preventing a gradual increase in the camera position error.

Koller *et al.* (KOLLER, 1997) solved the problem of accurately estimating camera motion in a known 3D environment by tracking known landmarks – rectangular patterns attached to walls – through corner detection, making use of extended Kalman filter techniques, and then comparing them directly to the 3D model of the environment. Stricker, Klinker and Reiners (STRICKER, 1998) presented a model-based tracking algorithm based on detection of linear features of landmarks in the scene.

3.5 3D POSE ESTIMATION FROM 2 VIEWS (STEREO)

In order to estimate pose from 2 views, basically two problems must be solved. The first, the correspondence problem, consists in determining which item or feature in one image corresponds to which item or feature in the other image. The second problem, the 3D reconstruction, is basically the computation and interpretation of the projected distance - in the image plane - between two corresponding items, called disparity. A set of disparities for the scene – called disparity map – can be converted to a 3D map of the scene, if the geometry of the environment is known. Correspondence and 3D reconstruction issues are explained and discussed in next sections.

3.5.1 Basics

In this section, basic concepts for understanding issues behind stereo systems are explained. Figure 51 shows the top view of a stereo system composed of

two pinhole or perspective cameras (modelled as seen in section 3.1.2). The left and right image planes are coplanar, represented respectively by the segments I_L and I_R. O_L and O_R are centers of projection. The optical axes are parallel. For this reason, the *fixation point*, defined as the intersection point of the optical axes, lies infinitely far from the cameras.

The method used in stereo configurations to determine the position in space of P and Q is *triangulation*. By intersecting the rays defined by the centers of projection and the images of P and Q, i.e. the points p_L, p_R, q_L, q_R, the 3D position of P and Q can be obtained. The success of this technique depends on the correct choice of corresponding points on both images. Considering the correspondence problem as solved, the 3D reconstruction of e.g. point P is calculated from its projections p_l and p_r.

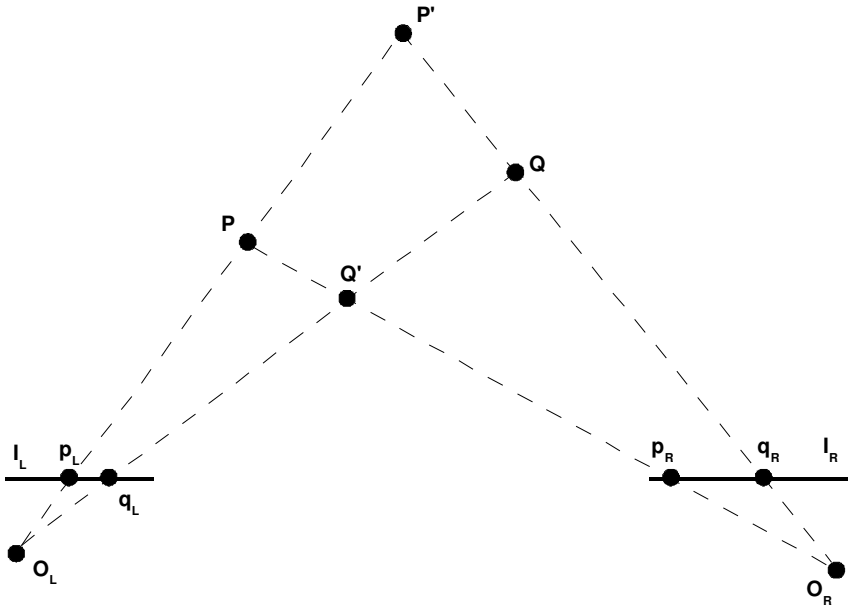

Figure 51 – Basic Stereo System Configuration (TRUCCO, 1998)

Figure 52 illustrates the reconstruction problem for point P. The distance T between the centers of projection O_L and O_R is called the *baseline* of the stereo configuration. Let x_L and x_R be the coordinates of p_L and P_R with respect to the principal points c_L and C_R, f is the focal length, and Z is the distance between P and the baseline. An equation for the similar triangles (p_L, P, P_R) and (O_L, P, O_R) can be written, yielding

$$\frac{T + x_L - x_R}{Z - f} = \frac{T}{Z}, \tag{72}$$

which can be solved for Z, resulting in

$$\begin{aligned} Z &= f \cdot \frac{T}{d} \quad , \\ d &= x_R - x_L \end{aligned} \tag{73}$$

where *d* is called disparity, a measure of the difference in image frame between the corresponding points in the two images. From (73) it can be noticed that the depth *Z* is inversely proportional to the disparity *D*. Actually, in a typical stereo system with optical axes that intersect in a fixation point at a finite distance from the cameras, disparity increases with the distance of the objects from the fixation point. This point, in the case shown above, lies in an infinite distance from the cameras.

In a stereo configuration, the intrinsic parameters are exactly as described in section 3.1.3. If one camera is considered as the reference frame, then the extrinsic parameters define the relative position and orientation of the other camera in relation to the first one, and definitions of section 3.1.4 are valid.

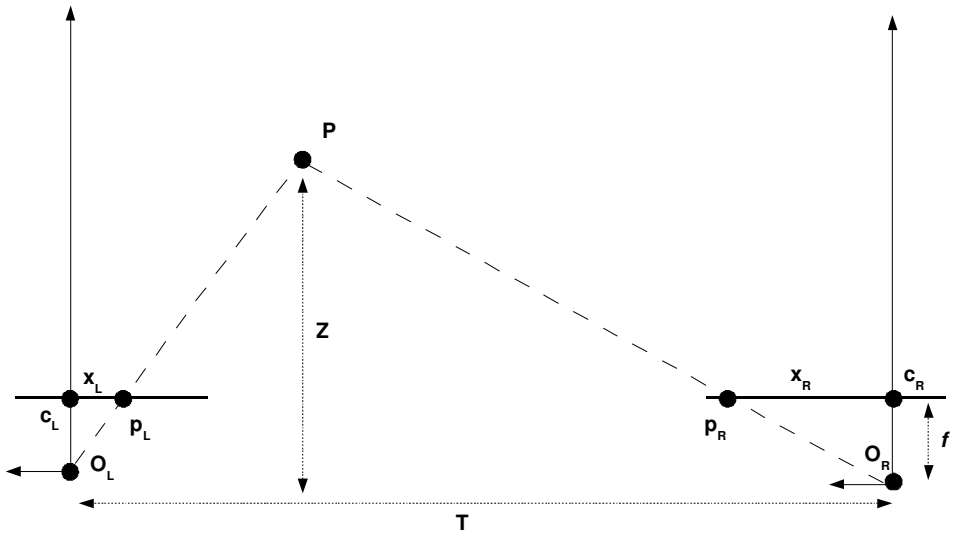

Figure 52 – 3D Reconstruction of Point P in Stereo Configuration (TRUCCO, 1998)

A variant of stereo vision is the use of only one moving camera grabbing images in different time instants, composing a set of two or more views of the same scene. If the structure of camera motion is known, it is then possible to extract 3D information from that set of views, what is called Temporal Stereo Vision.

3.5.2 Correspondence

The correspondence problem can be seen as a search problem, namely which element in one image corresponds to which element in the other image. It must be defined what kind of image element will be matched and which similarity measure will be adopted. Correspondence algorithms can be classified as correlation-based and feature-based. While correlation-based methods apply to the totality of image points, feature-based methods attempt to establish correspondences between smaller sets of image features.

In correlation-based methods, the elements to match are image windows of a defined size, and the similarity criterion is a measure of the correlation between windows in the two images. The correct corresponding element is given by the window that maximizes the correlation measure. Examples of correlation methods were

presented by Marr and Poggio (MARR, 1976) and Lucas and Kanade (LUCAS, 1981). Kanade and Okutomi (KANADE, 1991) presented ways of adapting the shape and size of the correlation windows to different image parts.

In feature-based techniques, the search for correspondences is restricted to a sparse set of features. Numerical and symbolic properties of features are used instead of image windows. Examples of features are edge points, lines and corners, which are detected using algorithms as shown in sections 3.2.4, 3.2.5 and 3.2.6. These methods use a measure of the distance or similarity between features to find corresponding elements. Geometric or analytical constraints, as the epipolar constraint shown in section 3.5.3, can be used to narrow down the number of possible matches for each feature, increasing search speed. Ayache and Faverjon (AYACHE, 1987) presented a technique based on edge segments correspondence.

Correlation-based methods are easier to implement, providing dense disparity maps, although demanding high computational power. Feature-based techniques are suitable if there is previous knowledge of the scene, enabling the choice of the right feature detector. They are also faster and less sensitive to illumination changes, but provide sparser disparity maps, although these could be enough information for the application. Generally speaking, correspondence algorithms have their performance worsened by occlusions (points in one image with no corresponding points in the other image) and spurious matches (false correspondences).

Jones and Malik (JONES, 1992) proposed a correspondence method based on orientation and spatial frequency disparities, which works in absence of direct positional correspondence and does not use correlation techniques. Scharstein and Szeliski (SCHARSTEIN, 2002) recently presented a comprehensive taxonomy study about stereo correspondence algorithms.

3.5.3 Epipolar Geometry

The basic geometry of a stereo configuration, known as *epipolar geometry*, is shown in Figure 53, where two pinhole or perspective cameras are represented, together with their projection centers, O_L and O_R, and image planes, π_L and π_R.

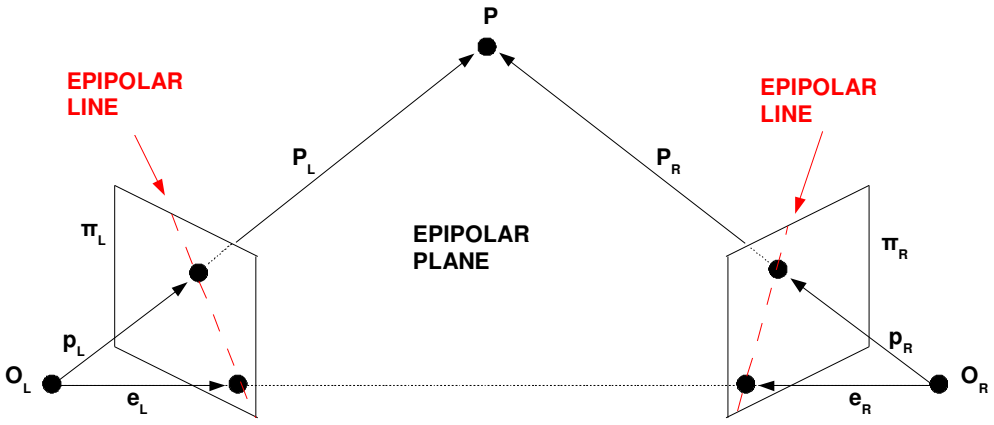

Figure 53 – Basic Representation of the Epipolar Geometry (TRUCCO, 1998)

Each camera identifies a 3D coordinates system, the origin of which coincides with the projection center, and the z axis with the optical axis. The focal lengths are denoted by f_L and f_R. Vectors $P_L = [X_L, Y_L, Z_L]^T$ and $P_R = [X_R, Y_R, Z_R]^T$ refer to the same 3D point P represented in the left and right camera reference frames, respectively. Vectors $p_L = [x_L, y_L, z_L]^T$ and $p_R = [x_R, y_R, z_R]^T$ refer to the projections of P onto the left and right image planes, respectively, expressed in the corresponding reference frame. For all the image points, equations $z_L = f_L$ and $z_R = f_R$ are valid.

The extrinsic parameters of the stereo system relate the reference frames of the left and right cameras, by defining a rigid transformation in 3D space, described by a translation vector $T = (O_R - O_L)$ and a rotation matrix R. Given an arbitrary point P in space, the transformation between P_R and P_L is a translation followed by a rotation, represented by

$$P_R = R \cdot (P_L - T). \tag{74}$$

The expression *epipolar geometry* is used because the points at which the line through the centers of projection intersects the image planes are called *epipoles*, denoted by e_L and e_R. The left *epipole* is the image of the right camera's projection center, as well as the right *epipole* is the image of the left camera's projection center.

As in (6), the relation between the 3D points and their projections are described in vector form by

$$\begin{aligned} p_L &= \frac{f_L}{Z_L} \cdot P_L \\ p_R &= \frac{f_R}{Z_R} \cdot P_R \end{aligned}. \tag{75}$$

The intersection between the plane defined by the points P, O_L and O_R, the so called *epipolar plane*, and each image is a line, called *epipolar line*. Considering P and the vector p_l, for instance, P can lie anywhere on the ray from O_l through p_l, but, since the image of this ray in the right image is the epipolar line through the corresponding point p_r, the correct corresponding point must lie on the epipolar line. This condition is known as the *epipolar constraint*, and establishes a mapping between points in the left image and lines in the right image, and *vice versa*. This is used to restrict the search for the match of p_l along the corresponding epipolar line, reducing the search for correspondences to a 1D problem. Alternatively, the same method can be used to verify whether or not a candidate match lies on the corresponding epipolar line. This procedure is usually utilized to reject false corresponding points due to occlusions.

The epipolar geometry can be determined using the concepts of essential and fundamental matrices.

3.5.4 The Essential Matrix

The epipolar plane through P (Figure 53) can be represented by the equation describing the coplanarity condition of the vector P_L, the translation vector between cameras, T, and the vector $P_L - T$, written as

$$\left(P_L - T\right)^T \cdot \left(T \times P_L\right) = 0\,. \tag{76}$$

Using (74) and the fact that R is an orthogonal matrix ($R^{-1} = R^T$), one can write

$$\left(R^T \cdot P_R\right)^T \cdot \left(T \times P_L\right) = 0\,. \tag{77}$$

where $T = \left[T_x, T_y, T_z\right]^T$. Considering the fact that a vector product can be represented as the multiplication of one vector by a matrix, one can write

$$T \times P_L = S \cdot P_L\,, \tag{78}$$

where

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}. \tag{79}$$

Substituting (78) in (77), and recalling that

$$\left(A \cdot B\right)^T = B^T \cdot A^T\,, \tag{80}$$

results in

$$P_R{}^T \cdot R \cdot S \cdot P_L = P_R{}^T \cdot E \cdot P_L = 0\,, \tag{81}$$

where

$$E = R \cdot S \tag{82}$$

is the *essential matrix* and establishes a direct relation between the epipolar constraint and the extrinsic parameters (section 3.1.4) of the stereo configuration. Using (75) and dividing the result by $Z_R \cdot Z_L$ yields

$$p_R{}^T \cdot E \cdot p_L = 0\,, \tag{83}$$

what shows that the essential matrix E is a mapping between points in both left and right projective planes.

Equation (82) shows that the epipolar geometry, represented by the essential matrix, can be calculated from the extrinsic parameters of the stereo configuration. Using the calibration methods presented in section 3.3, one can obtain the extrinsic parameters of the stereo system simply by combining the extrinsic parameters of each camera, initially calculated in relation to the same reference point. Considering T_L, R_L, T_R and R_R the extrinsic parameters of the two cameras in the world reference frame, the extrinsic parameters of the stereo system, T and R, are obtained (see (TRUCCO, 1998) for the derivation) from

$$\begin{aligned} R &= R_R \cdot R_L^T \\ T &= T_L - R^T \cdot T_R \end{aligned} . \tag{84}$$

3.5.5 The Fundamental Matrix

Let M_{int_L} and M_{int_R} be the matrices representing the intrinsic parameters (section 3.1.3) of the left and right cameras respectively. Let $\overline{u_L}$ and $\overline{u_R}$ be the points in pixel coordinates corresponding to p_L and p_R, that can be related by

$$\begin{aligned} p_L &= \left(M_{int\_L}\right)^{-1} \cdot \overline{u_L} \\ p_R &= \left(M_{int\_R}\right)^{-1} \cdot \overline{u_R} \end{aligned} . \tag{85}$$

Substituting (85) in (83), and using (80), yields

$$\left(\overline{u_R}\right)^T \cdot \left[\left(\left(M_{int\_R}\right)^{-1}\right)^T \cdot E \cdot \left(M_{int\_L}\right)^{-1}\right] \cdot \overline{u_L} = \left(\overline{u_R}\right)^T \cdot F \cdot \overline{u_L} = 0 , \tag{86}$$

where

$$F = \left(M_{int\_R}\right)^{-T} \cdot E \cdot \left(M_{int\_L}\right)^{-1} \tag{87}$$

is called the *fundamental matrix*, a mapping between points of both cameras directly in pixel coordinates. Equation (87) shows that the epipolar geometry can be reconstructed based only on some point matches in pixel coordinates, without previous knowledge of intrinsic or extrinsic camera parameters.

The fundamental matrix can be obtained by combining the projection matrices (section 3.1.5) of both cameras, which are initially related to the same reference point. As aforementioned in section (3.3), there are calibration methods that estimate directly the projection matrix, like the *eight-point algorithm*, proposed by Longuet-Higgins (LONGUET-HIGGINS, 1981).

3.5.6 3D Reconstruction

If intrinsic and extrinsic parameters of a stereo configuration are known, the epipolar geometry is also known. In this case, the 3D reconstruction problem can be unambiguously solved by triangulation. It can be seen in Figure 53 that the point P, whose projections are the corresponding points p_L and p_R, lies at the intersection of the two rays from O_L through p_L and from O_R through p_R. If the epipolar geometry is entirely known, the rays are known and their intersection can be computed.

However, since the stereo configuration parameters are known only approximately, the two rays will not exactly intersect in space, but will be very close to each other. Thus, the intersection can be considered as the midpoint of a vector connecting both rays (or vectors) on the minimum distance point between them.

The basic geometry of the problem is shown in Figure 54.

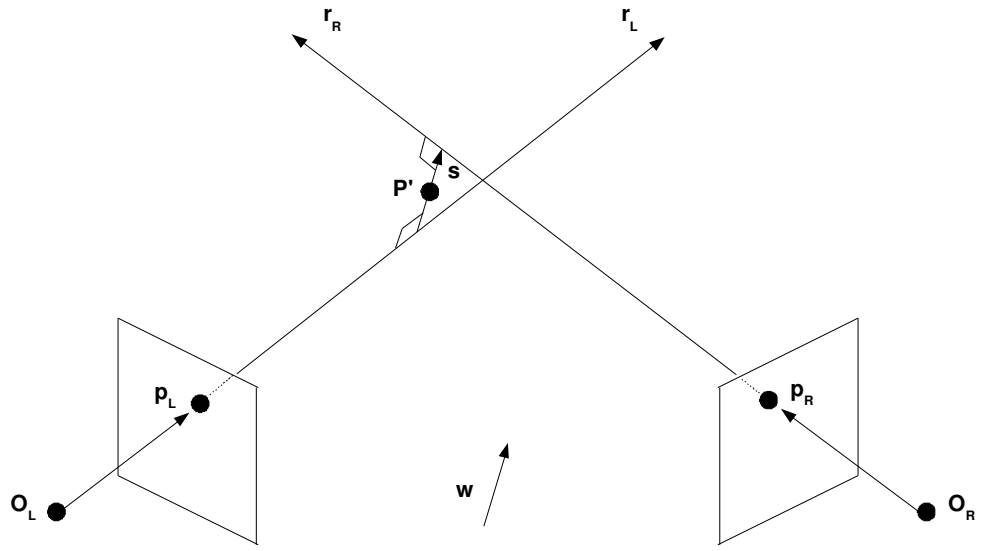

Figure 54 – 3D Reconstruction of Point P using Calculation of Intersection Point by Triangulation (TRUCCO, 1998)

Let the left camera origin point O_L be the world coordinates system origin. Let $k_1 \cdot p_L$ be the vector r_L, from O_L through p_L, and $T + k_2 \cdot R^T \cdot p_R$ the vector r_R, from O_R through p_R. Let *w* be a vector orthogonal to r_L and r_R, obtained from their vector product, expressed by $k_3 \cdot \left(p_L \times R^T \cdot p_R\right)$. Let *s* be a vector parallel to *w* and through r_L, represented by $k_1 \cdot p_L + k_3 \cdot \left(p_L \times R^T \cdot p_R\right)$, connecting r_L and r_R. The midpoint P', which is an approximation of P, can be calculated by first obtaining the endpoints of *s* and then averaging. The endpoint where *s* joins r_R can be expressed as $s = r_R$ or

$$k_1 \cdot p_L - k_2 \cdot R^T \cdot p_R + k_3 \cdot \left(p_L \times R^T \cdot p_R\right) = T \cdot \tag{88}$$

Since R, T, p_L and p_R are known, a linear system can be built with 3 unknowns and 3 equations (p_R and p_L are given in 3D camera coordinates), which can

be directly solved for k_1, k_2 and k_3. Since the endpoints of s are $k_1 \cdot p_L$ and $T + k_2 \cdot R^T \cdot p_R$, the midpoint P' can be easily calculated.

3.6 INTERFACE WITH VR/AR APPLICATIONS: FRAMEWORKS

Due to the large number of existing input and output devices used in virtual environments, their integration in VR and AR applications has become a task of increasing complexity.

A possible solution is the definition of frameworks, structures that provide standard interfaces for applications and devices, so that individual changes in those parts can be done independently. This is called device abstraction. He and Kaufman (HE, 1993) implemented the *Device Unified Interface* (DUI), which is a protocol for communicating between applications and input devices that achieves full independence between those parts by allowing users to interactively control the device operations and modify the device configuration with no effect on the application. Figure 55 shows a representation of the transparent relation provided by DUI or any framework which achieves device abstraction.

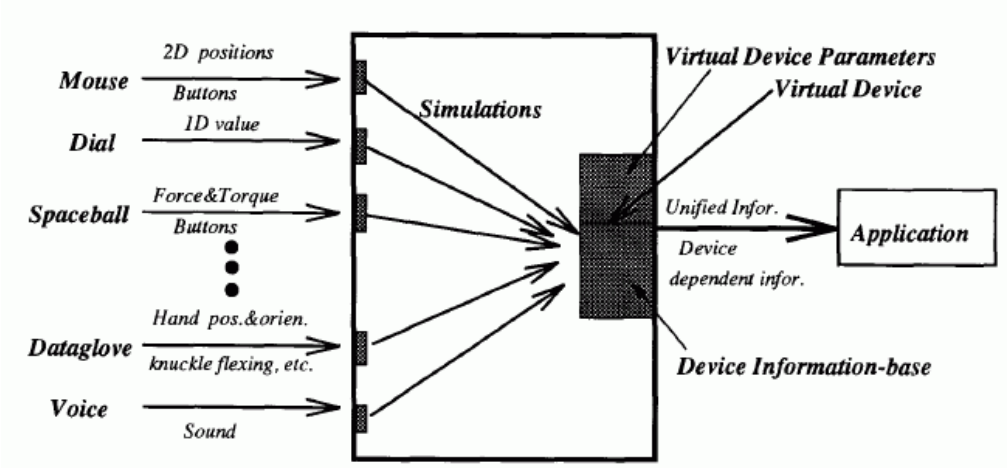

Figure 55 – Structure of the Device Unified Interface (DUI) (HE, 1993)

Shaw *et al.* (SHAW, 1993) presented the Minimal Reality (MR) Toolkit, a middleware solution initially designed to allow for parallel computing in virtual environments, which takes advantage of the distributed computing capabilities of workstation networks, improving the application's performance. The MR Toolkit provides a framework for developing new interaction techniques or AR/VR user interfaces, and for integrating them with existing applications without developing large amounts of extra software. Blach *et al.* (BLACH, 1998) created the Lightning VR system, which supports device independence but also focus on rapid behavioral prototyping for applications. The VR Juggler system (CRUZ-NEIRA, 2002) is an open source solution for developing and executing device independent VR/AR applications, which aims at introducing a software layer between the application and the hardware to provide a hardware-independent software system.

Also distributed VR/AR systems that allow collaborative working can be classified as frameworks, since device and also application independence is achieved. The MASSIVE (GREENHALGH, 1995) and the DIVE (CARLSSON, 1993) systems are examples of virtual environment systems focused on multi-user, distribution and human-computer interaction.

Recently proposed solutions merge the idea of device independence and collaborative working. The Avocado or Avango framework (TRAMBEREND, 1999) introduced the concept of shared scene-graph, where every user or process has access to complete information about the scene. As most of these frameworks, Avango uses an object-oriented structure. Avango is nowadays a commercially available solution. Opentracker (REITMAYR, 2001) is an open source library developed to be a unified tracking interface, allowing application access and fusion of input from different tracking systems over a sole data exchange protocol. The system uses configuration scripts with syntax based on XML. OpenTracker originally supports ARToolKit (section 2.6.1).

There are more comprehensive frameworks for AR applications, like the DWARF (Distributed Wearable Augmented Reality Framework) system, described in (BAUER, 2001), and the Studierstube Augmented Reality project (SCHMALSTIEG, 2002). These frameworks comprise not only device abstraction, but also application development.

3.7 SENSOR FUSION FOR MULTIPLE-CAMERA TRACKING SYSTEMS

Multisensor integration is defined as the synergistic use of information provided by multiple sensory devices to assist in the accomplishment of a task by a system (LUO, 1990). The potential advantages gained through the collaborative use of multisensory information can be decomposed into a combination of 4 fundamental aspects: redundancy, complementarity, timeliness and cost of the information.

Redundant information is provided from a group of sensors (or a single sensor over time) when each sensor is perceiving, possibly with a different exactness, the same features in environment. The fusion of redundant information can lead to reduction of overall uncertainty and thus serves to increase accuracy of the sensed information. Multiple sensors providing redundant information can also serve to increase reliability in the case of sensor failure or adverse measurement conditions. Complementary information from multiple sensors allows features to be perceived that are impossible to perceive only using the information from each individual sensor operating separately. More timely information may be provided by multiple sensors, in comparison to the speed at which it can be provided by a single sensor, due to either the actual speed of operation of each sensor or the processing parallelism that may be possible to achieve as part of the integration process. Multiple sensors may be capable of providing information at a lower cost when compared to a single sensor.

Practically, multisensor fusion refers to any stage in the integration process where there is an actual combination of different sources of sensory information into one representational format. Multisensor data fusion techniques can fuse information from complementary or redundant sensors or even from a single sensor over a period of time. Depending on sensors and goals of the multisensor data fusion techniques, the

latter can be divided into four levels, according to (LUO, 1990): signal, pixel, feature and symbol. Signal-level fusion decreases the covariance of the sensory data. Pixel-level fusion is intended to increase the information content associated to each pixel of an image. Feature-level fusion combines features derived from signals or images into meaningful representations of more reliable features. Symbol-level fusion allows information to be fused at the highest level of abstraction and it is usually used in decision-based systems. Figure 56 shows examples of this classification applied to the case of the automatic recognition of a tank.

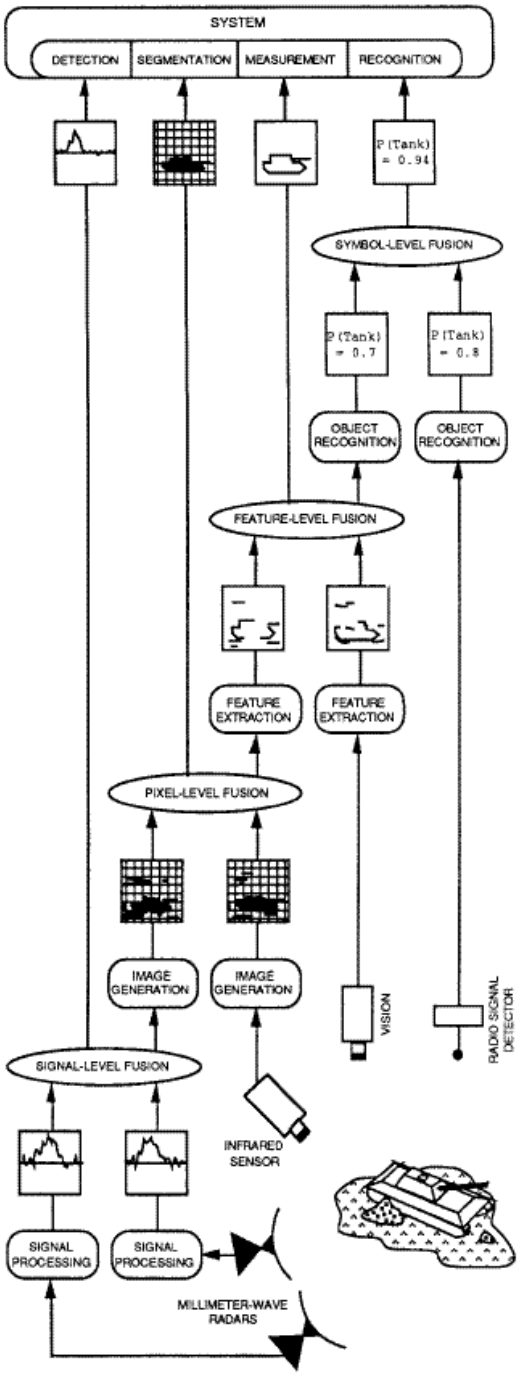

Figure 56 – Possible Uses of Signal, Pixel, Feature and Symbol-level Fusion (LUO, 1990)

Nicosevici *et al.* (NICOSEVICI, 2004) proposed another classification, focused on sensor fusion methods for underwater vehicle navigation, dividing sensor fusion techniques into four main categories, based on applications related to Unmanned Undersea Vehicles (UUV): filtering and estimation, mapping-oriented, behaviour-oriented and machine learning. Among these categories, filtering and estimation comprises the most basic sensor fusion techniques, whose purpose is mainly the estimation of one measurement based on several sensors in different operating conditions, thus with different levels of data fidelity.

In practice, when using similar sensors, it is sufficient to apply only one level of fusion. In more complex cases, when different groups of sensors are involved, each group is fused at a lower level, and sensory data coming from different groups is fused at a higher level into one representational format.

3.7.1 Sensor Fusion Algorithms

Algorithms to fuse information between sensors may vary in complexity and performance. One of the simplest and most intuitive general methods of fusion is the weighted average. First, redundant information provided by a group of sensors is filtered to eliminate spurious measurements, and then a weighted average of the information is taken and used as the fused value. While this method allows for the real-time processing of dynamic low-level data, in most cases the Kalman filter (KALMAN, 1960) is preferred because it provides a method that is nearly equal in processing requirements and results in estimates for the fused data that are optimal in a statistical sense.

The Kalman filter is used in a number of multisensor systems when it is necessary to fuse dynamic low-level redundant data in real time. The filter uses statistical characteristics of the measurement model to iteratively determine optimal estimates for the fused data. If the system can be described with a linear model and both system and sensor error can be modelled as white Gaussian noise, the Kalman filter will provide unique statistically optimal estimates for the fused data. The filter functions iteratively, operating in a cycle of prediction and adjustments. Extensions and variants of the Kalman filter have been proposed and widely used in applications ranging from GPS to financial market prediction.

For a comprehensive review, readers are referred to (LUO, 1999), which includes several frequently used methods as the Bayesian Theorem and the Dempster-Shafter Evidence Theory. Faceli, Carvalho and Rezende (FACELI, 2002) investigate artificial intelligence techniques for sensor fusion, consisting of Machine Learning (ML) methods, such as Artificial Neural Networks (ANNs). Ding *et al.* (DING, 2004) present a sensor fusion method for target identification using Dempster-Shafter evidence theory and based on fuzzy logic theory. Koval (KOVAL, 2001) presented a new competitive sensor fusion algorithm, which implements pixel-level fusion.

Thomopoulos (THOMOPOULOS, 1994) addresses the issue of sensor selectivity in the design of a fusion system, by incorporating data quality control and error detection capabilities in the fusion system design. Jin and Lee (JIN, 2002) proposed a spatial-temporal sensor fusion technique, where data sets of previous time

instants are properly transformed and fused into the current data sets, in order to obtain accurate measurements, resulting in improved exactness even with few sensors.

3.7.2 Large Area Tracking

Large area tracking can be achieved by using multiple sensors. An optical large area tracker can be built using multiple stereo or single camera trackers. In this case the main exploited advantages of sensor fusion are redundancy and complementarity. By using redundant multiple trackers, higher robustness against occlusion problems and higher exactness can be obtained, as well as the possibility of tracking large areas, what would be impossible to do with just one tracker. Scalability is also automatically obtained, since a multiple camera tracker can be indefinitely expanded.

Figure 57 shows an example of a marker-based multiple-camera tracking system, which could be built with stereo or single camera configurations.

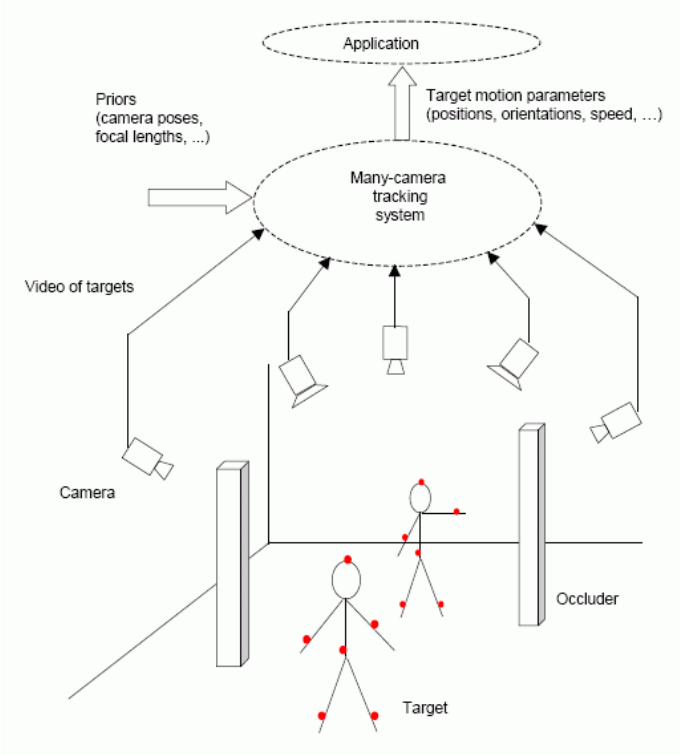

Figure 57 – Schematic of a Multiple Camera Tracking System (CHEN, 2002)

A multiple stereo system is based on combinations between any two cameras, each pair composing one stereo tracker. Data from all stereo trackers are then merged. A multiple single camera tracker system is based on the fusion of information from all single trackers.

Chen (CHEN, 2002) implemented M-Track, a many-camera scalable system architecture for real-time motion tracking, using one processor per camera and a central estimator based on extended Kalman filtering, what allows asynchronous input from

multiple cameras as well as smooth, incremental integration of information from local camera-processor pairs. This extension to Kalman filter provides a solution for linearizing non-linear dynamics or non-linear measurement relationships, what is common in many practical applications. The system implements a feature-level data fusion, according to the classification described in section 3.7.1. Figure 58 shows the basic M-Track architecture.

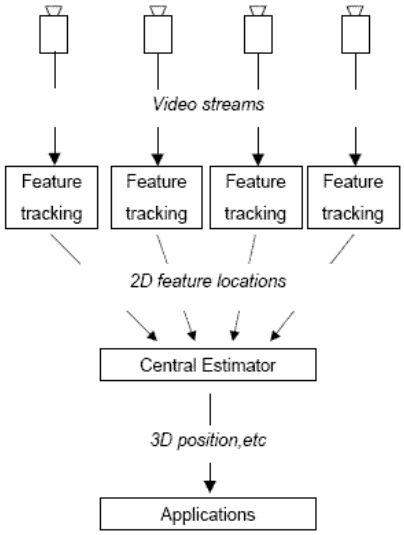

Figure 58 – Architecture of the M-Track System (CHEN, 2002)

Dockstader and Tekalp (DOCKSTADER, 2001) proposed a distributed, real-time computing platform for improving feature-based tracking of multiple interacting persons in the presence of articulated motion and occlusion, with the goal of target recognition. Both spatial and temporal data integration is performed within a unified framework of 3D position tracking to provide increased robustness to temporary feature point occlusion. The system implements a feature-level data fusion and employs a probabilistic weighting scheme for spatial data integration as a simple Bayesian belief network (BBN) with a dynamic, multidimensional topology. This corresponds to the selective use of multiple views of particular features based on measures of spatial-temporal tracking confidence. The system uses also a Kalman filter, in its final processing stage, to maintain a level of temporal smoothness on the vector of 3D trajectories. Figure 59 shows the basic flow diagram of the system.

Yonemoto *et al.* (YONEMOTO, 1999) implemented a similar multiple camera system using color markers, feature-level data fusion and stereo vision.

Calibration of multiple camera systems can be easily made if individual intrinsic and extrinsic parameters of the cameras are known (see section 3.3 for ordinary calibration algorithms). Basically, a common reference coordinates system must be established for all cameras, in relation to which the extrinsic parameters of the cameras are obtained. In sequence, the transformation or essential matrix (section 3.5.4) of each camera must be calculated in relation to the global reference. Finally, each individual transformation matrix is used during the data fusion step.

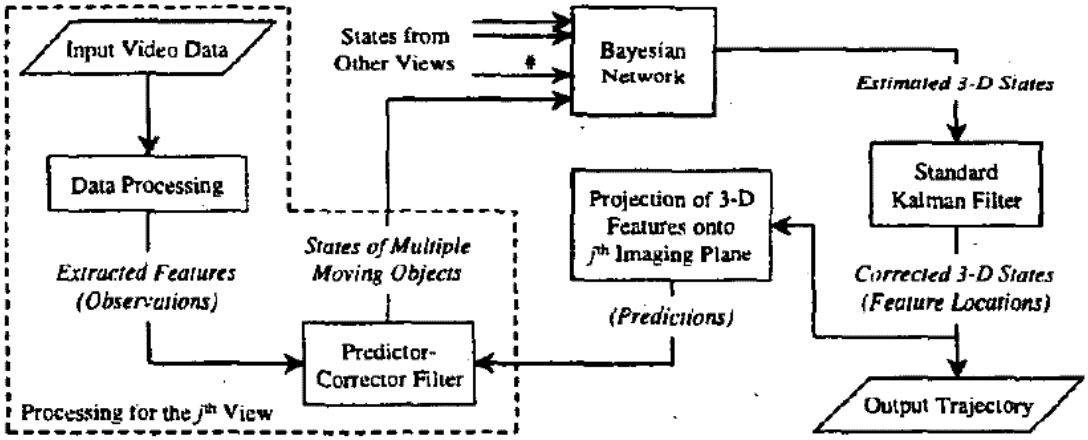

Figure 59 – Dockstader and Tekalp's System Block Diagram (DOCKSTADER, 2001)

Automatic calibration is also an option, especially in systems where cameras may be freely and frequently moved. For example, Remagnino and Jones (REMAGNINO, 2002) developed an automatic calibration procedure for multiple camera surveillance systems. In this method, extrinsic parameters are initially estimated in relation to a specific *ground plane coordinate system* for each camera, and then a Hough Transform approach is used to merge the set of camera-specific ground planes together.

4 IMPLEMENTED TRACKING SYSTEM

In this chapter the marker-based optical tracking system developed in this work is presented. Initially the one-camera tracking module is described in detail, followed by a description of the large area tracking solution. After that, the developed and adapted calibration procedures are presented.

4.1 ONE-CAMERA TRACKING MODULE

The system uses one-camera tracking modules to build a large area tracking system. Each module consists of a digital camera attached to a computer, where image pre-processing and an adapted version of the PTrack algorithm are executed. Figure 60 shows a block diagram of the one-camera tracking module.

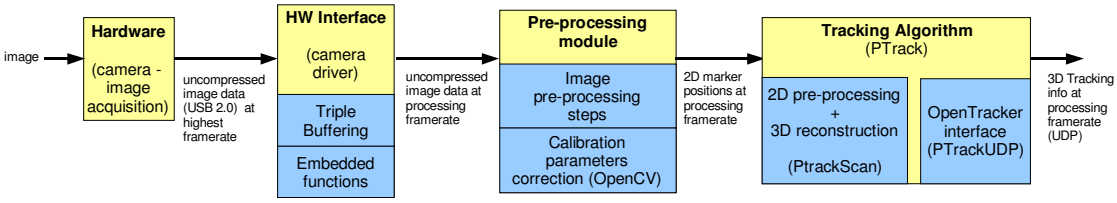

Figure 60 – Block Diagram of the One-camera Tracking Module

The one-camera tracking module consists of the camera itself, a hardware interface block, a pre-processing module and the PTrack module. In the original implementation of PTrack (section 2.6.2) the hardware interface block and the pre-processing module are both physically integrated within the ART camera, while PTrack runs on an ordinary PC. This block architecture is only feasible when the camera has embedded image processing functions, in which case it is called a "smart camera". Since an option was made to use regular cameras without embedded processing, due to high costs of cameras with such capabilities (see section 4.1.1), the hardware interface and pre-processing modules had to be physically separated from the image acquisition module. Since a computer was already required for PTrack, the natural choice for the new system was to integrate hardware interface and pre-processing modules into the same computer.

During the first experiments of the implemented system, the pre-processing module was not optimized enough, resulting in poor system performance when all three modules were simultaneously executed in the same computer. After gradual improvements it became possible to run the three modules together, with adequate performance results. The adopted architecture, showing physical separation between modules, is represented in Figure 61.

The following sections describe in more detail each part of the one-camera tracking architecture.

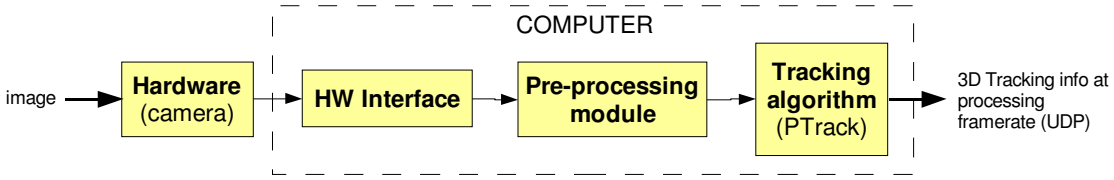

Figure 61 – Block Architecture of the One-camera Tracking Module

4.1.1 Hardware

According to the specifications presented in section 2.8, the new system must use affordable hardware components. A comprehensive research on existing cameras was conducted. The so called "smart cameras", which have embedded hardware for image processing, as the CANCam camera by Feith[1] or the Iris P-Series cameras by Matrox[2] were initially considered for the new system, but were later discarded due to high costs. An equally extensive research was made in order to choose the best matching pair of infrared LEDs for the flash strobes and daylight blocking filter for the camera's lenses.

The hardware module which was built consists of an IDS uEye UI1210-C camera (Figure 62) with a 640 x 480 resolution CMOS sensor, global shutter, attached to a C815B (TH) Pentax lens with 8.5 mm focal length and 56.5 deg FoV, configured to acquire grayscale images up to 55 frames per second, equipped with infrared flash strobes (940 nm wavelength) controlled by a digital trigger output of the camera.

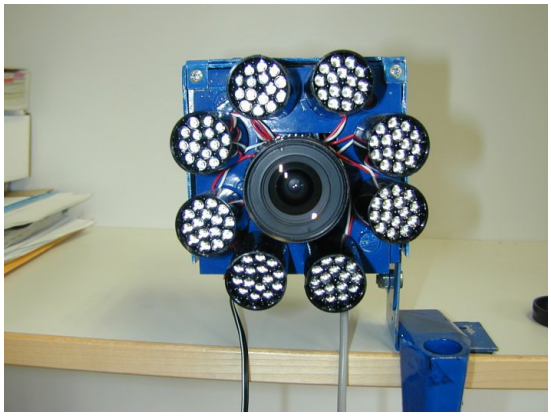

Figure 62 – IDS uEye UI1210-C Camera Equipped with Infrared Flash Strobes

[1] Feith Sensor to Image GmbH. CANCam camera website available at <http://www.feith.de/cancam.html>. Last accessed on December 16th, 2005.
[2] Matrox Electronic Systems Ltd., Iris P-Series cameras website available at <http://www.matrox.com/imaging/ products/iris_pseries/home.cfm>. Last accessed on December 16th, 2005.

The flash strobes consist of 8 modules of 15 IR LEDs each, totalizing 120 IR LEDs around the lens. Initially only 2 IR LED modules were built, but sufficient illumination level and homogeneity were not achieved, what was inferred based on empirical results. Thus, in order to obtain enough and uniform illumination, additional modules were gradually included until 8 modules were built, which is the maximum number that fits the camera frame. In this case, higher illumination levels result in higher camera reach, and higher illumination uniformity results in higher system robustness.

Between the lens and the CMOS sensor a RG-850 (850 nm) visible light blocking filter (a highpass filter with cutoff frequency corresponding to 850 nm) was built to minimize interference from undesired optical sources in daylight spectral region. The flash strobes have an independent DC power supply with 6 V output, while the camera is directly supplied by the USB connection. The complete hardware module costs around EUR 700.

At each frame the camera captures the image of labels and sends the uncompressed data to a PC through a USB 2.0 interface, which has a maximum transfer speed of 480 Mbit/s. With a maximum camera frame rate of 55 Hz, at 640 x 480 resolution and using grayscale images, the minimum required transfer speed in the interface is 135.2 Mbit/s. Since USB 1.1 interfaces reach up to 12 Mbit/s, a USB 2.0 interface is required. Also IEEE 1394 interface could be used, since it provides up to 393 Mbit/s (Firewire 400 or S400) or 786 Mbit/s (Firewire 800 or S800 - IEEE 1394b), if the camera had this built-in interface.

The labels used by PTrack are composed of six markers each (Figure 63). Four markers represent the corners of a square label with fixed edge length of 80 mm. One marker is on the top edge, identifying the top orientation of the label, splitting the edge length in 1/3 and 2/3 segments. The sixth marker is used for coding and must lie somewhere within the square formed by the others.

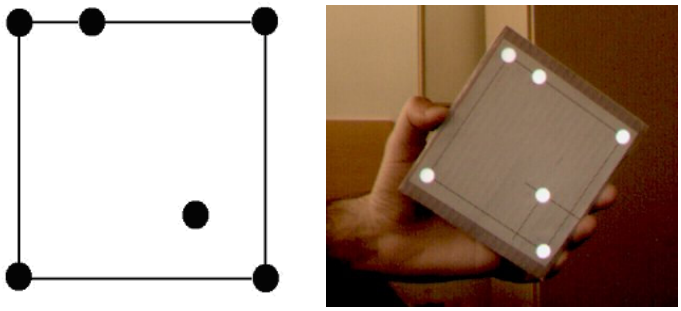

Figure 63 – Typical Label Layout for PTrack with 6 Retro-reflective Markers

The markers used are flat circular retro-reflective tags with diameter of 10 mm. Considering this diameter of markers, the camera can recognize them from distances varying from 60 cm up to 3.5 m. Given the camera's 56.5 deg FoV, the resulting working range with one camera is approximately 16 m³.

Regarding the type of markers employed, an attempt was made to use active markers in the system. Figure 64 shows a prototype of a non-tethered 2-marker label,

built with infrared LEDs and ordinary batteries, as well as a prototype of an active marker label for PTrack besides the actually implemented passive marker label.

Active markers were the first option for the implementation because it was assumed that the reach of active markers (LEDs, thus emmited light) was higher than passive markers (only reflected light). This assumption was confirmed. However, passive markers were preferred because they are non-tethered, thus allowing higher freedom of movement to the user. Another drawback of passive markers was CMOS sensor saturation when any of the active markers was aiming directly at the lenses, due to directivity properties of the LEDs.

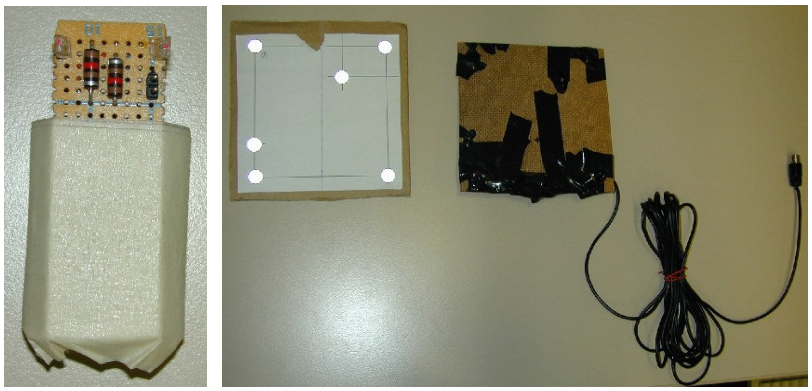

Figure 64 – Active Marker Prototypes: 2-marker Non-tethered Label (left); Passive and Equivalent Active Marker Tethered Label for PTrack (right)

4.1.2 Hardware Interface

A computer is needed to run image pre-processing tasks as well as the tracking algorithm. There is no need for framegrabbers, since the video signal is digitally recorded and transferred to the computer. Instead, only a hardware interface module must exist in order to provide the pre-processing algorithms with images from the camera. The computer processing power determines the performance of the tracking system. For example, a Pentium Centrino 1.4 GHz with 512 MB RAM was used for the development tests, which allowed overall frame rate up to 35 Hz.

In any digital camera, the image grabbing and transfer process has a duration defined by the sum of exposure time and transfer time. According to the Hardware User Manual of IDS uEye UI1210-C (IDS UEYE, 2004), the exposure time is the reciprocal value of the frame rate, thus 18.2 ms at 55 Hz. The transfer time, using the same frame rate, is 15.4 ms. So, the total time for image acquisition and transfer is 33.6 ms.

At first, in this work, an attempt was made to only acquire a new frame when the processing tasks of the last frame were done. This means to sum up the 33.6 ms in the total processing time, serializing the tasks. In this case, only one image buffer is needed in the PC's RAM, where image is saved during transfer.

Afterwards, attempts were made to execute tasks concurrently, using pipelining techniques. Initially 2 image buffers were created in the PC, number later

increased to 3, accessed by lock/unlock mechanisms.While image grabbing and transfer of the next frame use the 1st or the 2nd image buffer, image pre-processing tasks and tracking algorithm run on the 3rd image buffer. If image grabbing and transfer are much faster than other tasks, old images are discarded and replaced with new ones. It is always guaranteed that, when image pre-processing and tracking tasks are finished, at least one other image buffer will be ready to be processed. Besides, image grabbing and transfer are totally decoupled from image pre-processing and tracking tasks, being usually faster and having more stable update rate. This technique is called Triple Buffering and was implemented with help of functions already embedded in the device driver provided with the IDS uEye camera. Using this method, the throughput of the system equals the overall update rate and is directly related to the total time taken by image pre-processing and tracking algorithms to perform. Figure 65 depicts some steps of the Triple Buffering method.

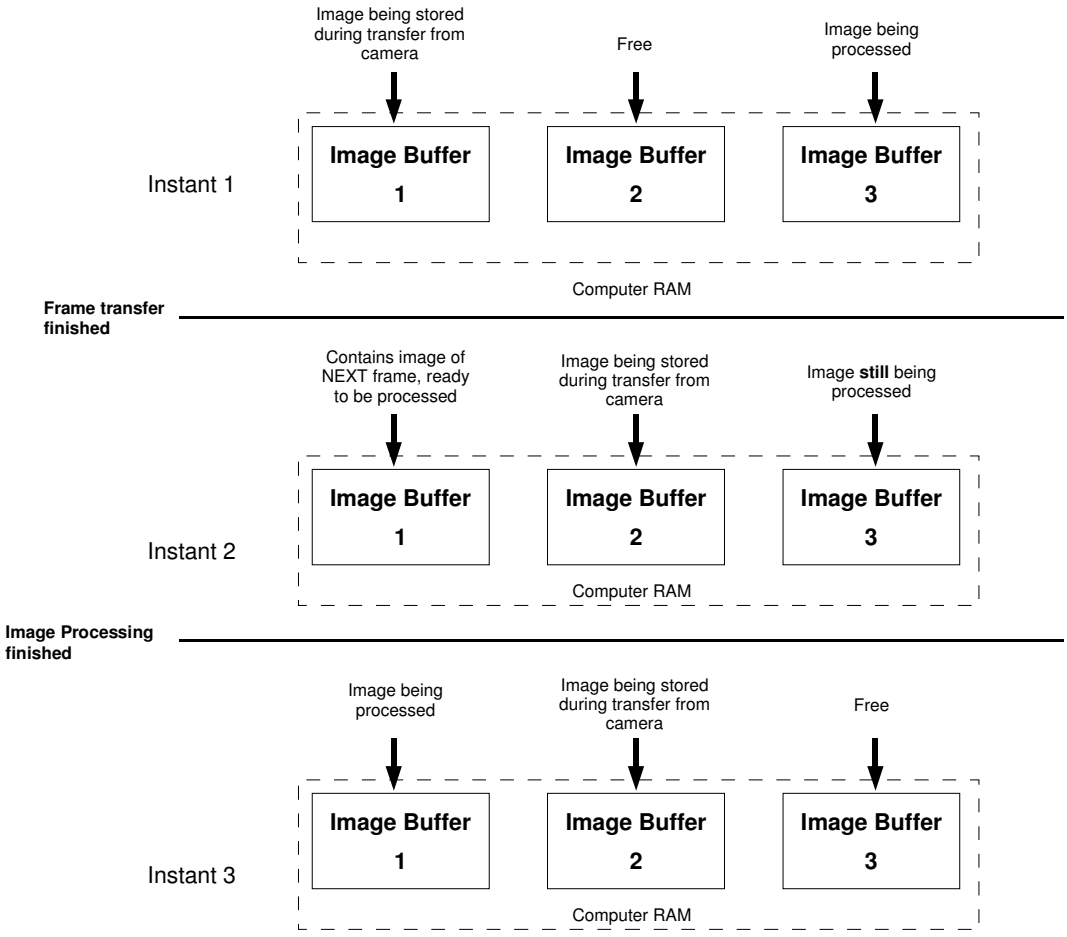

Figure 65 – Triple Buffering Technique

In this case, latency of the system is given by the expression

$$Latency = T_{image\_acquisition} + T_{image\_transfer} + (1..2) \times T_{image\_pre-processing+tracking}\,. \tag{89}$$

Other functionalities available in the camera's driver that are also used are brightness, contrast and gamma correction adjustments, whose values were empirically obtained. The parameters of these features are adjusted in such a way that markers' pictures in the acquired images, namely the bright white circular or elliptical regions, are enhanced in comparison to other, not relevant, image segments. In this manner, the task of image pre-processing algorithms is facilitated. A bad pixel correction function is also used, since the CMOS array sensor has usually a few bad pixels, which do not work properly. With this feature turned on, every pixel which has always the same intensity value under any illumination conditions is considered as having null intensity values, since it is a defective pixel.

4.1.3 Image Pre-Processing

After being transferred from camera to PC main memory, images are processed in a per frame basis by the pre-processing software module. All processing tasks are executed on the current image buffer selected. A typical (zoomed) grabbed image of the label containing two active markers, in Figure 64 (left), is presented in Figure 66.

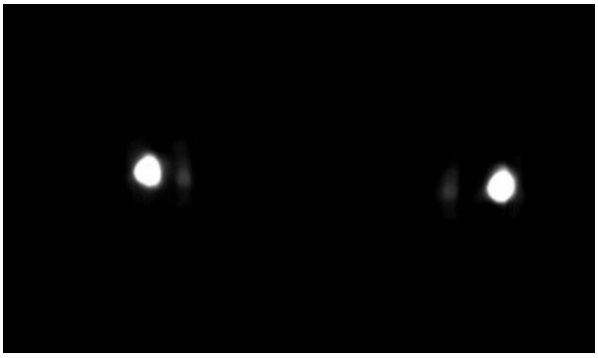

Figure 66 – Typical (zoomed) grabbed Image of 2 Retro-reflective Active Markers

The goal of image pre-processing task is to extract the 2D position of each marker on the plane of the camera's sensor. Initially a Global Thresholding algorithm is applied (section 3.2.2) to ensure more robustness against lighting variation, followed by a Blob Coloring algorithm (section 3.2.3), which results in a list of detected white regions in the image. Then, both size and geometrical shape constraints (section 3.2.8) are considered in order to select only proper regions, i.e., with reasonable size – size range criterion using minimal dimensions of 2 x 2 pixels - and shape – only regions with round or elliptical form are selected, using the black-white ratio test. These constraints remove also bad pixels and regions generated by optical noise.

Subsequently, the center of each region is calculated with subpixel precision using the Binary Centroid method (section 3.2.9). Afterwards, lens radial distortions are corrected (tangential distortion is negligible, hence not considered), resulting in the actual center of each region. Next, the intrinsic parameters of the camera are applied to resulting coordinates (section 3.1.3) by multiplication of these coordinates by the camera calibration matrix. The results are the 2D real coordinates of markers' centers on

the camera sensor's plane. Figure 67 shows the sequence of image pre-processing steps applied on a typical six-marker PTrack label.

During the development of this work some additional attempts were made to improve image pre-processing performance. Among those, the Top-hat operator (section 3.2.10) was implemented, evaluated and discarded due to the excessive processing load required. The results obtained with the Top-hat operator were equivalent to the final results obtained with the Global Thresholding algorithm. Also the Filling operator (section 3.2.7) was implemented but discarded, because experiments showed that the operator had no influence on the calculated marker centers and yet consumed some processing power.

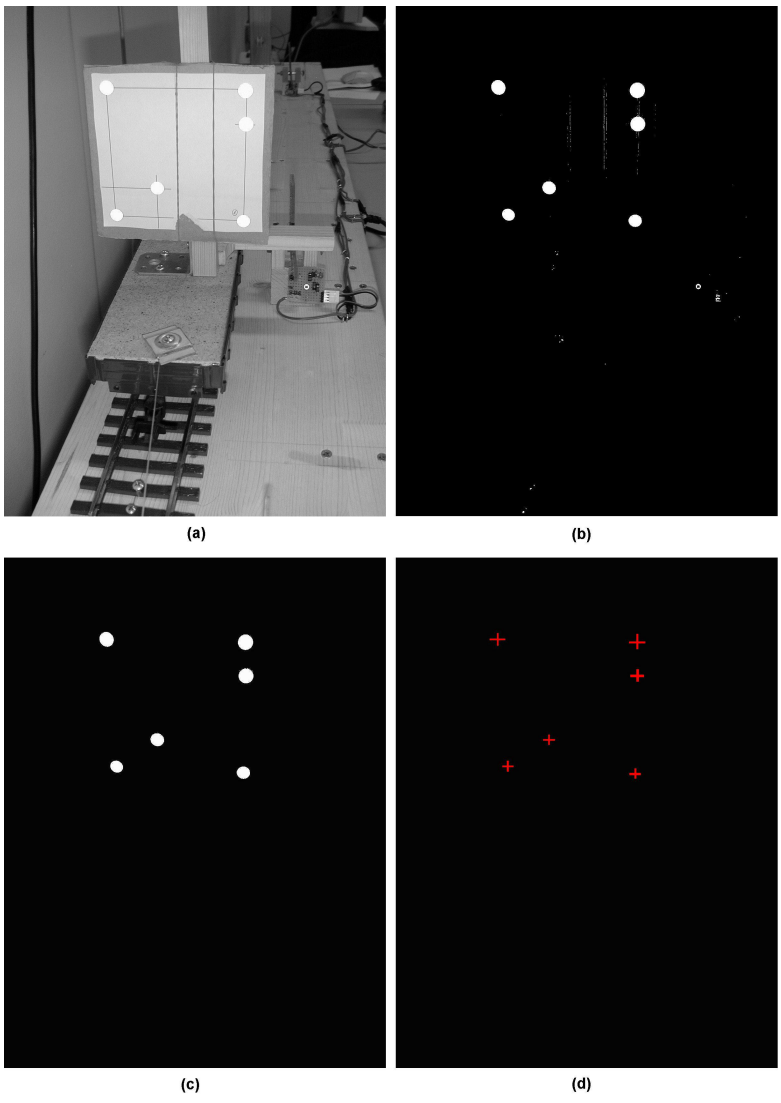

Figure 67 – Image Pre-processing Tasks applied on a Typical PTrack Label: (a) Original Grayscale Image; (b) after Global Thresholding; (c) after Size and Shape Constraints; (d) Calculated Marker Centers

Generally speaking, the brighter the markers are, the easier it is to identify them and extract their centers in the images. This inference supports the idea of having a

large number of infrared LEDs to illuminate the scene, in order to make image pre-processing tasks easier.

All routines were implemented using the Microsoft Visual C++ 6.0 Integrated Development Environment.

When compared to ART cameras of the ARTtrack system – the original hardware with which PTrack was used - the results of the pre-processing software module implemented in this work performed slightly worse, especially regarding marker position reliability and robustness against lighting conditions. The causes for this inferior performance lie probably on a better hardware construction by ART, as well as on the higher number of infrared LEDs in ART cameras. Also the decision of which image pre-processing steps to utilize affects directly the performance. In this case, the pre-processing tasks of ART cameras have probably a better choice of algorithms as the system implemented in this work. Also the higher illumination level contributes to better results of pre-processing steps.

4.1.4 PTrack

The PTrack algorithm (section 3.4.1.2) had to be adapted to work with the proposed hardware module. In the PTrackScan module, basically the maximum number of iterations, as well as the highest allowed error in incoming marker center coordinates, have been changed in order to cope with the smaller precision hardware module, when compared to ART cameras. Both image pre-processing and PTrack algorithms run as one single thread, since they use the same processing hardware – the PC. In original PTrack, image pre-processing was run in the cameras and the PTrack tracking algorithm on the PC.

Optimizations have also been implemented. Differential Tracking features were added to the algorithm. Originally, the position of markers was again calculated in every frame, not considering the tracking information already obtained in the last frame. Currently, in each new frame the position and orientation of a label in the last frame are used as a basis for the calculation of new position and orientation information. Considerable improvement in performance was obtained with this technique. The system with differential tracking was in average 70% faster than the regular implementation. The slower the label moves between frames, the higher is the gain in processing speed.

After increasing the maximum allowed error in marker positions provided by the pre-processing module, the radar sweep routine (Figure 47) could be adjusted. With higher allowed errors also the step of the radar sweep could be increased from 1 to 5 deg without noticeable loss of efficiency. Thus, the total number of radar sweep tests dropped from 180 to 36, increasing processing speed.

The PTrackUDP and the OpenTracker interface modules were also modified in order to support large area tracking features. An information quality measure was added to each detected label to serve as sensor fusion criterion. The quality criterion used is related to the angle between the label position vector and the Z axis, i.e., the greater the distance from the label to the optical axis (Z axis), the worse is the quality. For example, if the label is positioned on the Z axis, with coordinates (10, 20, 0), then

its quality is 100%. If the label is positioned with the maximum allowed angle from the Z axis (on the limits of FoV), in any given side of the axis, then its quality is near to 0%. Also the distance from the camera could be considered as a quality criterion, but tests showed that even with the label at the farthest detectable position from the camera (3.5 m) tracking data had the same quality (reliability and robustness) as at the nearest position (around 60 cm).

Also, the identification of which camera detects the label is sent with the tracking information, as well as the identification of which camera serves as reference point for the transmitted tracking data. These modifications do not inhibit stand-alone functionality of the one-camera tracking module, i.e., only a single module can also be used by configuring a wide-area tracking system with just one camera. Figure 68 shows the structure of UDP packets sent as output of the PTrackUDP and OpenTracker modules. Instead of position and orientation of the label, 3D position coordinates of markers belonging to the label are sent in order to facilitate merging of data, avoiding inconsistent results after sensor fusion. Position data is provided in millimeters.

Structure of UDP Packets as output of PTrack

Packet number	Camera ID	Detected label ID	Reference Camera ID	Information quality	Position Marker 1 (X, Y, Z)	Position Marker 2 (X, Y, Z)	Position Marker 3 (X, Y, Z)	Position Marker 4 (X, Y, Z)	Position Marker 5 (X, Y, Z)	Position Marker 6 (X, Y, Z)

Figure 68 – UDP Packets sent as Output of One-camera Tracking Modules

PTrack, when running on the hardware module implemented in this work, has a maximum theoretical update rate of 55 Hz, limited by the maximum frame rate of the camera. In test conditions (see section 5.5), the system reached a maximum of 29 Hz, what could be easily enhanced, for example, by using a computer with higher processing performance.

4.2 INCREASING THE WORKING VOLUME

To allow large or wide area tracking features when using two or more cameras, modifications were introduced in the existing OpenTracker module of PTrack, as shown in last section, creating a proprietary protocol in the UDP interface. The topology of the wide-area tracking system (Figure 69) consists of independent one-camera tracking modules, which transmit UDP packets containing tracking information to a central module, where the information is merged using sensor fusion techniques. If the one-camera tracking modules are located in different computers, UDP packets are sent over ethernet interfaces. The central module then transmits the final tracking information also using UDP packets. This information is directed to OpenTracker or, in case this is not available, to a visualization application.

The structure of the UDP packets (Figure 70) sent from the central module to OpenTracker or to the visualization application is specified by OpenTracker's input interface, which is the same used by the ART system. Among other requirements, the rotation matrix of the object (artifact) must be sent within the packet, although this is redundant since rotation angles are also sent. This structure was preserved in the scope

of this work because any change would require modifications in OpenTracker's source code, probably resulting in undesired incompatibility issues. By definition, OpenTracker is a generic framework for integration of different tracking systems into AR/VR environments. Thus, the tracking systems should adapt themselves to OpenTracker and not the other way around.

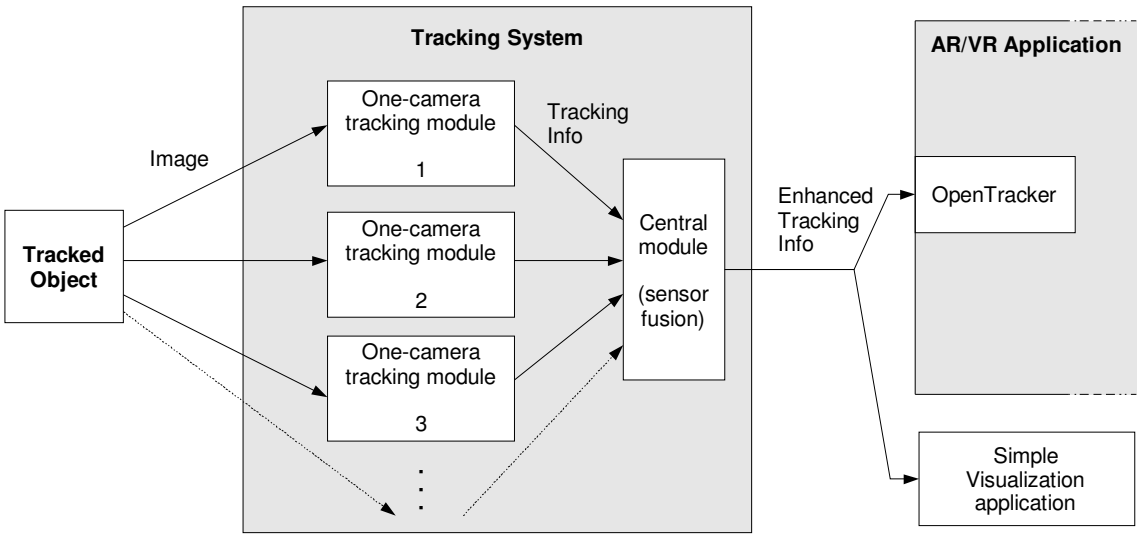

Figure 69 – Topology of the Wide-area Tracking System

Structure of UDP Packets as output of the Central Module

Packet number	Detected label ID	Position X	Position Y	Position Z	Rotation X (roll)	Rotation Y (yaw)	Rotation Z (pitch)	Rotation matrix (3 x 3)

Figure 70 – UDP Packets sent as Output of the Central Module

In the central module a simple Weighted Average sensor fusion algorithm (section 3.7.1) is executed with frequency directly related to the average frame rate of the system. First, tracking information received is sorted by the label identification field so that information related to the same label is grouped. Based on the quality criterion sent, the final tracking information about a given label is calculated by first obtaining the averaged marker positions using the expression

$$\begin{aligned} \bar{x} &= \frac{1}{\sum_{i=1}^{n} Quality_i} \cdot \sum_{i=1}^{n} Quality_i \cdot x_i \\ \bar{y} &= \frac{1}{\sum_{i=1}^{n} Quality_i} \cdot \sum_{i=1}^{n} Quality_i \cdot y_i \\ \bar{z} &= \frac{1}{\sum_{i=1}^{n} Quality_i} \cdot \sum_{i=1}^{n} Quality_i \cdot z_i \end{aligned}, \tag{90}$$

where n is the total number of occurrences of the same label in the scene. Next, the position of the label is extracted, being the coordinates of the first marker (p1 in Figure 24), and also the orientation is computed, which is calculated as the normal vector of the label. This can be easily obtained from the cross-product of two perpendicular edges of the label.

Kalman Filter performs probably better than simple Weighted Average and is planned as a future enhancement of this work. Main contributions of Kalman Filter are higher smoothness and robustness, due to trajectory estimation of marker positions.

The visualization application simply plots the 3D tracked objects on the screen as well as shows tracking information in written form.

4.3 CALIBRATION

Initially the DLT method (section 3.3.1) was implemented in this work, using the SVD technique to obtain the 11 DLT parameters. DLT was later discarded due to difficulties when trying to include lens distortion parameters estimation, since the solution of systems with nonlinear equations is non-trivial, and convergence was not always guaranteed. Additionally, DLT requires a non-planar set of markers in the calibration patterns, which become more complex to be built.

Zhang's method, already included in Intel OpenCV[3] library, was then preferred, since it provides calibration of lens distortion parameters without difficulties and allows complete calibration using only a planar set of markers. For the implementation the same image pre-processing steps from tracking algorithm were used, but only up to the calculation of markers' centers in pixel coordinates. OpenCV's implementation of the method includes routines to manage multiple views, enabling higher precision during calibration due to the use of redundant information.

The calibration procedure consists of two steps: first, each camera has its intrinsic parameters individually calibrated using a specific calibration pattern; then, the camera's extrinsic parameters can be calibrated in relation to another camera by placing a different calibration pattern in a region visible to both cameras. For each pair of adjacent cameras belonging to the wide-area tracking system a calibration of extrinsic parameters must be made.

4.3.1 Hardware: Calibration Patterns

For calibration of intrinsic parameters of each camera a grid with 16 retro-reflective markers in the intersections of lines was used, as shown in Figure 71.

For the calibration procedure a correspondence algorithm is needed, which relates the 2D positions of the detected markers with the previously known locations of the markers in the calibration pattern. This problem is similar to the one explained in section 3.5.2. For the proposed pattern, the correspondence algorithm is simplified if the user holds the pattern in a fixed orientation, without turning it. Translational movements

[3] OpenCV from Intel Corporation. Open-source Computer Vision Library. Available at <http://www.intel.com/ technology/computing/opencv>. Last accessed on December 16th, 2005.

are allowed. In this case, the algorithm must only sort the detected markers from top to bottom and from left to right, and the correspondence is established.

Calibration of extrinsic parameters of one camera in relation to another one is obtained by use of another calibration pattern, in shape of the 'L' letter, as shown in Figure 72. The pattern contains 8 coplanar markers.

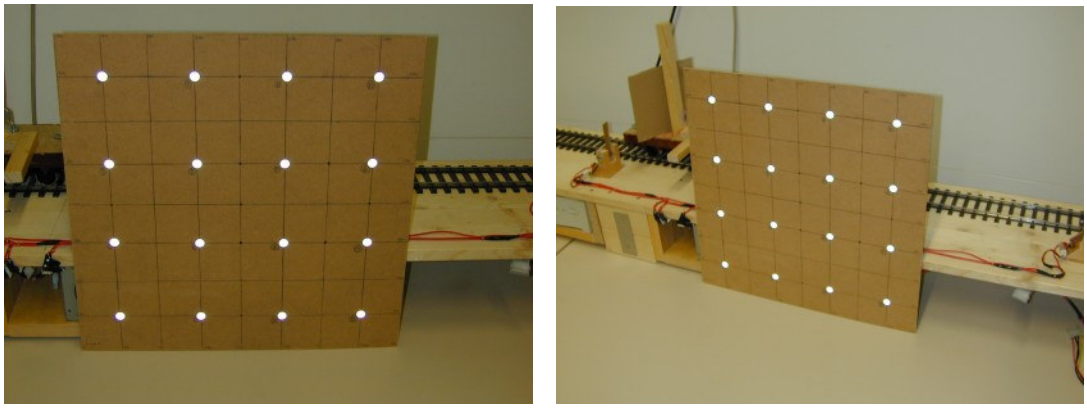

Figure 71 – Pattern for Calibration of Intrinsic Parameters using OpenCV

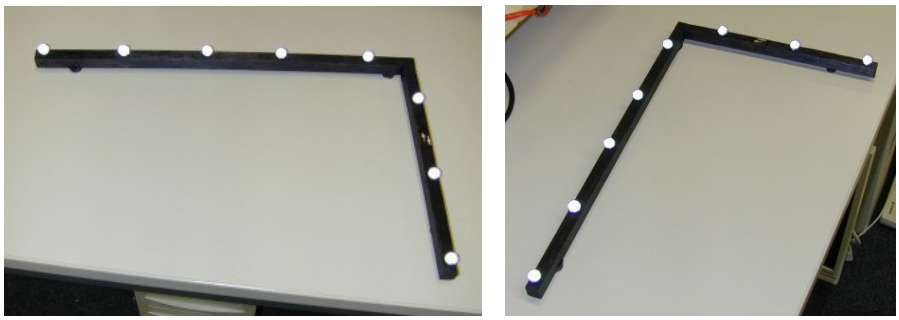

Figure 72 – Pattern for Calibration of Extrinsic Parameters using OpenCV, with 8 Coplanar Markers

The correspondence algorithm in this case seeks groups of markers forming two lines, with 3 and 5 markers. Once the lines are found, markers are sorted according to the distance from the point of intersection of the lines.

For the record, an attempt was made during the development of this work to use calibration patterns with active markers. This was done before making the decision of using passive markers and when still using the DLT method, so intrinsic and extrinsic parameters were calibrated together, using the same pattern. Figure 73 shows the calibration pattern built. For this pattern a correspondence algorithm similar to the one used for the grid pattern was implemented.

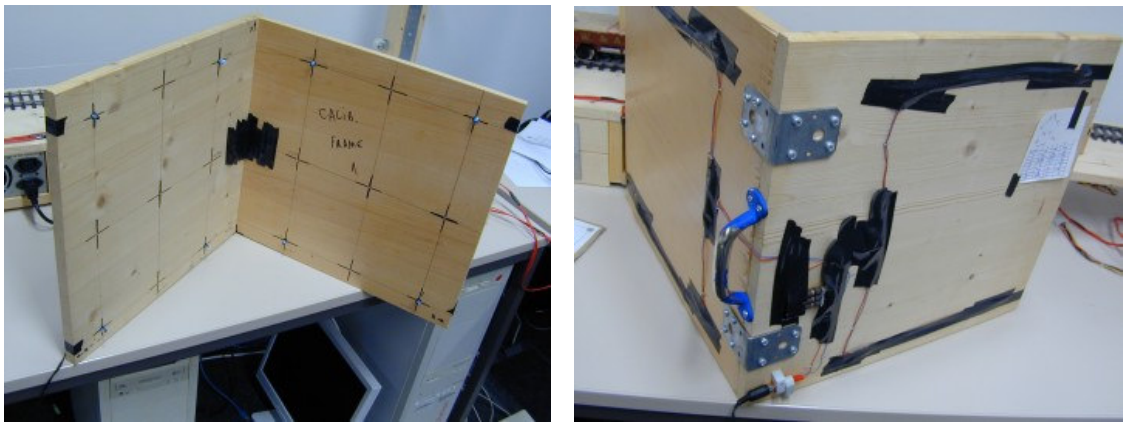

Figure 73 – Calibration Pattern with Active Markers: Front View with LEDs turned on (left) and Back View (right)

4.3.2 Calibration Procedure

Before initiating the calibration procedure the correct adjustment of the focal length of the cameras must be done by turning the lens barrel until the image is sharp, while watching in real-time the video frames taken by the camera with help of a visualization tool provided by the camera's manufacturer. Since infrared spectral region is used, the adjustment of focal length must be done with infrared image, because there is a so called "focus shift" or difference between the visible focus and the infrared focus. Strictly speaking, focal length is specific to a given wavelength (see (NIEUWENHUIS, 1991) for details). Figure 74 shows the variation in focal length for three different types of lenses, including lenses with infrared enhancing filter, the ones used in this work.

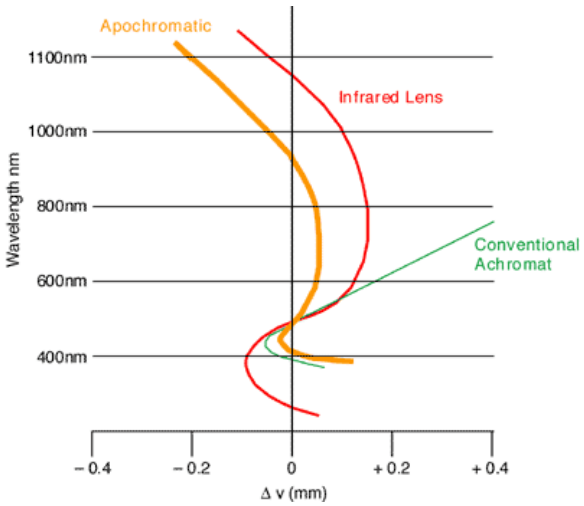

Figure 74 – Variation in Focal Length for Different Wavelengths with 3 Types of Lenses[4]

[4] The Medical and Scientific Photography website. Available at <http://msp.rmit.edu.au/index.html>. Last accessed on December 16th, 2005.

Due to the focus shift the focal length adjustment must not be done without the infrared enhancing filter.

Afterwards, calibration of intrinsic and extrinsic parameters is accomplished by using a calibration software developed specifically for this purpose. For intrinsic parameters calibration the grid pattern must be held in front of the camera, with the plane of the pattern as perpendicular as possible to the optical axis, approximately 1 m away from the lens. During the calibration procedure, which takes about 1 minute, the pattern can be moved as long as it remains perpendicular to the camera's optical axis and is not turned around it. The procedure must be repeated individually for each camera.

The wide-area tracking system must necessarily have a main camera, which is considered the origin of the global coordinate system. All other cameras have their extrinsic parameters related to the main camera, directly or through intermediary cameras. Figure 75 shows the topology of a 3-camera system, where the leftmost camera is the main one, the central one is directly related to the main one and the rightmost camera is indirectly related to the main camera through the central one.

In order to calibrate the extrinsic parameters, the user must specify to which camera they are related and then place the extrinsic parameter calibration pattern in the common region between two related cameras. The pattern can be moved during the procedure, which lasts about 1 minute. The procedure must be repeated until all cameras have their extrinsic parameters calibrated.

In both intrinsic and extrinsic parameters calibration, several samples of images are grabbed and parameters are calculated for each sample. In the end, parameters are averaged to yield final results.

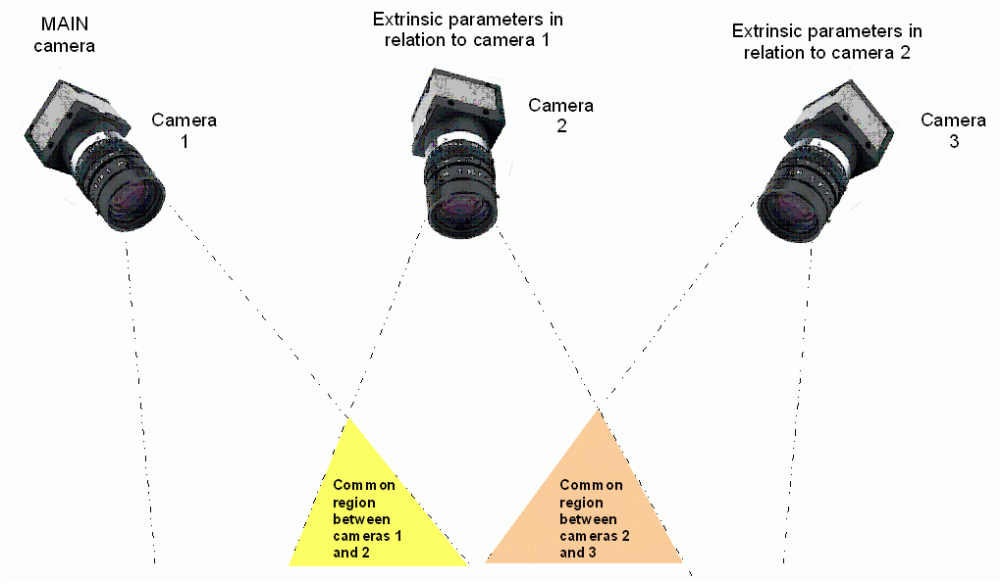

Figure 75 – Example of Multiple-camera Topology with 3 Cameras

Currently, there is a limitation in the extrinsic parameters calibration procedure: both cameras must be connected to the same computer, in order to be

calibrated. After calibration the one-camera tracking modules can be separated and operated in different computers.

4.3.3 Results

Both intrinsic and extrinsic parameters are saved as an ASCII file in the PC's hard disk where the calibration application runs. An example of the content of such files is shown in Figure 76. See sections 3.1.3 and 3.1.4 for parameters description.

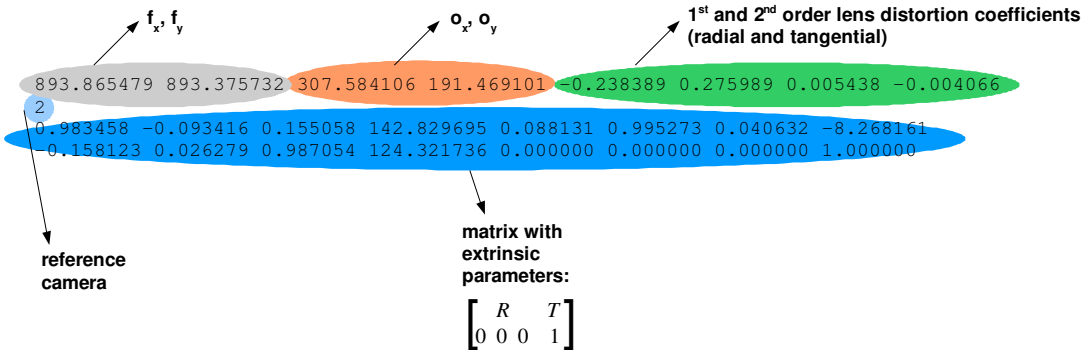

Figure 76 – Text File with Camera Calibration Parameters in Wide-area Tracking System

5 SYSTEM EVALUATION AND TESTS

This chapter describes the evaluation testbed built to measure the system's exactness, as described in section 2.1, in both translation and orientation. Testing procedures are explained. Results of tests for the tracking system developed in this work as well as for other systems used as comparison are presented and discussed.

5.1 OBJECTIVE AND REQUIREMENTS

Evaluation tests were developed in order to compare the performance of the tracking system implemented in this work with other tracking systems. Attributes measured in each system are mainly accuracy and precision, which together compose exactness, as defined in section 2.1.

The evaluation testbed should provide a means to accurately compare pre-defined physical target paths (real condition) to pose estimation results (measured condition) computed by the tracking system. A physical target path is the continuous change of position and orientation by a tracked target fiducial which physically moves from a starting to an end point. The evaluation testbed should allow pre-definition of target paths to define the nominal condition for an experiment. It should be possible to compare position and orientation of the tracked target to the results of pose estimation at any given time instant.

Each experiment should be repeatable, i.e., every time the experiment is carried out, pose estimation by the tracking system should return approximately the same results for the same pre-defined target path.

In order to evaluate orientation and translation independently, the evaluation testbed should consist of two experiments: one for the translational exactness measurement and one for the rotational exactness measurement. To avoid interferences, during translation experiment the orientation of the target should be kept fixed, and during orientation experiment the position of the target should not be changed.

Although orientation and translations errors are combined during practical use, an independent evaluation of errors after development is preferable from the technical point of view. This way, problems specifically related to orientation or translation parts of the algorithm can be independently identified.

The translation experiment should cover the whole working range of the hardware (from 60 cm up to 3.5 m, if possible). The rotation experiment should allow measurements of rotational exactness for a 360 deg turn of the target in front of the camera, using different angles of attack, so that most of the possible orientations of the target are covered. Both experiments should be built as accurately as possible, inducing the least amount of error in the measured results.

The evaluation testbed should allow testing of just one single one-camera tracking module as well as a setup with several one-camera tracking modules in a wide-area tracking system.

5.2 EXPERIMENTAL SETTINGS

Ideally, in both rotation and translation experiments a moving target should be tracked, since this condition is closer to real utilization than tracking of static targets. This was accomplished in the implemented translation experiment. In rotation experiment hardware limitations prevented use of ideal conditions, so the target is tracked only when fixed in one of the 48 positions of a complete turn. In each position, several samples of tracking information are taken. When done, the target is automatically moved to the next position.

5.2.1 Translation Experiment

In the translation experiment a label is placed on a car and carried in uniform motion along a track between two photocells. The distance between photocells is exactly 0.894 m. The camera or cameras are positioned 0.775 m from the first photocell. A data analyzer compares actual distance against nominal distance over time to calculate the statistical results.

Applying the concepts defined in section 2.1, this experiment measures the following properties of the tracking system:

- Accuracy: is the mean error in the measurement of position of an object, calculated as the difference between measured values (real condition) and the nominal values pre-defined by the target path in millimeter (mm).

- Precision: is the standard deviation of the error in the measurement of position of an object during a series of experiments in millimeters (mm).

5.2.2 Rotation Experiment

The rotational experiment is based on (MALBEZIN, 2002), where an ARToolKit fiducial target is placed on the floor and a camera attached to a tripod is moved around it, at different heights, so that different angles of attack are obtained.

In this work the rotation experiment consists of placing a label on a rotor with adjustable angle of attack (reminder: this is the angle at which the target or label is tilted from the position in which it is facing the camera) mounted on a stepper-motor. Each test-run covered 47 rotation steps at 7.5 deg each totalizing 352.5 deg. When moving from one rotation step to the next, the label showed some jittery during settling time, until it was completely motionless. In order to disregard tracking data calculated during this settling time, only data samples of the corresponding inner 60% of a 4 s time-slot of each step were used for computation of the results. For each step several samples were made and the average result was computed, then the average of all averages per step was computed for a test-run. This experiment measures the exactness of the normal vector by measuring the individual exactness values of this vector's heading, attitude and bank. The camera was positioned at 0.80 m from the rotor, facing it as aligned as possible.

Applying the concepts defined in section 2.1, the exactness measured by this experiment defines following properties of the tracking system:

- Accuracy: is the mean error in the measurement of orientation of an object, calculated as the difference between measured angle (real condition) and the nominal angle pre-defined by the target path in degrees (deg). The rotational accuracy is divided across three angles (Heading, Attitude, Bank)

- Precision: is the standard deviation of the error in the measurement of orientation of an object during a series of experiments in degrees (deg).

5.2.3 Test of Frame Rate *versus* Number of Labels

The overall frame rate of the tracking system depends on the computer processing load it requires. Thus, a necessary test for the system is to measure the performance degradation with increasing number of labels to be tracked.

This test shows the sensitivity of system's performance to the number of targets to be tracked. The less sensitive the system, the better is its overall performance.

5.3 TESTBED

In order to carry out translation and rotation experiments, a complete testbed had to be built (Figure 77), which has reasonable construction accuracy and allows partly or fully automated test procedures.

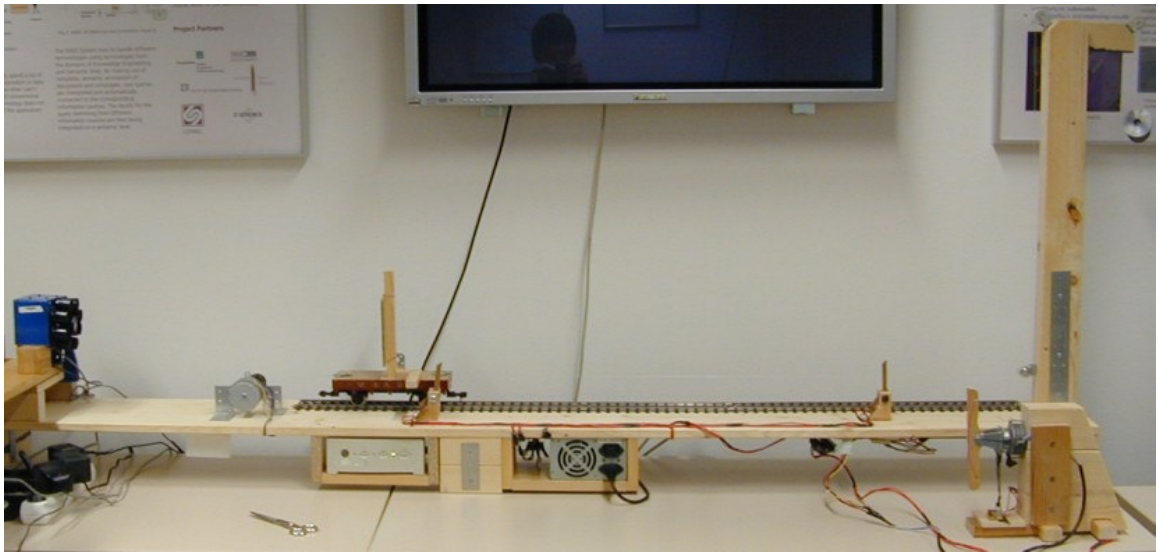

Figure 77 – Complete Testbed with both Rotation and Translation Experiments

Figure 78 contains a general outline of the testbed electrical and mechanical connections. The Appendix A presents detailed electrical schematics of the implementation.

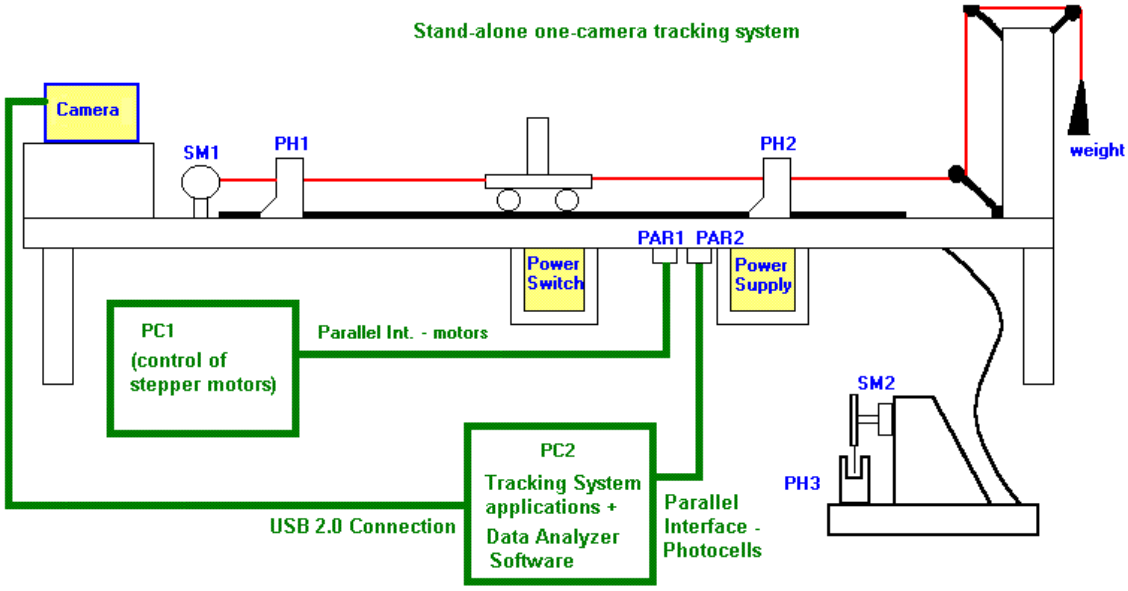

Figure 78 – Outline of Evaluation Testbed's Eletric Signal and Mechanical Connections (in Stand-alone Operation)

5.3.1 Translation Experiment

A professional testbed used by the company ART consists of a metal thread powered by a DC motor with constant speed. Considering the thread pitch and the rotation frequency of the linear motor, the tracked target can be put in motion very precisely, describing a linear uniform movement from a starting point to an end point. However, due to budget limitations, this solution could not be built to test the tracking system implemented in this work.

Instead, a low cost solution similar to the one used by ART was built (Figure 77 and Figure 79), consisting of 1,4 m of used metal model railway track with 45 mm gauge and a used metal-made low side car on which the target is placed. To apply uniform movement a stepper motor was used at frequencies above 30 Hz which would let it perform like a regular linear DC motor. Loss of steps is very unlikely, although possible. A mechanism to detect and correct loss of steps by the stepper motors could be built, or even a setup using a DC motor and encoders to have more reliable position information, but both options were disregarded due to budget limitations. A stepper motor axis was modified on one end to be able to release and pull a nylon string fixed on top of the low side car. The string is kept stretched over the track and is attached to a counterweight which hangs at the end of the track, on the opposite side of the camera or cameras. To record start and stop time instants of the experiment, two photocells are placed at the beginning and at the end of the track to detect when the low side car passes by. The photocells are connected to the parallel port of a computer running the data analyzer software described in section 5.3.3. The stepper motors are connected through a driver circuit to the parallel port of another computer and are controlled by the software described also in section 5.3.3.

All necessary steps for the translation experiment are now explained. First the low side car is positioned before the first photocell PH1 (see Figure 78) and the label to be tracked is fixed to the mast of the car facing the camera (or cameras), so that the label is approximately aligned with the center of the main camera's image plane.

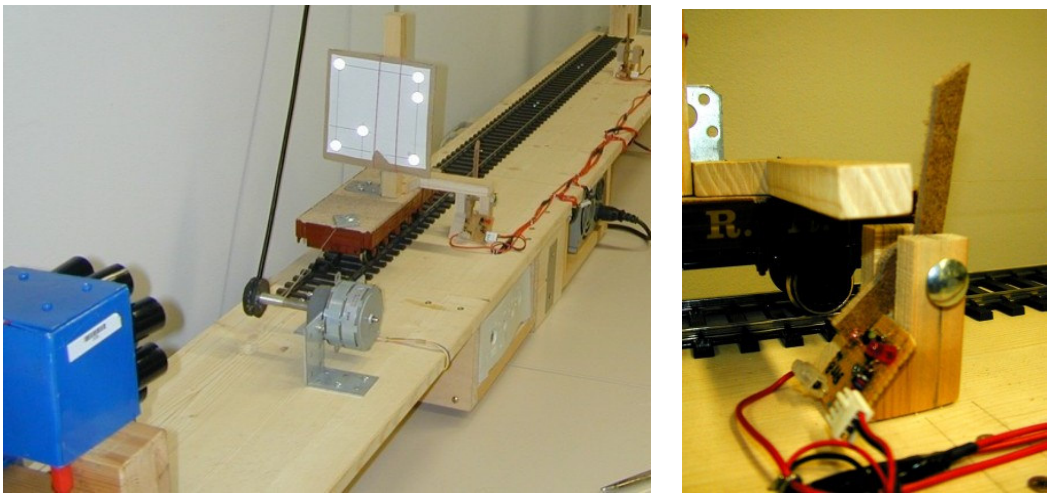

Figure 79 – Evaluation Testbed: Detailed View of Camera, Track, Low Side Car (left) and Photocell (right)

The tracking system (PTrack – one or several one-camera tracking modules - or ARToolKit) is then started on PC2 and broadcasts tracking data to the data analyzer software also on PC2, which receives it via the OpenTracker module. The stepper motor control software on PC1 is now programmed to let the first stepper motor SM1 cover the distance between photocells PH1 and PH2 and come back afterwards to the initial position. When the low side car passes the first photocell PH1, the data analyzer software is triggered and starts logging the tracking information, giving each sample a timestamp. When the second photocell PH2 is passed, logging stops and statistical results are computed, while the low side car returns to the initial position. In order to perform like a linear motor, the stepper motor is used with frequency higher than 25 Hz.

The parameters for the software which controls the stepper motors were set to move the wagon 1,500 steps at a speed of 30 Hz passing photocells PH1 and PH2. With these parameters the wagon needed on average 46.1 s to cover 0.894 m, resulting in an average speed of 0.01935 m/s. This speed was adopted because it was high enough to ensure uniform movement and reduce the influence of friction and, on the other hand, it was low enough to ensure reasonable number of data samples between PH1 and PH2.

The fully automated statistical analysis yields:

- The update rate of the tracking system - how many samples per second have been recorded - in Hz;
- The nominal distance between the photocells in m - in this case always 0.894 m;
- The actual distance measured between the photocells in m;
- The positional accuracy is computed in m. Given the overall time it took to cover the also known nominal distance between both photocells and

considering uniform movement, a function can be established which returns the nominal position of the label at any given time instant. Therefore, for each sample the difference between measured and nominal position at that time instant is computed. By averaging the obtained differences for all samples, the positional accuracy for the test-run is calculated;

- Furthermore, based on this data, the standard deviation of the differences is computed in m, resulting in the positional precision for the test-run.

The translation experiment log file produced by the data analyzer also contains the following information about each collected data sample:

- Timestamp in s, provided by the computer where the data analyzer software runs;
- Position in camera space coordinates in m;
- Normalized normal vector in camera space coordinates;
- Current position (distance from starting point) in m.

5.3.2 Rotation Experiment

Figure 80 depicts the evaluation testbed built for rotation experiment. A stepper motor is used with a rotor attached to it, whose angle of attack can be adjusted. The testbed is placed facing the main camera. If the camera is centered aligned to the axes of the motor, 360 deg rotation movements of a tracked label can be performed at different angles of attack, without moving or re-adjusting the camera. After each step of 7.5 deg the motor will halt to enable stable data acquisition over a certain period in the time interval, excluding possible jittering of the rotor, which can occur when moving from one position to the next. A photocell is placed on the experiment in order to record when a complete 360 deg turn begins and when it is finished, triggering the data analyzer software.

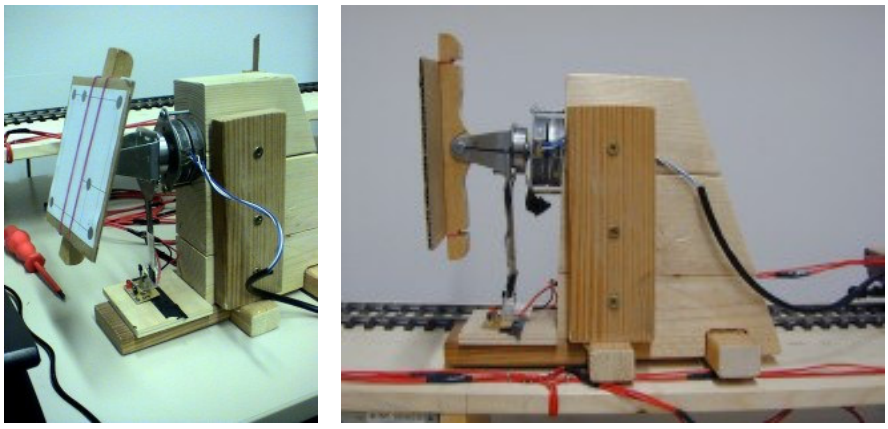

Figure 80 – Rotation Experiment Evaluation Testbed

Necessary steps for the rotation experiment are: first, the rotor is positioned in such a way that the photocell PH3 (see Figure 78) is blocked by the stem attached to the stepper motor; then, the label is attached to the rotor so that the axis of the stepper

motor SM2 is aligned with the optical axis. This is accomplished by using a laser level device. Next, the rotor is set to the desired angle of attack, using a protractor (angle measurement tool), to simulate a camera performing a complete turn around a label target at a certain inclination towards it. When this is done, the tracking system is switched on (PC2) and starts transmitting tracking information to the data analyzer software (also in PC2).

The stepper motor control application on PC1 is set to move the rotor exactly 48 steps to perform a 360 deg rotation which starts and stops with the photocell PH3 blocked. Each step has 7.5 deg. The stepper motor is programmed to use a frequency of 0.25 Hz and to remain at each position for 4 s. Similarly to the translational experiment, a function can be established to adequately describe the nominal normalized normal vector on top of the target label on the rotor at each given time instant of the rotation. Each time the stepper motor advances to a new position the rotor continues to oscillate for a short time until it stabilizes. The tracking data must be sampled when the rotor has stabilized at a certain position. For that reason, data collected in the initial 20% and in the final 20% ranges of the time interval spent in each position are disregarded for calculations, which consider only data sampled during the remaining 60% of the time interval. This workaround is only needed to compensate for imprecision sources introduced by the testbed, specifically the oscillation when switching between rotation angles.

The fully automated statistical analysis yields information similar to the translation experiment:

- The update rate of the tracking system - how many samples per second have been recorded in Hz;
- The positional accuracy of the normal vector estimate is given split up in three angles, namely Heading, Bank and Attitude. Only 47 rotor positions are taken into account for calculation, because the initial and the last positions trigger start and stop events of the data analyzer software and therefore measurement is skipped in this location, thus only 352.5 deg are covered. The accuracy for the three angles, in degrees, is calculated as the average difference between nominal and measured values for Heading, Bank and Attitude, using the values of all positions.
- Furthermore, based on the data, the standard deviation of the error of each of the three angles is computed, in deg;
- The measured angle of attack of the rotor in deg is computed from the average angles of each measured normal vector taken into consideration above to the optical axis of the camera.

The rotation experiment log file produced by the data analyzer also contains the following information about each collected data sample:

- Timestamp in s, provided by the computer where the data analyzer software runs;
- Position in camera space coordinates in m;
- Normalized Normal-vector in camera space coordinates.

5.3.3 Stepper Motor Control and Data Analysis Software

Besides the tracking system algorithms, two additional software applications are used to perform the experiments. One application sets and controls the stepper motors, so the physical target path can be previously defined and then executed. The other application logs and analyzes data from both experiments through OpenTracker and monitors the states of the photocells.

The software to set and control the physical path of the tracked target is part of the stepper motors' parallel port driving circuits and works for both motors (Figure 81). The parameters of interest for each motor are: number of steps (7.5 deg per step), frequency (rotation speed in Hz) and direction (clockwise or counterclockwise). The software allows for a number of sequential commands to be issued beforehand. At speeds above 25 Hz the stepper motor behaves like a linear DC motor.

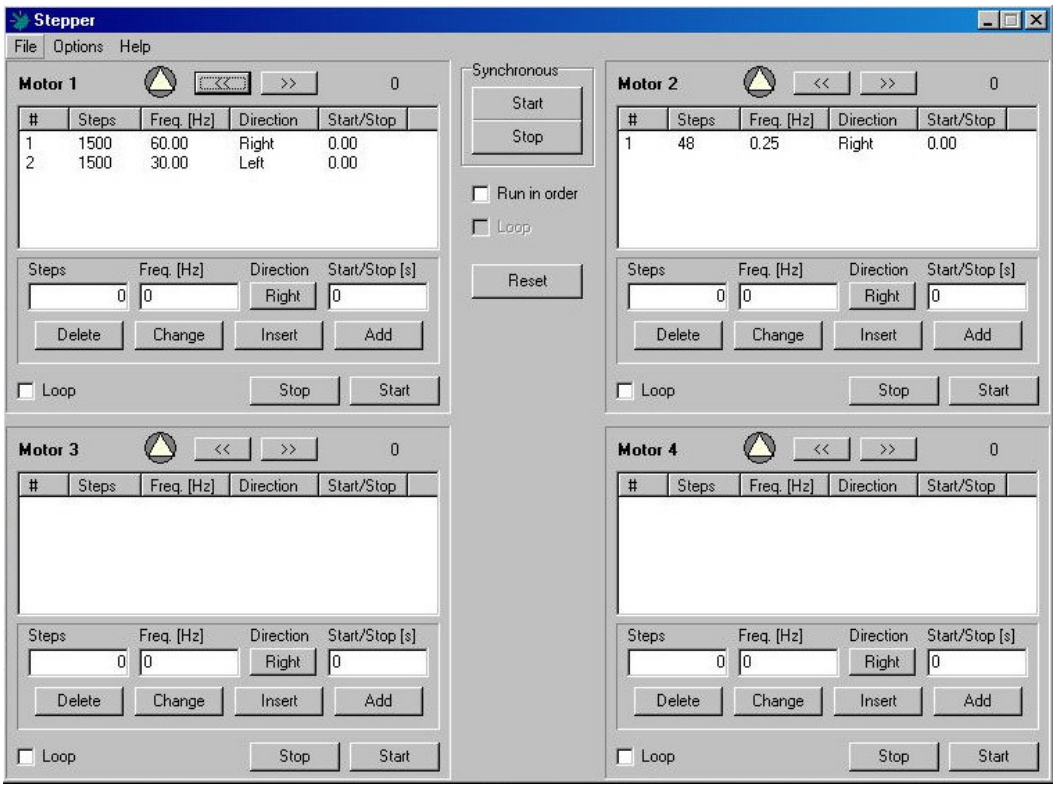

Figure 81 – Control Software for 4 Stepper Motors

The data analyzer software (Figure 82) is a command line application which receives its input from two different sources. On one hand it receives tracking data from a connected tracking system via the built-in Opentracker library. On the other hand it receives input from the photocells attached to the parallel port to start or stop logging. The application itself is a console application with command line options to select which experiment to conduct.

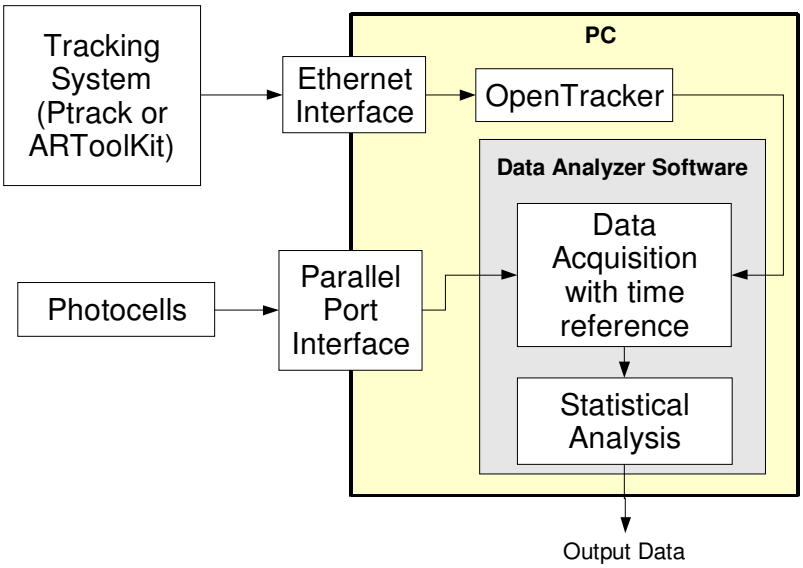

Figure 82 – Block Diagram of the Data Analyzer Software

5.4 SYSTEMS UNDER TEST

The testbed was used to evaluate the tracking system implemented in this work. Thus, the system was tested in its stand-alone version (only a single one-camera tracking module is tested) and in the wide-area tracking configuration (two or more one-camera tracking modules – in this case, only two modules were used).

In order to compare the system performance, also the ARToolKit tracking algorithm running on the same IDS uEye UI1210-C camera was tested. In (SANTOS, 2005) the PTrack algorithm, when running in even faster cameras, was tested against ARToolKit running on standard webcams. To avoid this unfair comparison, ARToolKit was adapted to run on the same cameras as PTrack.

Thus, the tests performed were:

- Scenario 1: PTrack on IDS Cameras, in stand-alone configuration;
- Scenario 2: ARToolKit on IDS Cameras (exists only in stand-alone version);
- Scenario 3: PTrack on IDS Cameras, in multiple-camera configuration.

5.5 RESULTS

Each system described in section 5.4 underwent the tests previously described in section 5.2 and the results are presented in the following sections.

5.5.1 Test Conditions

The experiments were conducted under following conditions:

- Tests were performed in a 5.8 m wide, 8.75 m long and 2.6 m high room, illuminated by 4 rows of 5 fluorescent lamps each, having each lamp 38 W power. Lamps were powered in interlaced configuration, so that 2 non-adjacent rows can be turned off independently from the others. Three illumination conditions were used: full illumination (when all lamps are turned on), half illumination (when only half of the lamps are turned on) and no illumination (when all lamps are turned off – only possible when using infrared spectral region). Lux is the unit of measurement of luminous flux density at a surface, defined in the SI (International System of Units). Using a Lux-meter, averages of 410 Lux and 205 Lux were measured along the testbed in full illumination and half illumination conditions, respectively.

- For the translation experiment, each run consists of 1,500 steps of the driving stepper motor at 30 Hz, allowing approximately 2 cm for acceleration and 2 cm for deacceleration. The main camera was positioned at 0.775 m from the first photocell.

- For the rotation experiment, each turn consists of 48 steps of the driving stepper motor at 0.25 Hz, turning in clockwise direction, seen from camera. Rotor and label were positioned at 0.80 m from the main camera, slightly farther than the point of minimum working distance, but in an optimal range for reach of flash strobes.

- For the experiment Frame Rate *versus* Number of Labels, the labels were held still at approximately 1 m from the main camera, facing it.

- Two computers were used: one running the software for stepper motor control and the other running the tracking system and the data analyzer software (see section 5.3.3).

All systems were tested under the same conditions, unless noted in the experiment's description.

5.5.2 PTrack on IDS Cameras in Stand-alone Configuration (Scenario 1)

When in stand-alone configuration, i.e., only a single one-camera module, the topology of the testbed is exactly as shown in Figure 77. This test setup is scenario 1.

5.5.2.1 Translation Experiment

The tracking system performed with an average update rate of 29.2 Hz, what resulted in an average of approximately 1,350 samples for a nominal distance of 0.894 m crossed at 0.0193 m/s. For each illumination condition three test runs were conducted. For each one, accuracy and precision results were obtained. For overall accuracy and precision, individual results were averaged.

Figure 83 shows the plotting of the complete translation experiment under full illumination, comparing nominal and measured travelled distance. Figure 84 shows accuracy and precision results considering three illumination conditions.

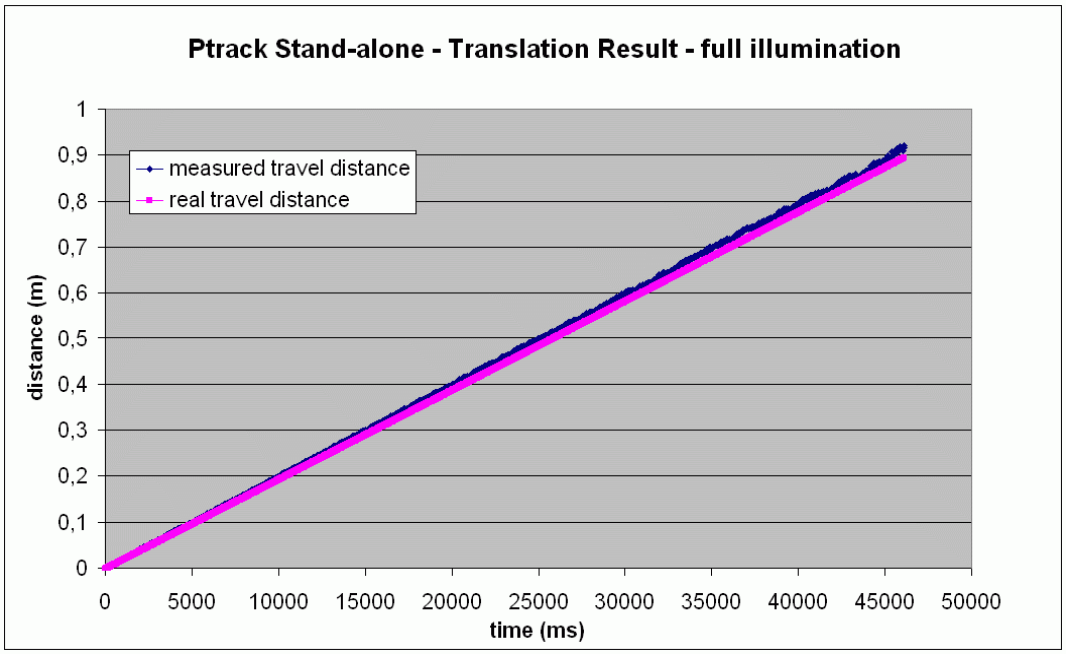

Figure 83 – Plot of Translation Experiment Results in Scenario 1 under Full Illumination

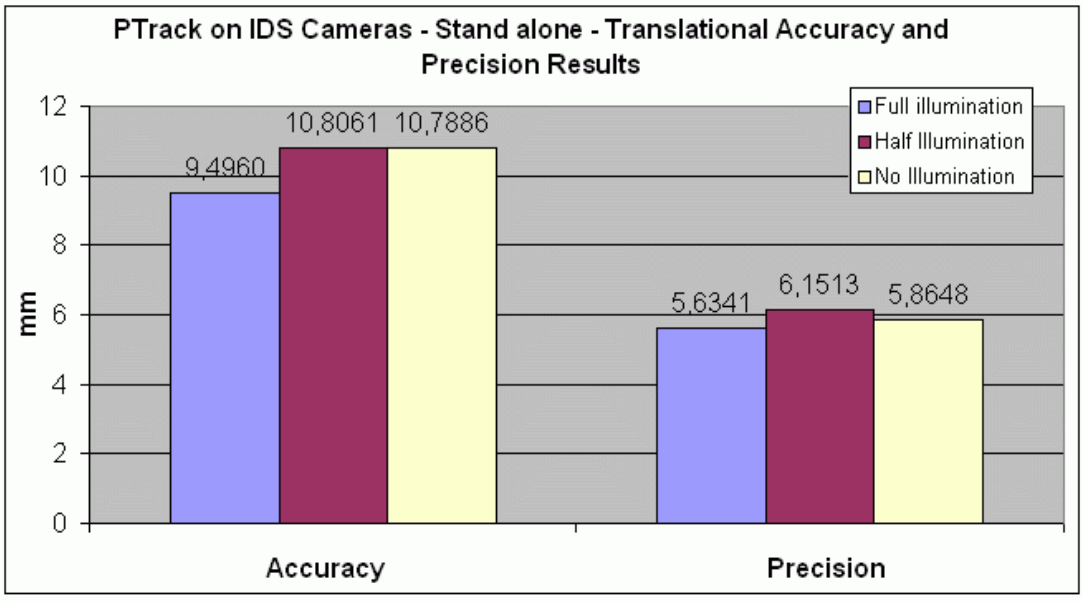

Figure 84 – Accuracy and Precision Results in Scenario 1 under Three Different Illumination Conditions

Figure 83 shows an increasing error in distance estimation on the z axis. This is probably directly related to a camera calibration procedure with imprecisions, what leads to an incorrect calibration of the focal length, resulting in depth estimation

errors. Also reduction of marker size in pixels may have an influence on position estimation, since the number of pixel locations used to calculate marker center is reduced. The average actual distance measured by the tracking system was 0.91 m, close to the nominal distance of 0.894 m.

Figure 84 shows that accuracy and precision values do not vary significantly under different illumination conditions, although the best performance is achieved under full illumination conditions. As a single representative measure of the whole system, overall values for translational accuracy and precision are shown in Table 13.

Table 13 – Average Translational Accuracy and Precision in Scenario 1

Accuracy (mm)	**10.3636**
Precision (mm)	**5.8834**

The tracking system in the stand-alone configuration achieves an average translational accuracy of roughly 10 mm and precision of roughly 6 mm. Considering only full illumination conditions, these values are somewhat better, but yet far from fulfilling specifications defined in section 2.8, in order to be comparable with the systems listed in Table 11.

In the experiments, some indications of imprecise procedures are explicit. Figure 83 shows imprecise calibration of the focal length parameter (due to incorrect calibration procedure or calibration patterns with insufficient construction accuracy). The fact that the system performs slightly better under full illumination condition (Figure 84) contradicts the theoretical statement that presence of optical noise (light emitted or reflected by all other sources except the camera infared flash strobes) worsens system performance. The statement is actually correct, and the unexpected behaviour of the system is probably due to incorrect focal length adjustment, which should be done in presence of infrared light only (section 4.3.2), or infrared flash strobes with insufficient emmiting power, thus needing ambient lighting for sufficient label illumination.

5.5.2.2 Rotation Experiment

The tracking system performed in the rotation experiment with an average update rate of 21.5 Hz. This is probably on account of the fixed reduced distance of 0.80 m between camera and label, which slightly increases image pre-processing computer load, reducing overall system update rate in comparison to the translation experiment.

For each of the following 11 angles of attack a test-run was made yielding on average 4,000 measurement samples for the complete turn: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 and 55 deg. For each angle of attack, the system was tested under the three different illumination conditions previously mentioned, resulting in 33 different test scenarios. Each test yielded as output accuracy and precision values for each of the three angles: Heading, Attitude and Bank.

As a more friendly visual representation of the results, the plot of the reconstructed (tracked) normalized normal vector of the label is used. Figure 85 shows this representation for the system running under full illumination with 20 deg angle of

attack. Figure 86 and Figure 87 show accuracy and precision results for Heading, Figure 88 and Figure 89 for Attitude, and Figure 90 and Figure 91 for Bank.

Table 14 shows the results obtained by calculating average accuracy and precision for all angle of attack values, as well as for all illumination conditions. Averaging among the values of the three angles yields overall rotational accuracy and precision values for the tracking system, which serve as a representative quantification of the complete system's performance. These values are shown in Table 15.

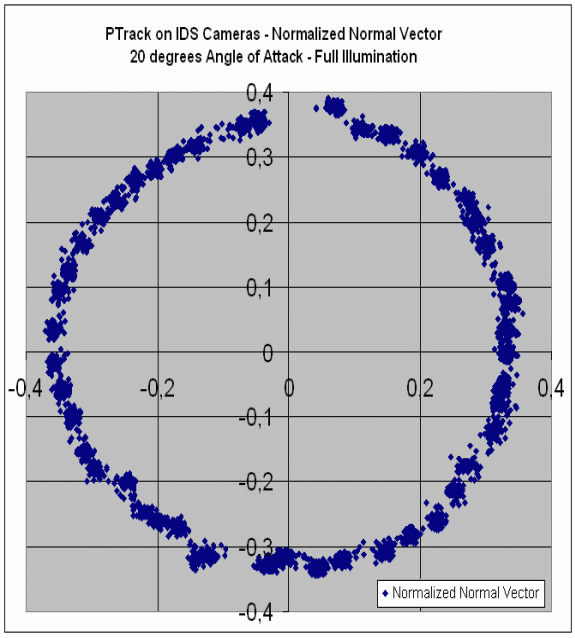

Figure 85 – Normalized Normal Vector of Label in Scenario 1 under Full Illumination, with 20 deg Angle of Attack

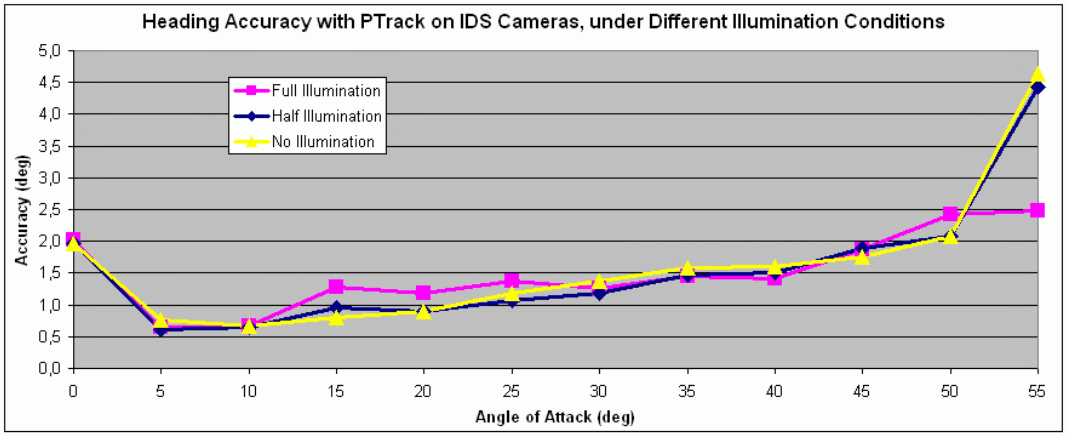

Figure 86 – Accuracy Results for Heading Angle in Scenario 1 under Three Different Illumination Conditions

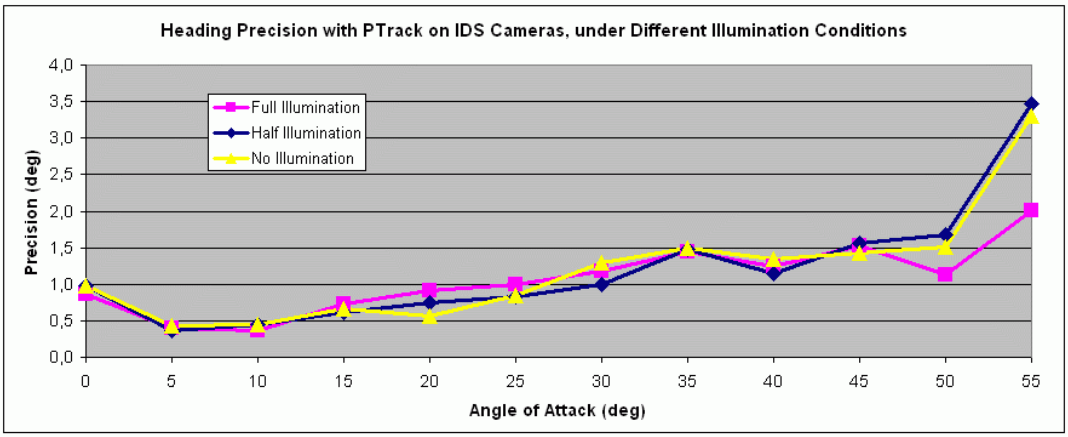

Figure 87 – Precision Results for Heading Angle in Scenario 1 under Three Different Illumination Conditions

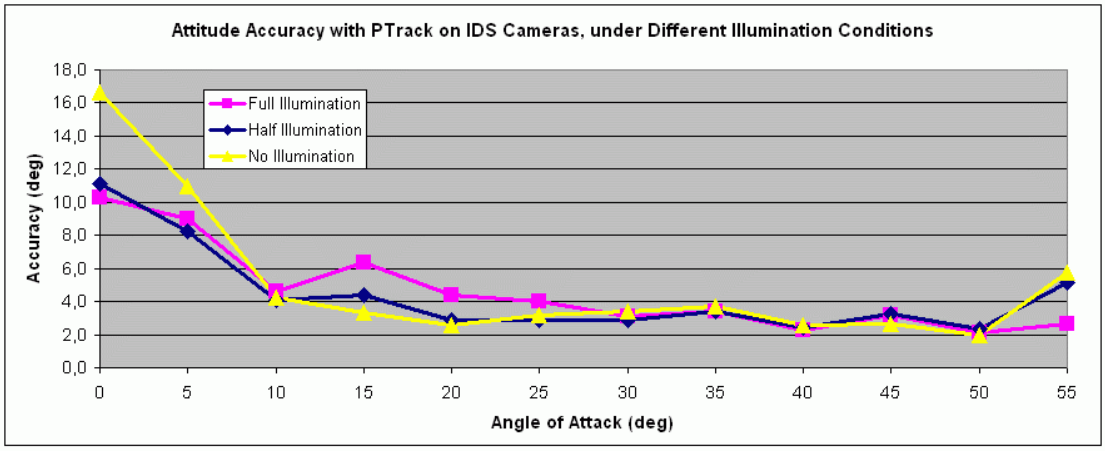

Figure 88 – Accuracy Results for Attitude Angle in Scenario 1 under Three Different Illumination Conditions

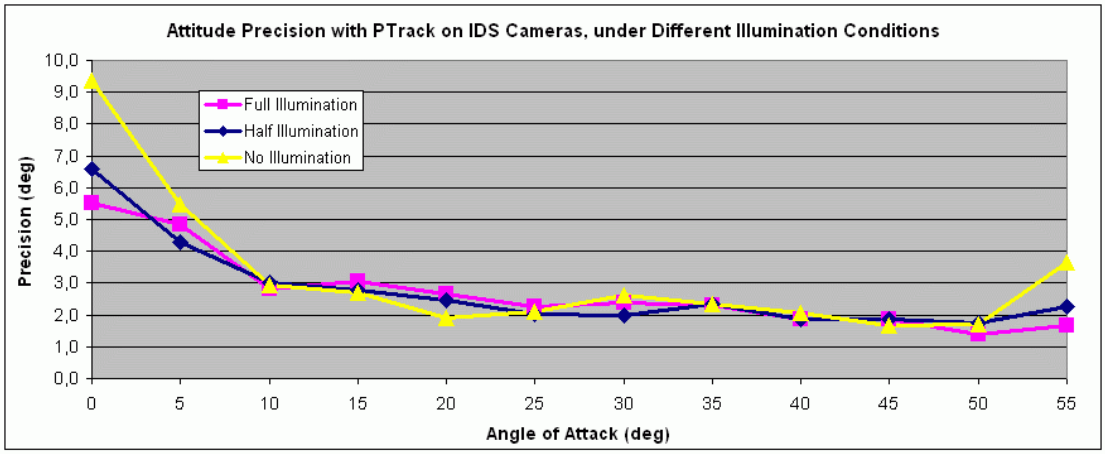

Figure 89 – Precision Results for Attitude Angle in Scenario 1 under Three Different Illumination Conditions

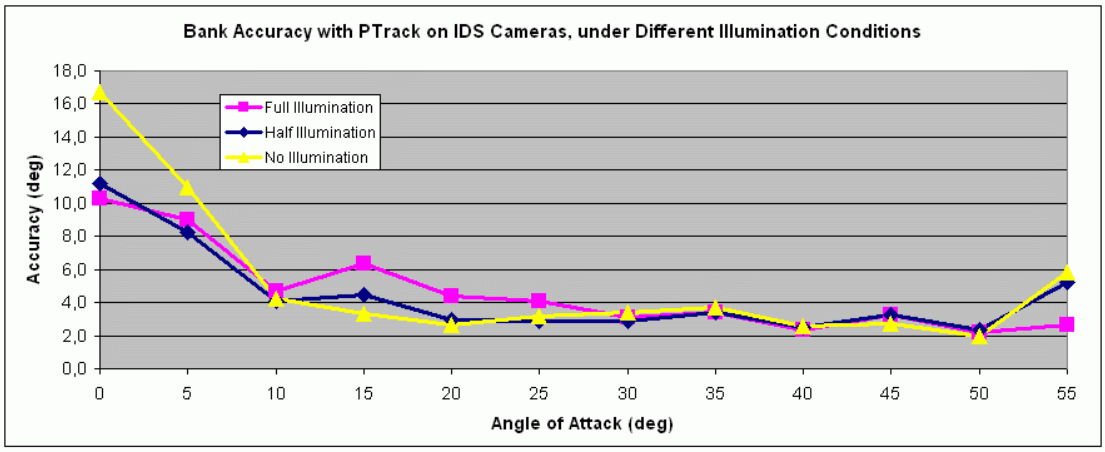

Figure 90 – Accuracy Results for Bank Angle in Scenario 1 under Three Different Illumination Conditions

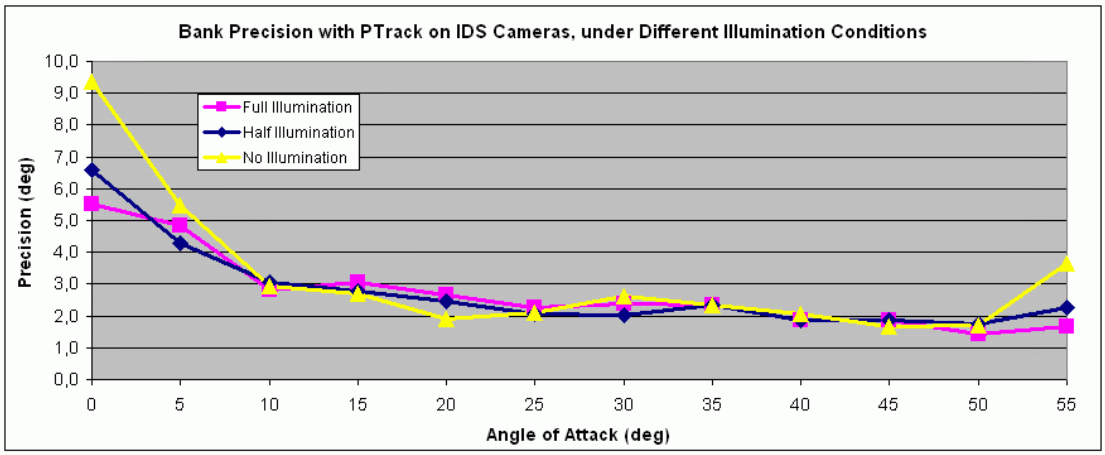

Figure 91 – Precision Results for Bank Angle in Scenario 1 under Three Different Illumination Conditions.

Table 14 – Overall Averaged Accuracy and Precision for Heading, Attitude and Bank in Scenario 1.

Angle	Value	Full Illum.	Half Illum.	No Illum.	Average
Heading	**Accuracy (deg)**	1.5034	1.5565	1.6090	**1.5563**
	Precision (deg)	1.0665	1.1935	1.1927	**1.1509**
Attitude	**Accuracy (deg)**	4.6175	4.4160	5.0856	**4.7064**
	Precision (deg)	2.7143	2.7717	3.2083	**2.8981**
Bank	**Accuracy (deg)**	4.6483	4.4490	5.1157	**4.7377**
	Precision (deg)	2.7216	2.7776	3.2140	**2.9044**

Table 15 – Overall Rotational Accuracy and Precision in Scenario 1.

Accuracy (deg)	**3.6668**
Precision (deg)	**2.3178**

As seen on Table 14, accuracy and precision values tend to get worse with decreasing illumination level. Table 16 shows the comparison between measured and nominal angle of attack values.

Table 16 – Comparison between Measured and Nominal Angle of Attack Values in Scenario 1 under Different Illumination Conditions.

Angle of Attack (deg)			
Nominal	**Measured**		
	Full Illumination	**Half Illumination**	**No Illumination**
0	3.1341	3.1770	3.1498
5	4.9399	5.0256	5.0045
10	10.2410	10.1972	10.2591
15	14.9659	14.8884	14.9378
20	20.1395	20.1342	20.1337
25	25.2639	25.2847	25.2746
30	29.8698	29.8813	29.9171
35	34.8711	34.8549	34.8913
40	40.3217	40.3322	40.3370
45	45.1746	45.1322	45.1563
50	50.4294	50.4310	50.3497
55	55.0502	54.9062	55.0330

If the label and the testbed were correctly and precisely positioned – normal vector of label aligned with the optical axis of the camera when angle of attack and rotation angle equal zero -, accuracy and precision values for the Heading angle should be always very low, since there is no variation in Heading angle. The angle of attack corresponds to the Bank angle, and the rotation angle corresponds to the Attitude angle.

Thus, the amount of accuracy and precision measured for Heading actually represents the amount of the lack of precision in the testbed construction and in execution of experiments. In this case, always below 5 deg. Attitude and Bank have similar accuracy and precision values, what is expected since the system does not have any bias in rotation measurement.

Accuracy and precision values measured at angles of attack near null are naturally high, since small changes in the position of the tracked label cause large changes in the reconstructed orientation. This occurs because, in this case, the vector length before normalization is very small, so vector components are magnified during normalization, leading to huge influence of small changes of position on the calculated normal vector. This explains the large values obtained for angle of attack zero.

In Figure 85 the 48 steps of rotation are clearly represented, and the points spread in a circular shape depict the precision of the measurements. In Figure 86 the high accuracy values in angles of attack 50 and 55 deg are explained by a lower reflection capacity of retro-reflective markers in higher angles of attack, what leads to worse marker center estimation and thereafter bad pose estimation.

Overall rotational accuracy and precision are worse than specifications of section 2.8, but performance can be enhanced as explained in section 5.5.2.1.

Regarding lighting, the system performs better under full illumination, possible reasons explained in section 5.5.2.1. Figure 16 shows that angle of attack values are correctly estimated, roughly unaffected by illumination conditions.

5.5.2.3 Frame Rate versus Number of Labels

This experiment evaluates the sensitivity of the performance of the tracking system to an increase in the number of tracked labels. Figure 92 shows the results obtained.

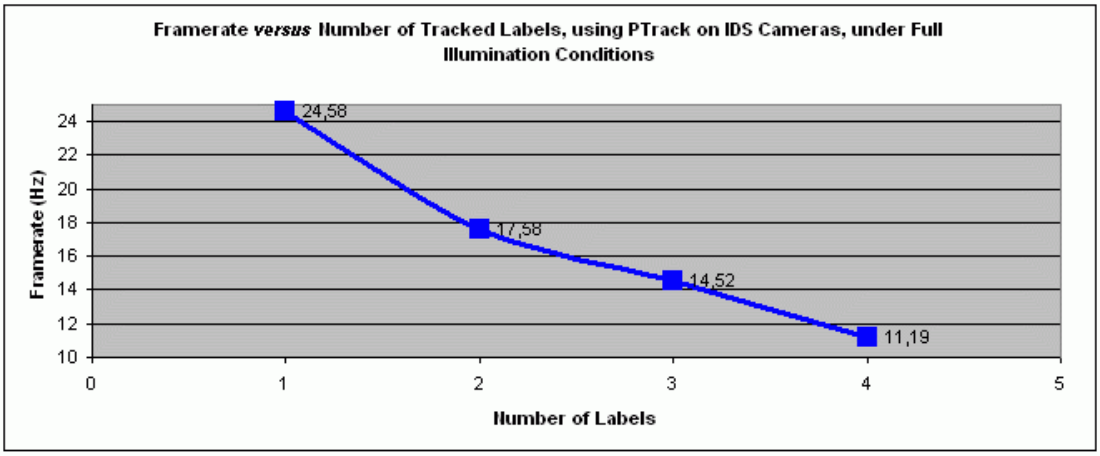

Figure 92 – Results of Test Frame Rate *versus* Number of Tracked Labels in Scenario 1 under Full Illumination Conditions

The results show that PTrack has a performance highly dependant on the number of tracked labels. With 4 labels, the performance reaches only 45% of its original value with only one label.

5.5.3 ARToolKit on IDS Cameras (Scenario 2)

The ARToolKit algorithm had to be adapted in order to work with the IDS Cameras. Basically, its framegrabber (ARFrameGrabber) was modified to use the input images from the IDS uEye UI1210-C device driver (using the proper API) instead of DirectShow drivers. Apart from the framegrabber, all other parts of the algorithm were kept unchanged.

The camera adjustments for ARToolKit are different from the ones used for PTrack, since the daylight blocking filter was removed and consequently the focal

length had to be slightly adapted. This approach makes Scenario 1 slightly different from Scenario 2, but the comparison of both scenarios remains valid. An alternative approach could be, for example, to build retro-reflective ARToolKit labels and then use exactly the same camera configuration for both PTrack and ARToolKit, i.e. keeping the daylight blocking filter when using ARToolKit.

Before testing, usual calibration procedures for ARToolKit were done, using as calibration pattern a white board with black circular patterns plotted on it, as recommended in ARToolKit's documentation.

The test setup with ARToolKit running on IDS Cameras defines scenario 2.

5.5.3.1 Translation Experiment

Since ARToolKit uses the visible region of light spectrum, only tests under full illumination conditions could be performed. Under half illumination some samples could be acquired, but not in enough number to justify execution of tests. Under no illumination tests were of course impossible.

In this case the tracking system performed with an average update rate of 22.1 Hz, what resulted in an average of approximately 1,025 samples for a nominal distance of 0.894 m crossed at 0.0193 m/s. Under full illumination conditions three test runs were conducted. For each one, accuracy and precision results were obtained. For overall accuracy and precision, individual results were averaged. Figure 93 shows the plotting of complete translation experiment results under full illumination.

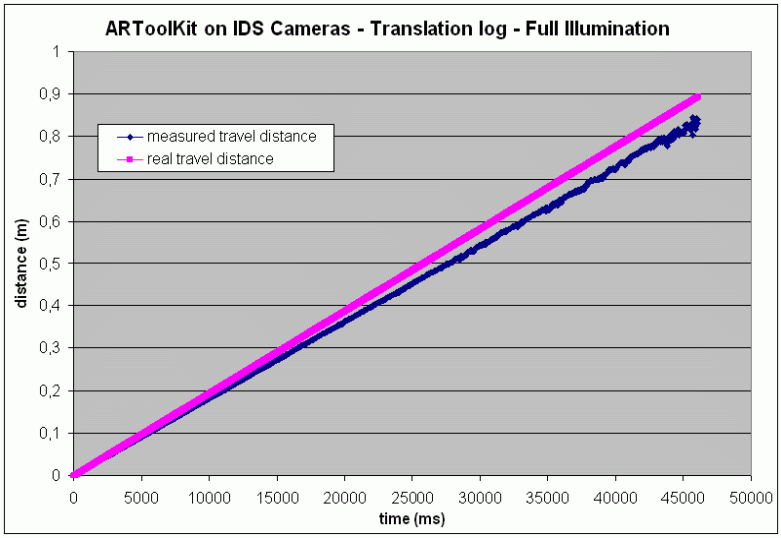

Figure 93 – Plot of Translation Experiment Results in Scenario 2 under Full Illumination

Figure 93 also shows an increasing error in distance estimation on the z axis. As in the case of Figure 83, this is probably due to imprecisions in the camera calibration procedure, what leads to an incorrect calibration of the focal length, resulting in depth estimation errors.

The average actual distance measured by the tracking system was 0.84 m, somewhat different from the nominal distance of 0.894 m. The error can be related to

imprecise estimation of depth values. Overall accuracy and precision values for translation are presented in Table 17.

Table 17 - Average Translational Accuracy and Precision in Scenario 2

Accuracy (mm)	**29.6926**
Precision (mm)	**17.4829**

The performance results of ARToolKit on the cameras used in this work, achieving roughly 30 mm in translational accuracy and 17 mm in translational precision, are comparable to results obtained in (MALBEZIN, 2002), with a different test setup. The increasing error shown in Figure 93 is probably due to imprecise calibration. Also some irregular samples can be seen near to the end of the tracked distance, which occur on account of the proximity with ARToolKit's maximum reach.

5.5.3.2 Rotation Experiment

The tracking system performed also with an average update rate of 22.1 Hz. For each of the following 11 angles of attack a test-run was made yielding on average 4,050 measurement samples for the complete turn: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 and 55 deg. As output each test yielded accuracy and precision values for each of the three angles: Heading, Attitude and Bank.

The plot of the reconstructed (tracked) normalized normal vector of the label is used to show tracking data quality. Figure 94 shows this representation for the system running under full illumination with 20 deg angle of attack. Figure 95 and Figure 96 show accuracy and precision results for Heading, Attitude and Bank angles.

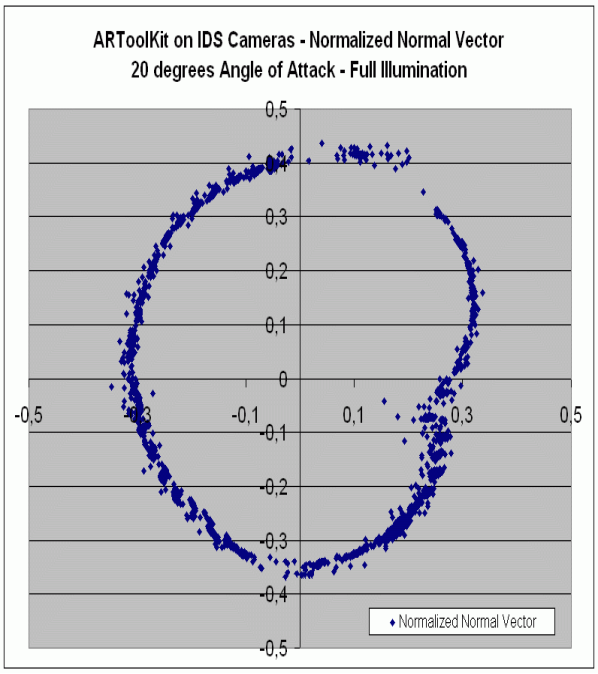

Figure 94 – Normalized Normal Vector of Label in Scenario 2 under Full Illumination with 20 deg Angle of Attack.

Averaging accuracy and precision for all angles of attack, and afterwards averaging again the values of the three angles, results in overall rotational accuracy and precision values for the tracking system, as shown in Table 18.

Figure 94 differs coarsely from a perfect circle, showing ARToolKit's deficiency at some angles of rotation and overall imprecision in calculation of the label's normalized normal vector. Heading precision and accuracy values, as seen in Figure 95 and Figure 96, are close to zero, even closer than the values obtained with the stand-alone version of PTrack (section 5.5.2.2), showing that the whole setup was better aligned. Attitude and Bank angles have very similar results. Very poor accuracy and precision values for Attitude and Bank were obtained at angles of attack 0 and 5 deg, due to the same reasons presented for PTrack, but with extremely higher values - reaching 45 deg in accuracy at angle of attack zero.

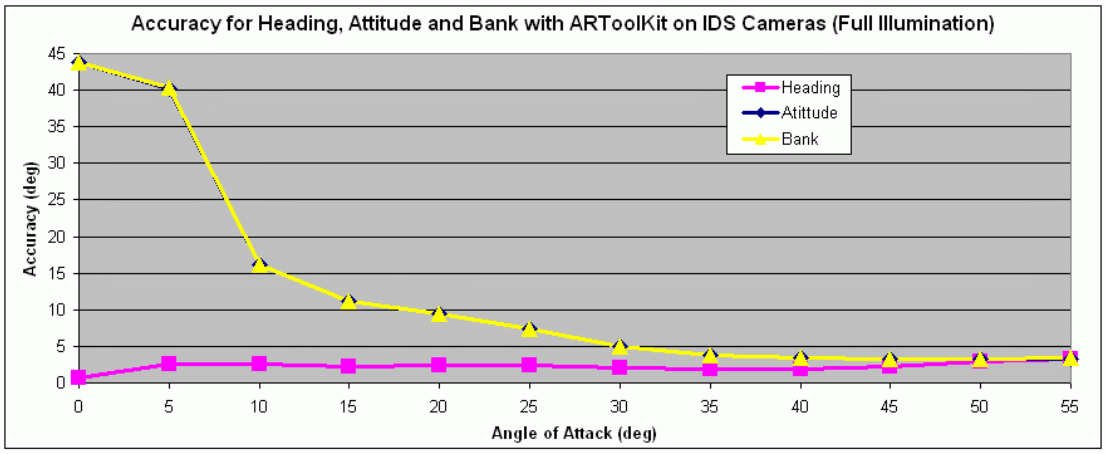

Figure 95 – Accuracy Results for Heading, Attitude and Bank Angles in Scenario 2 under Full Illumination

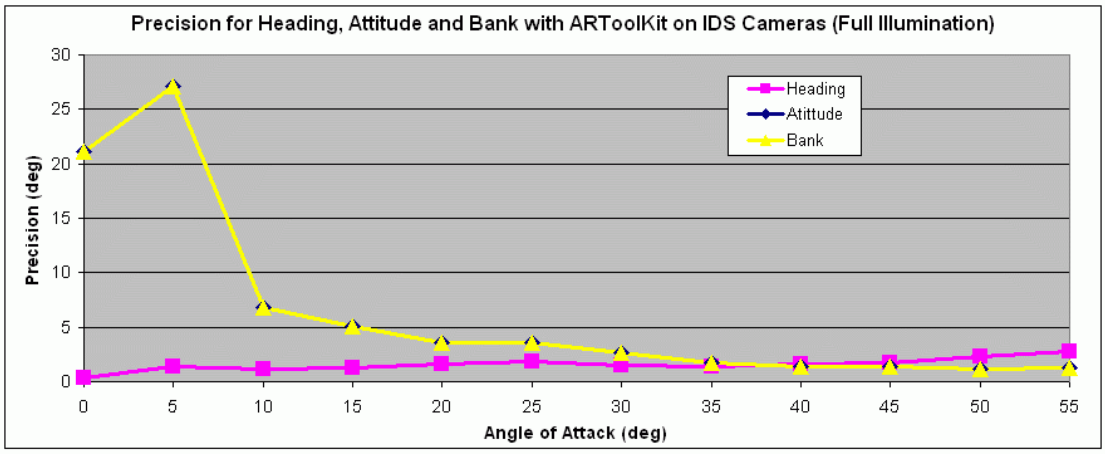

Figure 96 – Precision Results for Heading, Attitude and Bank Angles in Scenario 2 under Full Illumination

Table 19 shows a comparison between measured and nominal angle of attack values.

Table 18 – Overall Averaged Accuracy and Precision Values for Heading, Attitude and Bank Angles in Scenario 2

Angle	Average Values	
	Accuracy (deg)	Precision (deg)
Heading	2.2748	1.5728
Attitude	12.4901	6.4063
Bank	12.5315	6.4091
Overall Average	**9.0988**	**4.7961**

Table 19 – Comparison between Measured and Nominal Angle of Attack Values in Scenario 2

Angle of Attack (deg)	
Nominal	**Measured (Full Illumination)**
0	1.8445
5	4.5502
10	10.2637
15	15.2763
20	19.6623
25	24.6757
30	30.4380
35	34.8986
40	40.3249
45	45.2040
50	49.8929
55	55.5189

Overall accuracy reaches roughly 9 deg, and overall precision, roughly 5 deg. These values are two times larger than but still comparable to the ones of PTrack on IDS cameras. Table 19 shows that angle of attack values are correctly estimated at roughly all points.

5.5.3.3 Frame Rate versus Number of Labels

This experiment measures the sensitivity of the performance of the tracking system to an increase in the number of tracked labels. Figure 97 shows the results obtained.

Results show that ARToolKit has a performance degradation of only 20% when tracking 5 labels, in comparison to tracking of only one label, thus reaching 80% of its original performance.

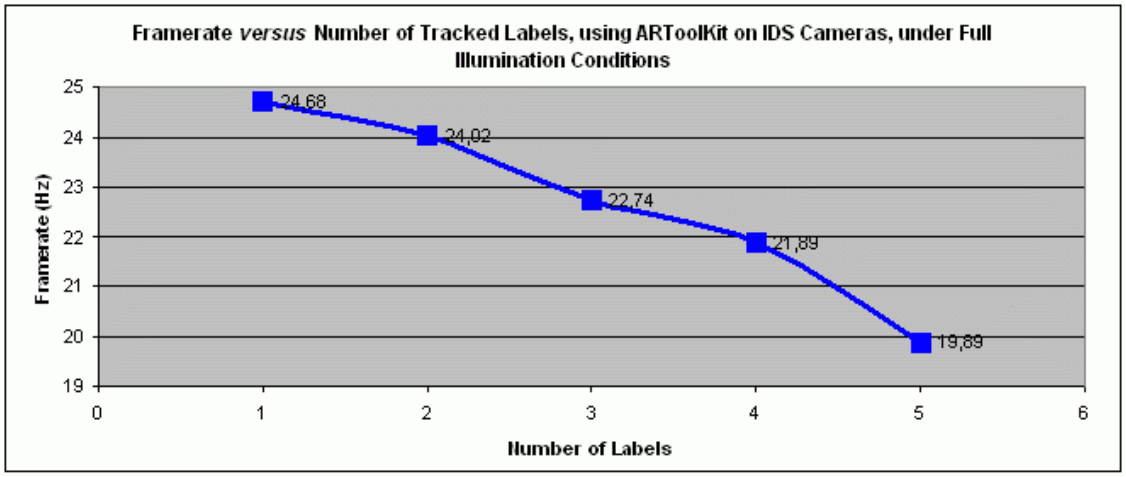

Figure 97 – Results of Test Frame Rate *versus* Number of Tracked Labels under Full Illumination Conditions in Scenario 2

5.5.4 PTrack on IDS Cameras in Multiple-camera Configuration (Scenario 3)

When in multiple-camera configuration (wide-area tracking) the testbed topology is similar to the stand-alone configuration, with the main camera in the same position as the single camera. Additionally, a second camera is positioned 67 cm to the right of the main camera, with optical axis pointing to the label when this is at 1.2 m distance from the main camera, as shown in Figure 98. This setup ensures that both cameras see the label during the whole travelling route. Figure 99 and Figure 100 show the implemented testbed. This test setup defines scenario 3.

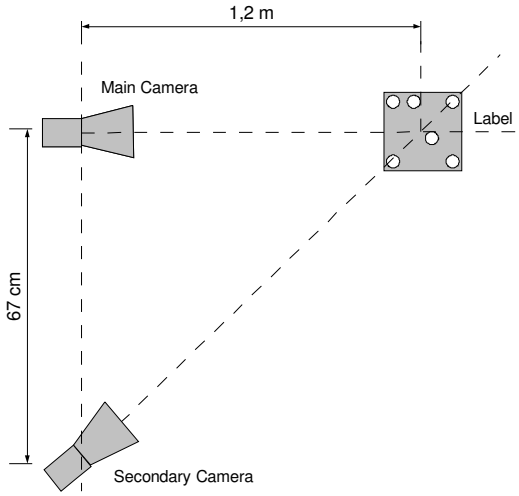

Figure 98 – Multiple-camera Topology for Experiment in Scenario 3

Ideally, both cameras should not have their optical axes aligned with the label, in order to have equivalent conditions for tracking information quality, based on the criterion defined in section 4.2. Nevertheless, the testbed built was planned for

single-camera systems, and space limitations did not allow the use of different positioning for cameras, resulting in one camera with optical axis aligned with the label.

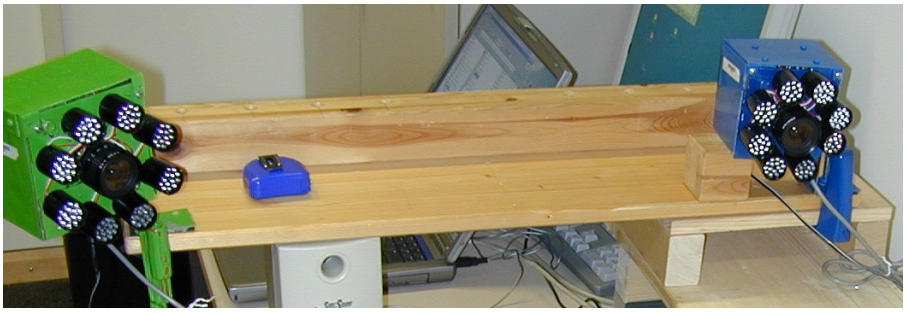

Figure 99 – View of both Cameras in PTrack Multiple-camera Configuration

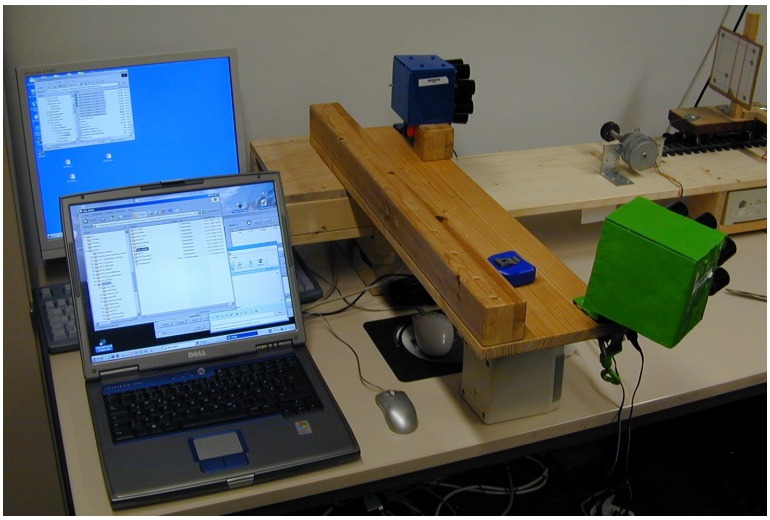

Figure 100 – View of both Cameras, Label (on Low Side Car) and Computers in Scenario 3

Considering this topology as multiple-sensor system with sensor fusion, the expected results would be an enlarged working volume and increased exactness, as stated in section 3.7. However, due to imprecisions in calibration procedures and absence of a synchronization technique for data originated in different modules, exactness is expected to be slightly deteriorated.

Experiments were conducted similarly to stand-alone configuration, except that another one-camera tracking module was connected to the system, as represented in Figure 69. The new module was executed on the same computer as the first one, so the frame rate of each module was drastically reduced due to sharing of same processing resources.

5.5.4.1 Translation Experiment

The tracking system performed with an average update rate of 12.7 Hz, due to both instances of the tracking application running on the same machine. This resulted

in an average of approximately 650 samples for a nominal distance of 0.894 m crossed at 0.0193 m/s. For each illumination condition three test runs were conducted. For each one, accuracy and precision results were obtained. For overall accuracy and precision, individual results were averaged.

Figure 101 shows the plotting of the complete translation experiment under full illumination, comparing nominal and measured travelled distance. Figure 102 shows accuracy and precision results considering the three illumination conditions.

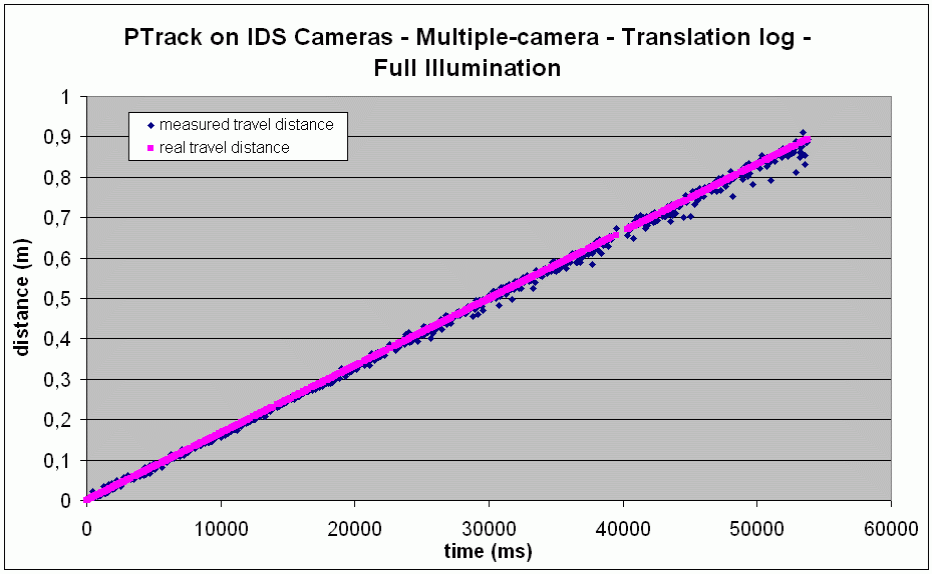

Figure 101 – Plot of Translation Experiment Results in Scenario 3 under Full Illumination

Figure 101 shows intermittent data with large amount of error, especially in large distances from the camera. This is due to consideration of data from the secondary camera as highly important, probably because it was the only available data at the time. In the positions where data from the secondary camera is exclusively considered, the camera sees the label on the very right side, far from the camera's optical axis, thus with low quality, as stated in section 4.1.4. So, sporadically, the only received tracking data is the one with poor quality, which is considered by the system. This occurs because there is no synchronization in the central module of data received from different one-camera tracking modules referring to the same label. Synchronization is a future enhancement of this work.

The average actual distance measured by the tracking system was 0.890 m, very close to the nominal distance of 0.894 m.

Figure 102 shows that accuracy and precision deteriorate with decreasing illumination level, thus the best performance is achieved under full illumination conditions. This manifestation was already detected in PTrack with stand-alone configuration, but it is more intense in the multiple-camera case, probably due to the fact that both cameras have imprecisions in focal length adjustment.

Overall values for translational accuracy and precision are shown in Table 20. The large overall accuracy and precision values can be in part explained by the absence of a synchronization technique between tracking data received from the two

one-camera tracking modules. Running software applications of both modules in the same computer minimizes this effect, but it remains as a hinderance for higher system performance. The poor accuracy can be also related to imprecise calibration of the extrinsic parameters, done accordingly to section 4.3.2. Overall translational accuracy is better than in stand-alone configuration, but precision is worse.

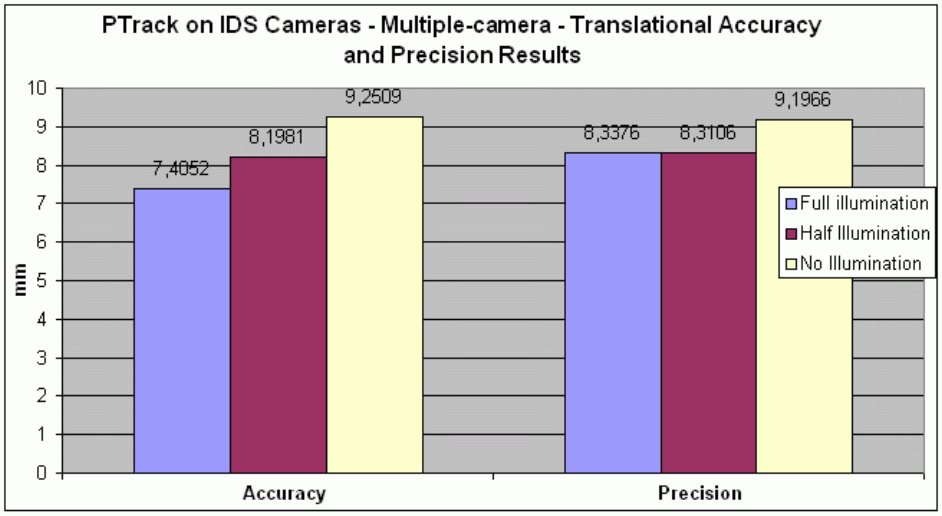

Figure 102 – Accuracy and Precision in Scenario 3 under Three Different Illumination Conditions

Table 20 – Average Translational Accuracy and Precision in Scenario 3

Accuracy (mm)	**8.2847**
Precision (mm)	**8.6149**

5.5.4.2 Rotation Experiment

For rotation experiment, the label was positioned exactly as in the experiment of the stand-alone configuration. Due to the viewing angle between the secondary camera and the label – in other words, the angle of attack of the label in relation to the secondary camera -, the experiments could be executed only up to an angle of attack of 15 deg.

The tracking system performed in the rotation experiment with an average update rate of 12.1 Hz. For each of the following angles of attack a test-run was made, yielding on average 2,340 measurement samples for the complete turn: 0, 5, 10 and 15 deg. For each angle of attack, the system was tested under the three different illumination conditions, resulting in 12 different test scenarios. Each test yielded as output accuracy and precision values for each of the three angles: Heading, Attitude and Bank.

Figure 103 shows the normalized normal vector of the label tracked by the system running under full illumination with 15 deg angle of attack. Figure 104, Figure

105 and Figure 106 show accuracy and precision results for Heading, Attitude and Bank, respectively.

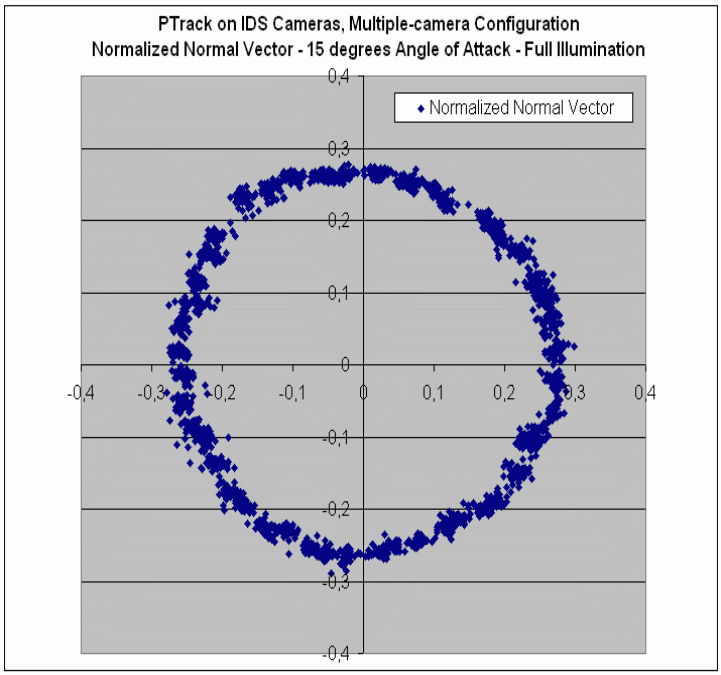

Figure 103 – Normalized Normal Vector of Label in Scenario 3 under Full Illumination with 15 deg Angle of Attack

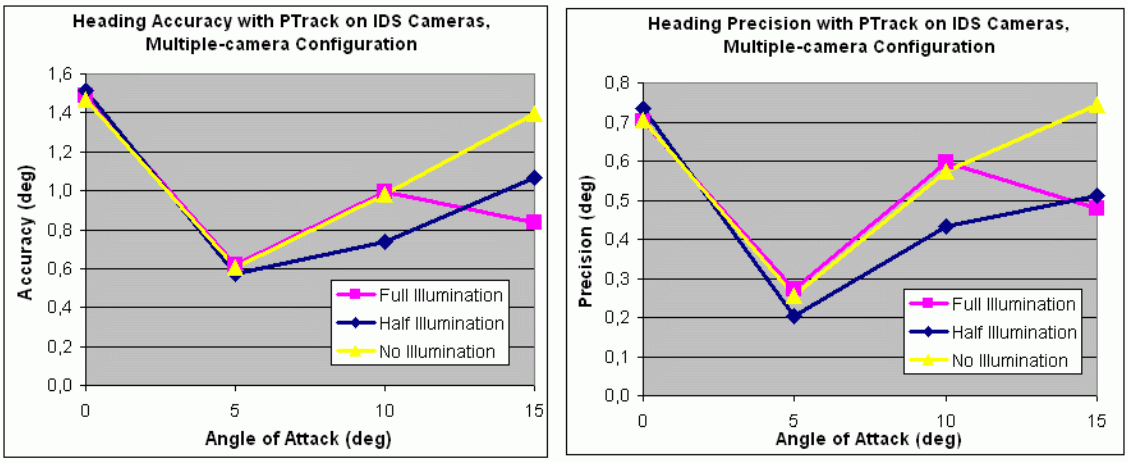

Figure 104 – Accuracy and Precision Results for Heading Angle in Scenario 3 under Different Illumination Conditions

Averaging the values of accuracy and precision for all angle of attack values, as well as for all illumination conditions, yields values shown in Table 21. Averaging the values of the three angles yields overall rotational accuracy and precision values for the tracking system, as shown in Table 22.

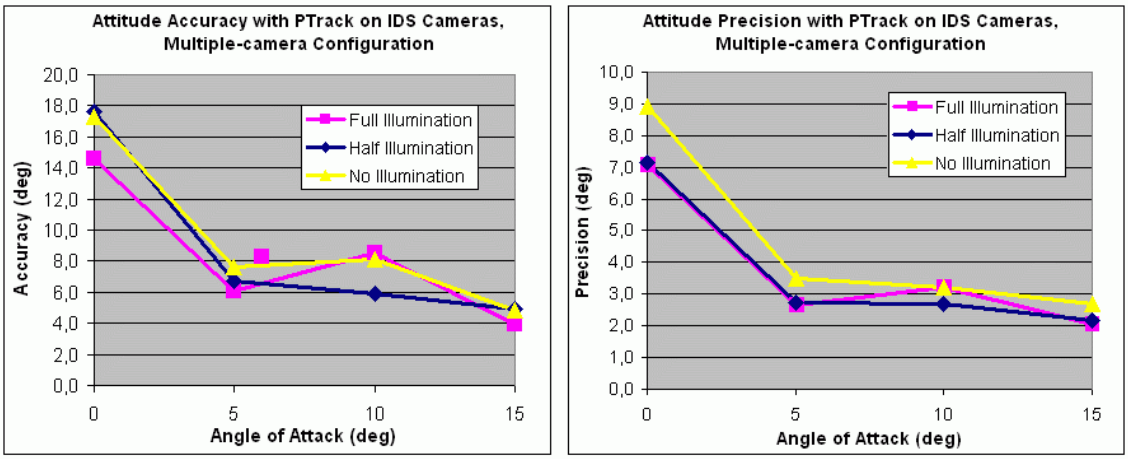

Figure 105 – Accuracy and Precision Results for Attitude Angle in Scenario 3 under Different Illumination Conditions

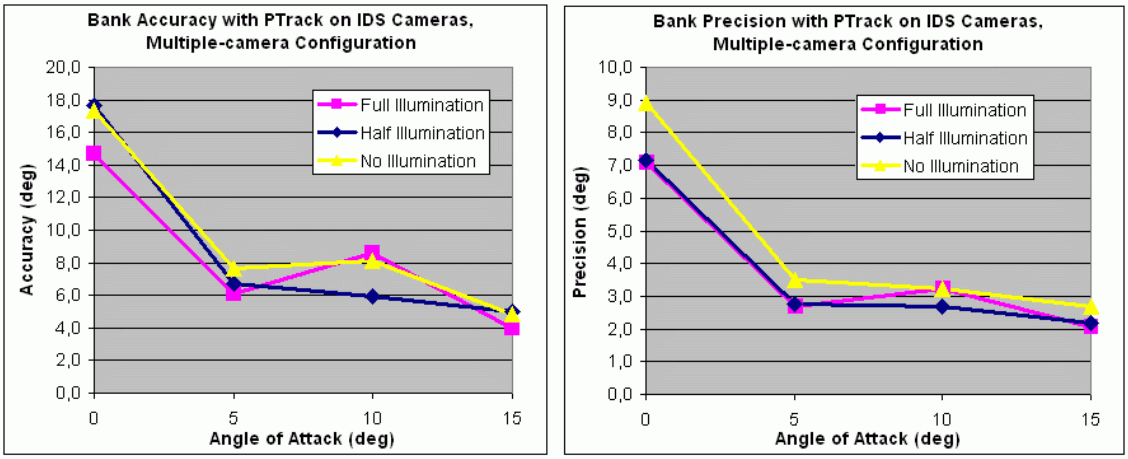

Figure 106 – Accuracy and Precision Results for Bank Angle in Scenario 3 under Different Illumination Conditions

Table 21 – Overall Averaged Accuracy and Precision for Heading, Attitude and Bank Angles in Scenario 3

Angle	Value	Full Illum.	Half Illum.	No Illum.	Average
Heading	**Accuracy (deg)**	0.9855	0.9726	1.1118	**1.0233**
	Precision (deg)	0.5126	0.4698	0.5692	**0.5172**
Attitude	**Accuracy (deg)**	8.2990	8.7782	9.4633	**8.8468**
	Precision (deg)	3.7552	3.6913	4.5764	**4.0076**
Bank	**Accuracy (deg)**	8.3311	8.8111	9.4981	**8.8801**
	Precision (deg)	3.7587	3.6971	4.5767	**4.0108**

As seen on Table 21, accuracy and precision values tend to get slightly worse with decreasing illumination level. Table 23 shows the comparison between measured and nominal angle of attack values.

Table 22 – Overall Rotational Accuracy and Precision in Scenario 3

Accuracy (deg)	**6.2501**
Precision (deg)	**2.8452**

Table 23 – Comparison between Measured and Nominal Angle of Attack Values in Scenario 3 under Different Illumination Conditions

Angle of Attack (deg)			
Nominal	**Measured**		
	Full Illumination	**Half Illumination**	**No Illumination**
0	2.5053	2.4675	2.4631
5	5.3769	4.9524	5.0742
10	9.8704	10.0035	10.0103
15	15.3152	15.2988	15.2034

Heading precision and accuracy are very high, as should be in theory. Attitude and Bank angles have very similar results. Very poor accuracy and precision for Attitude and Bank were obtained at angle of attack zero, due to the same reasons presented for PTrack and ARToolKit.

Again, under full illumination condition, the system performs better than under half or no illumination. For rotation, overall accuracy value reaches 6.25 deg, almost twice as high as PTrack in stand-alone configuration, and overall precision value reaches 2.8 deg, close to the 2.3 of PTrack in stand-alone setup. Table 23 shows that angle of attack values are correctly estimated at roughly all points.

The multiple-camera configuration reaches larger working range than stand-alone configuration (16 m³, according to section 4.1.1). Figure 107 depicts a sketch representation of this enlargement for the configuration used in this test (Figure 98).

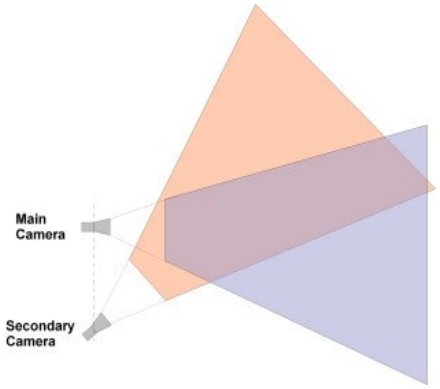

Figure 107 – Sketch of Increase in Working Range in Multiple-camera Tracking Configuration, when compared to Stand-alone Configuration

5.6 COMPARISONS AND DISCUSSION

In this section, the performance results obtained in this work are compared to each other and to results attained in previous publications.

5.6.1 PTrack on IDS Cameras *versus* PTrack on ART Cameras

Santos (SANTOS, 2005) used ART Cameras for the performance tests of PTrack (see section 2.6.2). Those cameras have embedded image pre-processing and deliver 2D marker coordinates to the computer. Table 24 shows a comparison of PTrack running on ART Cameras and on the hardware module developed in this work (IDS Cameras), in stand-alone configuration, considering overall averaged results.

Regarding translational performance, PTrack performed 5 times worse on IDS Cameras than on ART Cameras, but still remained in an acceptable range of roughly 10 mm accuracy.

In rotational performance, PTrack on IDS Cameras performed almost as good as on ART Cameras, what is an impressive result, since ART cameras have professional manufacturing precision and the built-in algorithms are the result of continuous enhancement and years of experience, while the hardware implemented in this work was built with prototyping precision and the algorithms are in their first version.

The difference in the update rate is a direct consequence of using PC based image pre-processing instead of embedded processing, as well as using algorithm steps which are not as optimized and efficient as the steps executed in pre-processing by ART cameras.

Table 24 – Performance Comparison of PTrack on ART and IDS Cameras

Attribute		PTrack on ART Cameras	PTrack on IDS Cameras
Translational	**Accuracy (mm)**	1.9985	10.3636
	Precision (mm)	-	5.8834
Rotational	**Accuracy (deg)**	3.0785	3.6668
	Precision (deg)	-	2.3178
Maximum Measured Update Rate (Hz)		57	29

5.6.2 ARToolKit *versus* PTrack, both on IDS Cameras (Scenarios 1 and 2)

ARToolKit was modified to work on the same IDS Cameras as PTrack, with minor adaptions in its framegrabber module and also in the camera. Table 25 shows a direct comparison of PTrack and ARToolKit, both running on IDS Cameras, considering overall averaged results.

Table 25 – Performance Comparison between Scenarios 1 and 2

Attribute		PTrack on IDS Cameras	ARToolKit on IDS Cameras
Translational	**Accuracy (mm)**	10.3636	29.6926
	Precision (mm)	5.8834	17.4829
Rotational	**Accuracy (deg)**	3.6668	9.0988
	Precision (deg)	2.3178	4.7961
Maximum Measured Update Rate (Hz)		29	22.1

ARToolKit performs between 2 and 3 times worse than PTrack and similar to the performance obtained in previous results, for example when running with a regular webcam (see section 5.6.3).

In translational performance, comparison of Figure 83 and Figure 93 shows that the ARToolKit has a larger depth estimation error, probably due to larger calibration error in focal length parameter. ARToolKit has a calibration procedure which frequently requires user participation, thus being error-prone and not as reliable as semi or full automatic procedures as the one used in PTrack.

Figure 108 and Figure 109 show detailed comparison data for the rotation experiment. Since Heading angle is a measure of imprecisions in calibration and test procedures, and Attitude and Bank angles have similar values, only Bank angle is presented as comparison measure.

In general, PTrack performs better than ARToolKit, but accuracy values are very close in angles of attack between 35 and 50 deg. In this same interval, precision of ARToolKit is even slightly better than PTrack.

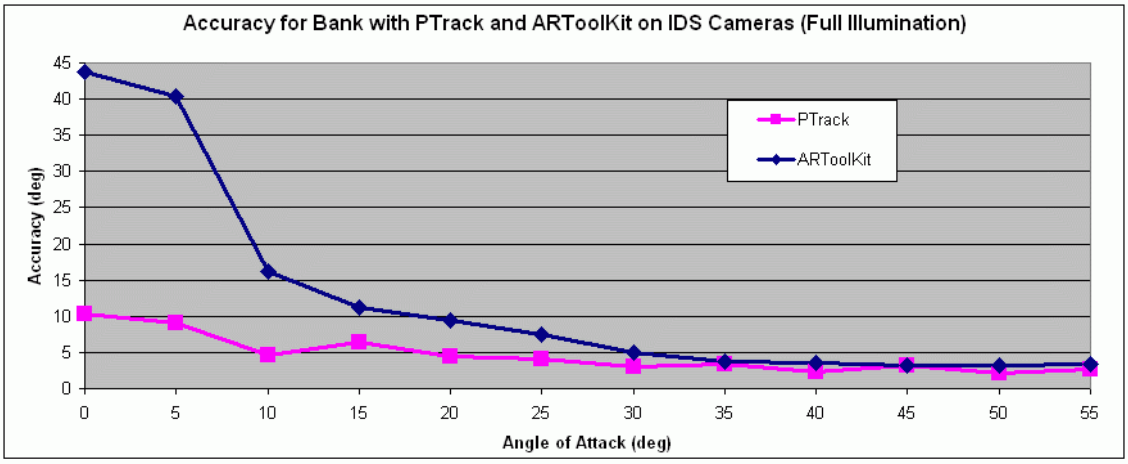

Figure 108 – Comparison of Accuracy Values for Bank Angle between Scenarios 1 and 2 under Full Illumination

Comparing achieved update rates of both systems, it can be seen that PTrack has higher sensitivity to the distance between label and camera than ARToolKit. PTrack showed lower update rate when the label was fixed and positioned for rotation tests (0.8 m from camera) than when running along the track in translation tests, considering the

average update rate. ARToolKit showed no influence from distance of label on the update rate of the system.

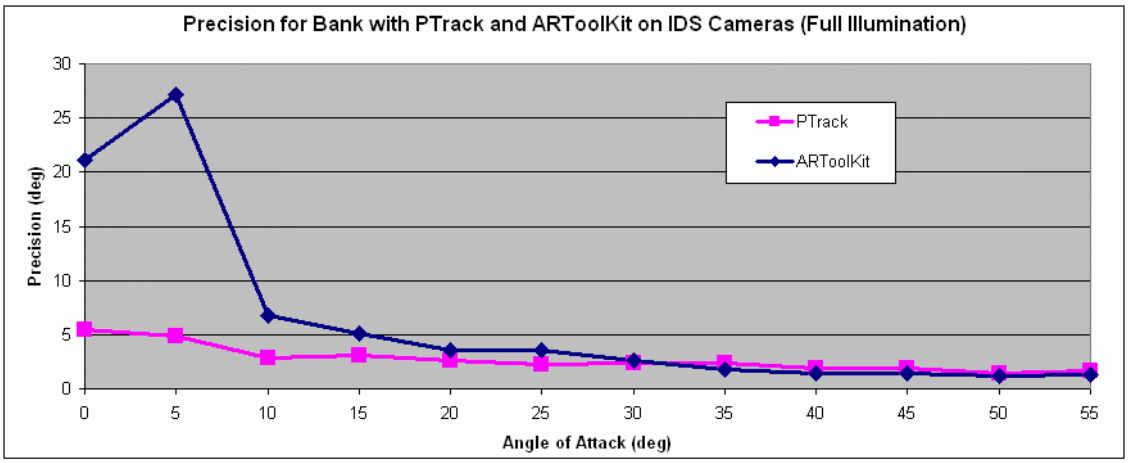

Figure 109 – Comparison of Precision Values for Bank Angle between Scenarios 1 and 2 under Full Illumination

Comparing results of frame rate tests, it can be observed that ARToolKit's performance is less sensitive to the number of tracked labels than PTrack's, as can be seen in Figure 110.

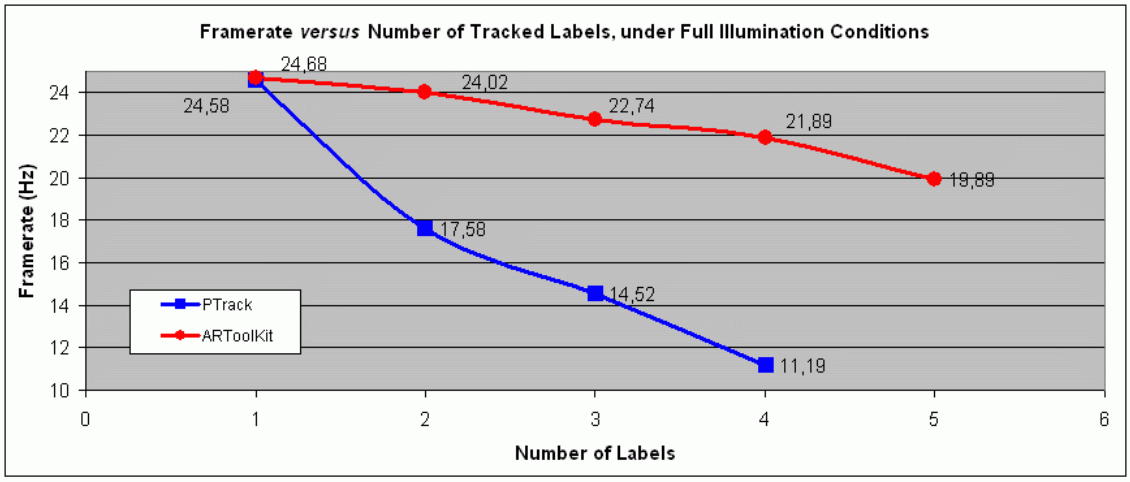

Figure 110 – Comparison Results of Test Frame Rate *versus* Number of Labels, under Full Illumination Conditions, in Scenarios 1 and 2

The plots of normalized normal vector shown in Figure 85 and Figure 94 are typical for smaller angles of attack (below 35 deg). In this range, PTrack performs better than ARToolKit. In higher angles of attack, both systems have very similar performance, with the normalized normal vector being very smooth and correctly reconstructed, as shown for example in Figure 111 and Figure 112.

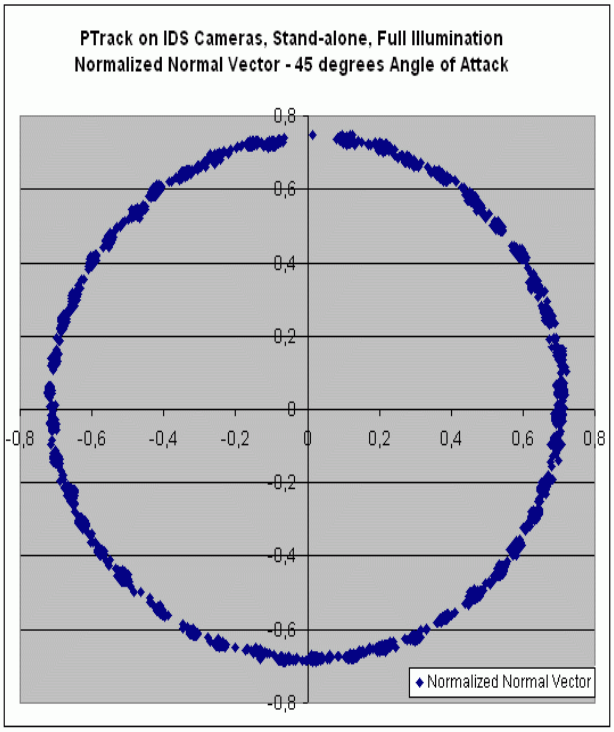

Figure 111 – Normalized Normal Vector of Label in Scenario 1 under Full Illumination with 45 deg Angle of Attack

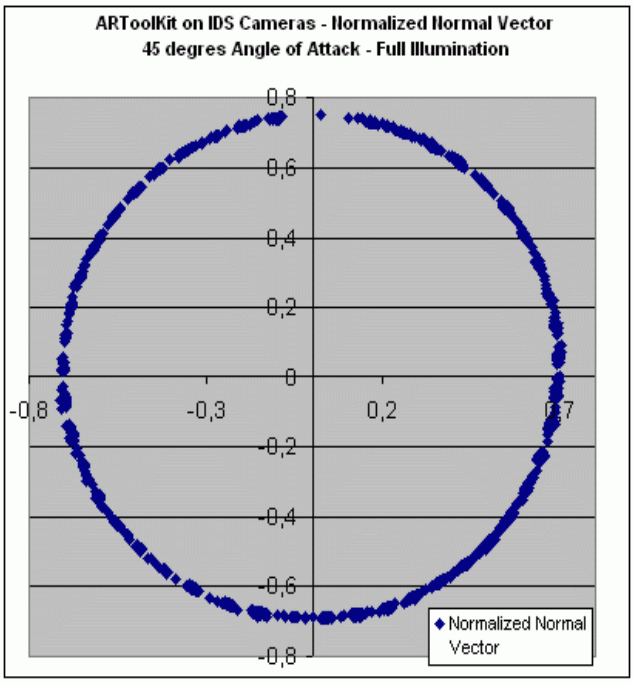

Figure 112 – Normalized Normal Vector of Label in Scenario 2 under Full Illumination with 45 deg Angle of Attack

5.6.3 ARToolKit on IDS Cameras *versus* ARToolKit on Webcams

In (SANTOS, 2005) ARToolKit was tested using a Logitech Express Webcam, with 352 x 288 pixels resolution and 15 Hz maximum frame rate. Table 26 compares the results obtained with the webcam with the results obtained in this work, using an IDS uEye UI1210-C camera accessed through a modified ARToolKit framegrabber module.

Table 26 – Direct Performance Comparison of ARToolKit running on Webcams and on IDS Cameras (Scenario 2)

Attribute		ARToolKit on Webcam	ARToolKit on IDS Cameras
Translational	**Accuracy (mm)**	16.81	29.6926
	Precision (mm)	-	17.4829
Rotational	**Accuracy (deg)**	26.12	9.0988
	Precision (deg)	-	4.7961
Maximum Measured Update Rate (Hz)		10	22.1

Running on IDS Cameras, ARToolKit showed worse translational accuracy than on the webcam, probably due to imprecise calibration, which caused large depth estimation error (section 5.5.3.1). Rotational accuracy is almost 3 times better. Rotational precision was not directly presented in (SANTOS, 2005), but can be indirectly inferred as being between 15 and 20 deg. In comparison with (PIEKARSKI, 2002), where rotational precision values between 20 and 30 deg were measured, ARToolKit running on IDS cameras performed roughly 6 times better.

5.6.4 PTrack: Stand-alone *versus* Multiple-camera Setups (Scenarios 1 and 3)

As stated in section 5.5.4, although better accuracy and precision values would be expected from a system implementing sensor fusion techniques, this was not the case in the multiple-camera configuration of PTrack, due to imprecise calibration and absence of synchronization techniques for data from different sources. Table 27 shows a direct comparison of PTrack on both configurations.

Table 27 – Performance Comparison Between Scenarios 1 and 3

Attribute		PTrack Stand-alone	PTrack Multiple-camera
Translational	**Accuracy (mm)**	10.3636	8.2847
	Precision (mm)	5.8834	8.6149
Rotational	**Accuracy (deg)**	3.6668	6.2501
	Precision (deg)	2.3178	2.8452

Regarding translation, the precision value was higher in multiple-camera than in stand-alone configuration, as expected. The accuracy value, however, was lower, showing that the contribution of sensor fusion techniques exceeded the deterioration caused by the previously mentioned factors.

Regarding rotational performance, the multiple-camera setup had worse accuracy but almost the same precision as the stand-alone configuration.

As the example shown in Figure 107, the working range in a multiple-camera setup is larger than in a stand-alone configuration. The total increase depends on the positioning of cameras.

5.7 SUMMARY AND CONCLUDING REMARKS

In this chapter the complete evaluation procedure for the tracking system developed within the scope of this work was presented. The goals and requirements of the evaluation tests for the tracking system were defined, and a general description of each experiment was given. Basically, rotation and translation tests must be executed in order to evaluate the system performance when measuring orientation and location of a tracked label.

The testbed built was then described in detail, including all software and hardware parts which had to be developed specifically for the tests. Planning and implementation were based partially on a professional setup for evaluation of tracking systems and partially on previous publications related to the test and evaluation issue. This description for testbed assembly can be used to build a low cost solution for evaluation of tracking systems using any technology, not only optical.

Results obtained from the evaluation of PTrack running on IDS cameras showed that the system performed worse than PTrack running on ART Cameras, but better than expected from the comparison, especially rotation performance which achieved similar values.

In comparison with ARToolKit running on the same camera, PTrack performed between 2 and 3 times better, what relates to translation and rotation exactness, in average. However, PTrack revealed much higher performance sensitivity to the number of labels tracked in the scene, as depicted in Figure 110.

In comparison to its performance running on a regular webcam, ARToolKit had worse translational performance but much better rotational performance when running on the hardware module implemented in this work.

The system running in multiple-camera configuration performed slightly worse than in stand-alone setup, except for the translational accuracy, which reached 8.3 mm, somewhat better than 10.4 mm obtained when using a single tracking module.

A natural comparison which must be done is between specifications of the desired system, listed in section 2.8, and the performance actually obtained. Using the grading method previously defined, Table 28 shows the score obtained by each attribute of the implemented tracking system, as well as the overall grade. The overall score is calculated considering grades from “Insufficient” to “Very Good” as having scores varying linearly from 1 to 4, respectively.

Table 28 – Classification of Implemented Tracking System, considering Scenario 1, according to Grading Method defined in section 2.8

Attribute		Measured Value	Grade
Update Rate (Hz)		29.2	Satisfactory
Translational	**Accuracy (mm)**	10.3636	Insufficient
	Precision (mm)	5.8834	Satisfactory
Rotational	**Accuracy (deg)**	3.6668	Good
	Precision (deg)	2.3178	Good
Working Range (m^3)		16	Very Good
Cost (EUR)		700 + computer	Very Good
Overall Grade			**Good**

The only specification not met by this version of PTrack, running on the implemented hardware in its stand-alone version, was translational accuracy, which performed slightly worse than the acceptable range, therefore not substantially deficient. However, the multiple-camera version of PTrack meets the specification, being classified as “Satisfactory”. All other attributes were met. For costs calculation, it was considered that a computer with sufficient processing power for the system would never cost more than EUR 1,300, which is a plausible assumption.Working range and costs were assigned the rank “Very Good”, standing out as best attributes of the system. Working range can be even enlarged by using multiple-camera setup, but in exchange of exactness, which deteriorates. The overall grade of the new implemented system was “Good”, indicating that it achieved results very close to the best performance specified in section 2.8.

As mentioned in section 5.5.2.1, exactness performance can be increased by eliminating sources of imprecision. This can be done by improving calibration procedures and focal length adjustment. Further enhancements are mentioned in chapter 6. Once the improvements are implemented, the specified accuracy and precision values are very likely to be reached.

In a general manner, the new system implemented can be considered suitable for use in AR/VR environments as it provides sufficient accuracy, precision, update rate and working range, which are comparable to professional systems. The low cost feature allows broader dissemination of optical tracking systems. Using the multiple-camera tracking feature, the working range achieved allows the system to be used in new applications and scenarios, where wide-area tracking is required.

6 CONCLUSION AND FUTURE WORK

In this work, basic concepts and application scenarios for tracking systems are presented, as well as existing tracking technologies and a comparison among them. Next, the decision for optical technology and a list of commercial and research optical tracking systems are presented and, based on a comparison among those systems, specifications for a new tracking system are defined.

Afterwards, fundamental concepts for implementation of the new system are explained, followed by descriptions of the system itself and of the evaluation testbed and methods used to assess its attributes.

Within the scope of this work, commercial-off-the-shelf (COTS) cameras were analyzed, acquired and modified with infrared flash strobes and a daylight blocking filter, demanding mechanical modifications to enclose the new hardware. Labels with active and passive markers were built and tested. A hardware interface software module was written based on the camera's driver, and the image pre-processing module was entirely implemented, using calibration parameters provided by Intel's OpenCV software library. The PTrack algorithm was adapted and optimized for the new cameras. Next, all hardware and software modules were integrated. A calibration application was developed using OpenCV's routines. Calibration patterns were built. Finally, a testbed was built and used to evaluate the implemented system and compare it to other existing solutions.

Based on analysis of evaluation results, it can be said that the new system implemented in this work is an interesting contribution, consisting of a low cost system with reasonable attributes and sufficient performance for many AR/VR applications. Besides, modularity allows the system to be expanded by use of multiple one-camera tracking modules, resulting in enlarged working range.

Results of this work were partially included in the paper (SANTOS, 2006), where the author of this work contributed as co-author, accepted for the IEEE Virtual Reality Conference 2006 (VR'2006).

Motivated by the very promising results obtained, some preliminary contacts towards a commercial exploration of the developed optical tracking system have been started.

Due to the fact that this work embraces different research areas, many enhancements, which could not be implemented within this thesis, are planned and can be done in future research works.

Regarding the hardware module described in section 4.1.1, a new revision can include an increase in the number of infrared LEDs, equally distributed around the lens, similar to professional cameras as can be seen in Figure 13 and Figure 14. Additionally, a new daylight blocking filter can be used, which is better tuned to the wavelength of used infrared LEDs. For multiple-camera tracking, a synchronization module should be built in order to trigger simultaneously image acquisition and flash strobes of several cameras. This module is necessary in a stereo configuration, where simultaneous image grabbing by both cameras is essential.

The hardware interface module described in section 4.1.2 can be adapted to support several cameras by using a common camera interface, e.g. DirectShow drivers. Image pre-processing algorithms described in section 4.1.3 can be further enhanced. An ellipse fitting method is recommended to calculate markers' centers instead of the simple Binary Centroid method currently used, resulting in less jittering of marker positions. Prediction algorithms can be used to estimate new marker positions, preventing wrong calculations.

As proposed in (SHORTIS, 1994), a method to enhance robustness against ambient lighting variation is subtracting from the current scene image, taken with flash strobes illumination, an image of the scene grabbed in absence of flash strobes illumination. This can be applied during normal system operation, thus not demanding offline calibration procedures.

The large area tracking feature described in section 4.2 can be improved by adding time synchronization for reception, in central module, of UDP packets coming from different one-camera tracking modules and containing information about the same label. This improvement would avoid jittering in enhanced tracking information and is actually essential for correct merging of tracking information. This can be understood as an improvement of the sensor fusion technique as well. The use of prediction algorithms, e.g. Kalman filter, would increase the sensor fusion performance by preventing wrong pose estimation results.

Currently, the calibration procedure of multiple-camera scenarios, as described in section 4.3.2, demands connection of two related cameras to the same computer, in order to calibrate the transformation matrix between cameras. An interesting enhancement is the possibility of remote calibration in this case. This would allow the user to maintain the same connections between cameras and computers during calibration and operation of the system.

The development of a stereo tracking version of the system, using the same hardware, hardware interface and image pre-processing modules, requires only a new software module which would establish correspondence between 2D markers' positions, calculate 3D marker positions based on the stereo configuration and at last identify artifacts based on a set of markers and on the relations between them. The stereo system is likely to have higher accuracy, precision and update rate. It is planned to be implemented within the scope of a project between a company and SENAI/CETA-RS, resulting in a commercial system.

As seen in this section, this work can be further enhanced and can serve as a basis for future works and research projects.

7 REFERENCES

ABAWI, D.; BIENWALD, J.; DÖRNER, R. Accuracy in Optical Tracking with Fiducial Markers: An Accuracy Function for ARToolKit. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY, 3., Arlington, USA, Nov. 2004. **Proceedings…** Arlington, 2004. p. 260-261.

ABDEL-AZIZ, Y.I.; KARARA, H.M. Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-range Photogrammetry. In: SYMPOSIUM ON CLOSE-RANGE PHOTOGRAMMETRY, Falls Church, USA, 1971. **Proceedings…** Falls Church, 1971. p. 1-18.

ABIDI, M.A.; CHANDRA, T. A New Efficient and Direct Solution for Pose Estimation Using Quadrangular Targets: Algorithm and Evaluation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 17, n. 5, p. 534-538, May 1995.

ANSAR, A.; DANIILIDIS, K. Linear Pose Estimation from Points or Lines. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 25, n. 5, p. 578-589, May 2003.

APPEL, M.; NAVAB, N. Registration of Technical Drawings and Calibrated Images for Industrial Augmented Reality. In: IEEE WORKSHOP ON APPLICATIONS OF COMPUTER VISION, 5., Palm Springs, USA, Nov. 2000. **Proceedings…** Palm Springs, 2000. p. 48-55.

AYACHE, N.; FAVERJON, B. Efficient Registration of Stereo Images by Matching Graph Descriptions of Edge Segments. **International Journal of Computer Vision**, v. 1, n. 2, p. 107-132, Apr. 1987.

AZUMA, R. A Survey of Augmented Reality. **Presence: Teleoperators and Virtual Environments**, v. 6, n. 4, p. 355-385, Aug. 1997.

AZUMA, R. *et al.* Recent Advances in Augmented Reality. **IEEE Computer Graphics and Applications**, v. 21, n. 6, p. 34-47, Nov. 2001.

BAUER, M. *et al.* Design of a Component-Based Augmented Reality Framework. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON AUGMENTED REALITY, 2., New York, USA, Oct. 2001. **Proceedings…** New York, 2001. p. 45-54.

BENÖLKEN, P.; GRAF, H.; STORK, A. Texture-Based Flow Visualization in Augmented and Virtual Reality Environments. In: INTERNATIONAL CONFERENCE IN CENTRAL EUROPE ON COMPUTER GRAPHICS, VISUALIZATION AND COMPUTER VISION, 12., Bory, Czech Republic, Feb. 2004. **Proceedings…** Plzen, Czech Republic, 2004. p.21-24.

BISHOP, G.; WELCH, G.; ALLEN, B.D. Tracking: Beyond 15 Minutes of Thought – Course #11. In: SIGGRAPH ANNUAL CONFERENCE IN COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, Los Angeles, USA, Aug. 2001. **Proceedings…** New York, USA, 2001.

BLACH, R. *et al.* A Highly Flexible Virtual Reality System. **Future Generation Computer Systems**, v. 14, n. 3-4, p. 167-178, Aug. 1998.

BRINK, A.D.; PENDOCK, N.E. Minimum Cross-entropy Threshold Selection. **Pattern Recognition**, v. 29, n. 1, p. 179-188, Jan. 1996.

BUAES, A. G. A Survey on the Available Optical Tracking Systems for AR/VR Indoor Applications. Porto Alegre, Brazil, Feb. 2005. Available at <http://agreff.sites.uol.com.br>. Last accessed on December 16th, 2005.

CANNY, J. A Computational Approach to Edge Detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 8, n. 6, p. 679-698, Nov. 1986.

CARLSSON, C.; HAGSAND, O. DIVE – a Multi-User Virtual Reality System. In: IEEE VIRTUAL REALITY ANNUAL INTERNATIONAL SYMPOSIUM, Seattle, USA, Sep. 1993. **Proceedings…** Seattle, 2001. p. 394-400.

CASASENT, D.; PSALTIS, D. Position, Rotation, and Scale Invariant Optical Correlation. **Applied Optics**, v. 15, n. 7, p. 1,795-1,799, July 1976.

CHEN, Q.-S.; DEFRISE, M.; DECONINCK, F. Symmetric Phase-only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 12, n. 12, p. 1,156-1,168, Dec. 1994.

CHEN, X. **Design of Many-Camera Tracking Systems for Scalability and Efficient Resource Allocation.** 2002. 134 p. Ph.D Thesis - Stanford University, Stanford, USA, 2002.

CHIA, K.W.; CHEOK, A.D.; PRINCE, S.J.D. Online 6 DoF Augmented Reality Registration from Natural Features. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY, 1., Darmstadt, Germany, Sep. 2002. **Proceedings…** Darmstadt, 2002. p. 305-313.

CHUNG, J.; KIM, N.; KIM, J; PARK, C. POSTRACK: A Low Cost Real-time Motion Tracking System for VR Application. In: INTERNATIONAL CONFERENCE ON VIRTUAL SYSTEMS AND MULTIMEDIA, 7., Berkeley, USA, 2001. **Proceedings…** Berkeley, 2001. p. 383-392.

CLARKE, T.A.; COOPER, M.A.R.; FRYER, J.F. An Estimator for The Random Error in Subpixel Target Location and Its Use in The Bundle Adjustment. In: CONFERENCE ON OPTICAL 3D MEASUREMENT TECHNIQUES, 2., Zurich, Switzerland, Oct. 1993. **Proceedings…** Karlsruhe, Germany, 1993. p. 161-168.

CRUZ-NEIRA, C. *et al.* VR Juggler – An Open Source Platform for Virtual Reality Applications. In: AIAA AEROSPACE SCIENCES MEETING AND EXHIBIT, 40., Reno, USA, Jan. 2002. **Proceedings…** Reno, 2002. p. 14-17.

DHOME, M. *et al.* Determination of the Attitude of 3-D Objects from a Single Perspective View. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 11, n. 12, p. 1,265-1,278, Dec. 1989.

DING, H. *et al.* Data Fusion Algorithm Based on Fuzzy Logic. In: WORLD CONGRESS ON INTELLIGENT CONTROL AND AUTOMATION, 5., Hangzhou, China, June 2004. **Proceedings…** Hangzhou, 2004. p. 3,101-3,103.

DOCKSTADER, S.L.; TEKALP, A.M. Multiple Camera Fusion for Multi-Object Tracking. In: IEEE WORKSHOP ON MULTI-OBJECT TRACKING, Vancouver, Canada, July 2001. **Proceedings…** Vancouver, 2001. p. 95-102.

DORFMÜLLER, K. An Optical Tracking System for VR/AR-Applications. In: VIRTUAL ENVIRONMENTS CONFERENCE AND FIFTH EUROGRAPHICS WORKSHOP, Vienna, Austria, 1999. **Proceedings…** Berlin, Germany, 1999. p. 33-42.

DUDA, R.O.; HART, P.E.; **Pattern Classification and Scene Analysis**. New York, USA: Wiley, 1993. 482 p. ISBN: 0471223611.

FACELI, K.; CARVALHO, A.C.P.L.F.; REZENDE, S.O. Combining Intelligent Techniques for Sensor Fusion. In: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING, 9., Singapore, Nov. 2002. **Proceedings…** Singapore, 2002. p. 1,998-2,002.

FISCHLER, M.A.; BOLLES, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. **Communications of the ACM**, v. 24, n. 6, p. 381-395, June 1981.

FRANTZ, D.; KIRSCH, S.; WILES, A. Specifying 3D Tracking System Accuracy – One Manufacturer's View. In: BILDVERARBEITUNG FÜR DIE MEDIZIN, Berlin, Germany, Mar. 2003. **Proceedings…** Berlin, 2003. p. 234-238.

GONZALEZ, R.; WOODS, R. **Digital Image Processing**. New York, USA: Addison-Wesley, 1992. 716 p. ISBN: 0201508036.

GREENHALGH, C.; BENFORD, S. MASSIVE: A Distributed Virtual Reality System Incorporating Spatial Trading. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS, 15., Vancouver, Canada, May 1995. **Proceedings…** Vancouver, 1995. p. 27-34.

HAND, C. A Survey of 3-D Input Devices. Technical Report CS TR94/2, Department of Computer Science, De Montfort University. Leicester, England, Nov. 1993. 15 p.

HARRIS, C.G.; STEPHENS, M. A Combined Corner and Edge Detector. In: ALVEY VISION CONFERENCE, 4., Manchester, England, 1988. **Proceedings…** Manchester, 1988. p. 147-151.

HE, T.; KAUFMAN, A.E. Virtual Input Devices for 3D Systems. In: IEEE CONFERENCE ON VISUALIZATION, 4., Seattle, USA, Oct. 1993. **Proceedings…** Seattle, 1993. p. 142-148.

HERTZ, L.; SCHAFER, R.W. Multilevel Thresholding Using Edge Matching. **Computer Vision, Graphics, and Image Processing**, v. 44, n. 3, p. 279-295, Dec. 1988.

HILDEBRAND, A. *et al.* ARCHEOGUIDE: An Augmented Reality Based System for Personalized Tours in Cultural Heritage Sites. **Cultivate Interactive**, n. 1, July 2000.

HORAUD, R. *et al.* An Analytic Solution for the Perspective 4-Point Problem. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, San Diego, USA, June 1989. **Proceedings…** San Diego, 1989. p. 500-507.

HOUGH, P.V.C. Machine Analysis of Bubble Chamber Pictures. In: INTERNATIONAL CONFERENCE ON HIGH ENERGY ACCELERATORS AND INSTRUMENTATION, Geneva, Switzerland, 1959. **Proceedings…** Geneva, 1959. p. 536-554.

HUNG, Y.; YEH, P.; HARWOOD, D. Passive Ranging to Known Planar Point Sets. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, St. Louis, USA, 1985. **Proceedings…** St. Louis, 1985. p. 80-85.

IDS Imaging Development Systems GmbH. **uEye UI-121x-C**: Hardware User Manual, Version 1.11a, 2004. 41 p.

JIN, T.S.; LEE, J.M. Space and Time Sensor Fusion for Mobile Robot Navigation. In: IEEE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS, L'Aquila, Italy, July 2002. **Proceedings…** L'Aquila, 2002. p. 409-414.

JONES, D.G.; MALIK, J. Determining Three-Dimensional Shape from Orientation and Spatial Frequency Disparities. In: EUROPEAN CONFERENCE ON COMPUTER VISION, 2., Santa Margherita Ligure, Italy, May 1992. **Proceedings…** Santa Margherita Ligure, 1992. p. 661-669.

JURIE, F.; DHOME, M. A Simple and Efficient Template Matching Algorithm. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION, 8., Vancouver, Canada, July 2001. **Proceedings…** Vancouver, 2001. p. 544-549.

KALMAN, R.E. A New Approach to Linear Filtering and Prediction Problems. **Transactions of the ASME – Journal of Basic Engineering**, v. 82, series D, p. 35-45, Mar. 1960.

KANADE, T.; OKUTOMI, M. A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, Sacramento, USA, Apr. 1991. **Proceedings…** Sacramento, 1991. p. 1,088-1,095.

KATO, H.; BILLINGHURST, M. Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. In: IEEE AND ACM INTERNATIONAL WORKSHOP ON AUGMENTED REALITY, 2., San Francisco, USA, Oct. 1999. **Proceedings…** San Francisco, 1999. p. 85-94.

KATO, H. *et al.* Virtual Object Manipulation on a Table-Top AR Environment. In: IEEE AND ACM INTERNATIONAL SIMPOSIUM ON AUGMENTED REALITY, Munich, Germany, Oct. 2000. **Proceedings…** Munich, 2000. p. 111-119.

KITCHEN, L.; ROSENFELD, A. Gray-level Corner Detection. **Pattern Recognition Letters**, v. 1, n. 2, p. 95-102, 1982.

KOCH, R. *et al.* Marker-less Image-based 3D Tracking for Real-time Augmented Reality Applications. In: INTERNATIONAL WORKSHOP ON IMAGE ANALYSIS FOR MULTIMEDIA INTERACTIVE SERVICES, 6., Montreux, Switzerland, Apr. 2005. **Proceedings…** Montreux, 2005.

KOLLER, D. *et al.* Real-time Vision-Based Camera Tracking for Augmented Reality Applications. In: SYMPOSIUM ON VIRTUAL REALITY SOFTWARE AND TECHNOLOGY, Lausanne, Switzerland, Sep. 1997. **Proceedings…** Lausanne, 1997. p. 87-94.

KOVAL, V. The Competitive Sensor Fusion Algorithm for Multisensor Systems. In: INTERNATIONAL WORKSHOP ON INTELLIGENT DATA ACQUISITION AND ADVANCED COMPUTING SYSTEMS: TECHNOLOGY AND APPLICATIONS, Crimea, Ukraine, July 2001. **Proceedings…** Crimea, 2001. p. 65-68.

LI, H.; MANJUNATH, B.S.; MITRA, S.K. A Contour-based Approach to Multisensor Image Registration. **IEEE Transactions on Image Processing**, v. 4, n. 3, p. 320-334, Mar. 1995.

LI, C.H.; TAM, P.K.S. An Iterative Algorithm for Minimum Cross-entropy Thresholding. **Pattern Recognition Letters**, v. 19, n. 8, p. 771-776, June 1998.

LONGUET-HIGGINS, H.C. A Computer Algorithm for Reconstructing a Scene from Two Projections. **Nature**, v. 293, p. 133-135, Sep. 1981.

LOWE, D.G. Three-Dimensional Object Recognition from Single Two-Dimensional Images. **Artificial Intelligence**, v. 31, n. 3, p. 355-395, Mar. 1987.

LUCAS, B.D.; KANADE, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 7., Vancouver, Canada, Aug. 1981. **Proceedings…** Vancouver, 1981. p. 674-679.

LUO, R.C.; KAY, M.G. A Tutorial on Multisensor Integration and Fusion. In: ANNUAL CONFERENCE OF IEEE INDUSTRIAL ELECTRONICS SOCIETY, 16., Pacific Grove, USA, Nov. 1990. **Proceedings…** Pacific Grove, 1990. p. 707-722.

LUO, R.C.; SU, K.L. A Review of High-level Multisensor Fusion: Approaches and Applications. In: IEEE INTERNATIONAL CONFERENCE ON MULTISENSOR FUSION AND INTEGRATION FOR INTELLIGENT SYSTEMS, Taipei, Taiwan, Aug. 1999. **Proceedings…** Taipei, 1999. p. 25-31.

MALBEZIN, P.; PIEKARSKI, W.; THOMAS, B. Measuring ARToolKit Accuracy in Long Distance Tracking Experiments. In: IEEE AND ACM INTERNATIONAL AUGMENTED REALITY TOOLKIT WORKSHOP, 1., Darmstadt, Germany, Sep. 2002. **Proceedings…** Darmstadt, 2002.

MARR, D.; POGGIO, T. Cooperative Computation of Stereo Disparity. **Science**, v. 194, n. 4262, p. 283-287, Oct. 1976.

MEYER, F. Contrast Feature Extraction. In: **Quantitative Analysis of Microstructures in Material Sciences, Biology and Medicine - Special issue of Practical Metallography.** Stuttgart, Germany: Riederer-Verlag, 1977.

MORAVEC, H. Visual Mapping by a Robot Rover. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, Japan, Aug. 1979. **Proceedings…** Tokyo, 1979. p. 599-601.

MULDER, J.D.; VAN LIERE, R. The Personal Space Station: Bringing Interaction within Reach. In: VIRTUAL REALITY INTERNATIONAL CONFERENCE, 4., Laval, France, June 2002. **Proceedings…** Laval, 2002. p. 73-81.

MULDER, J.D.; JANSEN, J.; VAN RHIJN, A. An Affordable Optical Head Tracking System for Desktop VR/AR Systems. In: INTERNATIONAL WORKSHOP ON IMMERSIVE PROJECTION TECHNOLOGY, 7., and Eurographics Workshop on Virtual Environments, 9., Zurich, Switzerland, May 2003. **Proceedings…** Zurich, 2003. p. 215-223.

NAKAGAWA, Y.; ROSENFELD, A. Some Experiments on Variable Thresholding. **Pattern Recognition**, v. 11, n. 3, p. 191-204, 1979.

NICOSEVICI, T. *et al.* A Review of Sensor Fusion Techniques for Underwater Vehicle Navigation. In: OCEANS'04, MTS/IEEE TECHNO-OCEAN'04, Kobe, Japan, Nov. 2004. **Proceedings…** Kobe, 2004. p. 1,600-1,605.

NIELSEN, M.B.; KRAMP, G.; GRØNBÆK, K. Mobile Augmented Reality Support for Architects Based on Feature Tracking Techniques. In: WORKSHOP ON INTERACTIVE VISUALIZATION AND INTERACTION TECHNOLOGIES, Krakow, Poland, June 2004. **Proceedings…** Krakow, 2004.

NIEUWENHUIS, G. Lens Focus Shift Required for Reflected Ultraviolet and Infrared Photography. **Journal of Biological Photography**, v. 59, n. 1, p. 17-20, Jan. 1991.

PIEKARSKI, W.; THOMAS, B. Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments. In: IEEE AND ACM INTERNATIONAL AUGMENTED REALITY TOOLKIT WORKSHOP, 1., Darmstadt, Germany, Sep. 2002. **Proceedings…** Darmstadt, 2002.

PRESS, W.M. *et al.* **Numerical Recipes in C: The Art of Scientific Computing**. New York, USA: Cambridge University Press, 1992. 994 p. ISBN: 0521431085.

PUN, T. A New Method for Gray-level Picture Threshold Using The Entropy of The Histogram. **Signal Processing**, v. 2, n. 3, p. 223-237, 1980.

REDDY, B.S.; CHATTERJI, B.N. An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration. **IEEE Transactions on Image Processing**, v. 5, n. 8, p. 1,266-1,271, Aug. 1996.

REITMAYR, G.; SCHMALSTIEG, D. OpenTracker – An Open Software Architecture for Reconfigurable Tracking based on XML. In: IEEE VIRTUAL REALITY CONFERENCE, Yokohama, Japan, Mar. 2001. **Proceedings…** Yokohama, 2001. p. 285-286.

REKIMOTO, J.; SAITOH, M. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, Pittsburgh, USA, 1999. **Proceedings…** New York, 1999. p. 378-385.

REMAGNINO, P.; JONES, G.A. Automated Registration of Surveillance Data for Multi-Camera Fusion. In: INTERNATIONAL CONFERENCE ON INFORMATION FUSION, 5., Annapolis, USA, July 2002. **Proceedings…** Annapolis, 2002. p. 1,190-1,197.

RIBO, M.; PINZ, A.; FUHRMANN, A.L. A New Optical Tracking System for Virtual and Augmented Reality Applications. In: IEEE INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE, 18., Budapest, Hungary, May 2001. **Proceedings…** Budapest, 2001. p.1,932-1,936.

RIBO, M. State of the Art Report on Optical Tracking. Technical Report VRVis 2001-25, Technical University of Vienna. Vienna, Austria, Sep. 2001. 13 p.

ROBERTS, L.G. Machine Perception of Three Dimensional Solids. In: **Optical and Electro-optical Information Processing**. Cambridge, USA: MIT Press, 1966. p.159-197.

SANTOS, P. *et al.* 3D Interactive Augmented Reality in Early Stages of Product Design. In: INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION, 10., Crete, Greece, June 2003. **Proceedings…** Mahwah, USA, 2003. p. 1,203-1,207.

SANTOS, P. **A 2D to 3D Geometric Interpolation Algorithm for Marker-based Single-camera Tracking**. 2005. 168 p. Master Thesis - Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, 2005.

SANTOS, P. *et al.* PTrack: Introducing a Novel Iterative Geometric Pose Estimation for a Marker-based Single Camera Tracking System. In: IEEE VIRTUAL REALITY CONFERENCE, Alexandria, USA, Mar. 2006. **Proceedings…** Alexandria, USA, 2006. p. 143-150.

SAUVOLA, J.; PIETIKÄINEN, M. Adaptive Document Image Binarization. **Pattern Recognition**, v. 33, p. 225-236, 2000.

SCHARSTEIN, D.; SZELISKI, R. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. **International Journal of Computer Vision**, v. 47, n. 1-3, p. 7-42, Apr. 2002.

SCHMALSTIEG, D. *et al.* The Studierstube Augmented Reality Project. **Presence: Teleoperators and Virtual Environments**, v. 11, n. 1, p. 33-54, Feb. 2002.

SCHWALD, B.; SEIBERT, H.; WELLER, T. A Flexible Tracking Concept Applied to Medical Scenarios Using an AR Window. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY, Darmstadt, Germany, Sep. 2002. **Proceedings…** Darmstadt, 2002. p. 261-262.

SEZGIN, M.; SANKUR, B. Survey Over Image Thresholding Techniques and Quantitative Performance Evaluation. **Journal of Electronic Imaging**, v. 13, n. 1, p. 146-165, Jan. 2004.

SHAH, M. **Fundamentals of Computer Vision**. Orlando, USA: Computer Science Department - University of Central Florida, 1997. 132 p. Available at <http://www.cs.ucf.edu/courses/cap6411/book.ps>. Last accessed on December 16th, 2005.

SHAW, C. *et al.* Decoupled Simulation in Virtual Reality with the MR Toolkit. **ACM Transactions on Information Systems**, v. 11, n. 3, p. 287-317, July 1993.

SHI, J.; TOMASI, C. Good Features to Track. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, Seattle, USA, June 1994. **Proceedings…** Seattle, 1994. p. 593-600.

SHORTIS, M.R.; CLARKE, T.A.; SHORT, T. A Comparison of Some Techniques for The Subpixel Location of Discrete Target Images. In: SPIE VIDEOMETRICS III, Boston, USA, Nov. 1994. **Proceedings…** Boston, 1994. p. 239-249.

SHUM, H.-Y.; SZELISKI, R. Construction and Refinement of Panoramic Mosaics with Global and Local Alignment. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION, 6., Mumbai, India, Jan. 1998. **Proceedings…** Washington, USA, 1998. p. 953-958.

SIMON, G.; FITZGIBBON, A.W.; ZISSERMAN, A. Marker-less Tracking Using Planar Structures in the Scene. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON AUGMENTED REALITY, Munich, Germany, Oct. 2000. **Proceedings…** Munich, 2000. p. 120-128.

SIMON, G.; BERGER, M.-O. Reconstructing While Registering: A Novel Approach for Marker-less Augmented Reality. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY, Darmstadt, Germany, Sep. 2002. **Proceedings…** Darmstadt, 2002. p. 285-294.

STRICKER, D.; KLINKER, G.; REINERS, D. A Fast and Robust Line-based Optical Tracker for Augmented Reality Applications. In: INTERNATIONAL WORKSHOP ON AUGMENTED REALITY, 1., San Francisco, USA, Nov. 1998. **Proceedings…** San Francisco, 1998. p. 31-46.

STRICKER, D. Tracking with Reference Images: a Real-time and Marker-less Tracking Solution for Outdoor Augmented Reality Applications. In: CONFERENCE ON VIRTUAL REALITY, ARCHEOLOGY AND CULTURAL HERITAGE, Glyfada, Greece, Nov. 2001. **Proceedings…** New York, USA, 2001. p. 77-82.

TAO, Y.; HU, H. Building a Visual Tracking System for Home-based Rehabilitation. In: ANNUAL CONFERENCE OF THE CHINESE AUTOMATION AND COMPUTING SOCIETY IN THE UK, 9., Luton, England, Sep. 2003. **Proceedings…** Luton, 2003. p. 443-448.

THOMOPOULOS, S.C.A. Sensor Selectivity and Intelligent Data Fusion. In: IEEE INTERNATIONAL CONFERENCE ON MULTISENSOR FUSION AND INTEGRATION FOR INTELLIGENT SYSTEMS, Las Vegas, USA, Oct. 1994. **Proceedings…** Las Vegas, 1994. p. 529-537.

TOMASI, C.; KANADE, T. Shape and Motion from Image Streams under Orthography: a Factorization Method. **International Journal of Computer Vision**, v. 9, n. 2, p. 137-154, 1992.

TRAMBEREND, H. Avocado: A Distributed Virtual Reality Framework. In: IEEE VIRTUAL REALITY CONFERENCE, Houston, USA, Mar. 1999. **Proceedings…** Houston, 1999. p. 14-21.

TRUCCO, E.; VERRI, A. **Introductory Techniques for 3-D Computer Vision**. New Jersey, USA: Prentice Hall, 1998. 343 p. ISBN: 0132611082.

TSAI, R.Y. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, Miami Beach, USA, June 1986. **Proceedings…** Miami Beach, 1986. p. 364-374.

TSAI, R.Y. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses. **IEEE Journal of Robotics and Automation**, v. RA-3, n. 4, p. 323-344, Aug. 1987.

VAN LIERE, R.; MULDER, J.D. Optical Tracking Using Projective Invariant Marker Pattern Properties. In: IEEE VIRTUAL REALITY CONFERENCE, Los Angeles, USA, Mar. 2003. **Proceedings…** Los Angeles, 2003. p. 191-198.

WANG, X.; GAO, J.; WANG, L. A Survey of Subpixel Localization for Image Measurement. In: INTERNATIONAL CONFERENCE ON INFORMATION ACQUISITION, Hefei, China, June 2004. **Proceedings…** Hefei, 2004. p. 398-401.

WELCH, G. *et al.* High-Performance Wide-Area Optical Tracking – The HiBall Tracking System. **Presence: Teleoperators and Virtual Environments**, v. 10, n. 1, p. 1-21, Feb. 2001.

WEST, G.A.; CLARKE, T.A. A Survey and Examination of Subpixel Measurement Techniques. In: CLOSE-RANGE PHOTOGRAMMETRY MEETS MACHINE VISION (SPIE), Zurich, Switzerland, Sep. 1990. **Proceedings…** Bellingham, USA, 1990. p. 456-463.

WONG, K.W. Mathematical Formulation and Digital Analysis in Close-range Photogrammetry. **Photogrammetric Engineering Remote Sensing**, v. 41, n. 11, p. 1,355-1,373, 1975.

YEN, J.; CHANG, F.; CHANG, S. A New Criterion for Automatic Multilevel Thresholding. **IEEE Transactions on Image Processing**, v. 4, n. 3, p. 370-378, Mar. 1995.

YONEMOTO, S. *et al.* A Real-time Motion Capture System with Multiple Camera Fusion. In: INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND PROCESSING, 10., Venice, Italy, Sep. 1999. **Proceedings…** Venice, 1999. p. 600-605.

YUAN, J.S.-C. A General Photogrammetric Method for Determining Object Position and Orientation. **IEEE Transactions on Robotics and Automation**, v. 5, n. 2, p. 129-142, Apr. 1989.

ZHANG, Z. Flexible Camera Calibration by Viewing A Plane from Unknown Orientations. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION, 7., Kerkyra, Greece, Sep. 1999. **Proceedings…** Los Alamitos, USA, 1999. p. 666-673.

ZHANG, Z. A Flexible New Technique for Camera Calibration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 22, n. 11, p. 1,330-1,334, Nov. 2000.

ZHANG, X.; FRONZ, S.; NAVAB, N. Visual Marker Detection and Decoding in AR Systems: A Comparative Study. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY, Darmstadt, Germany, Sep. 2002. **Proceedings…** Darmstadt, 2002. p. 97-106.

ZHOU, H.; HU, H. A Survey – Human Movement Tracking and Stroke Rehabilitation. Technical Report: CSM-420, Department of Computer Sciences, University of Essex. Essex, England, Dec. 2004. 32 p.

ZOGHLAMI, I.; FAUGERAS, O.; DERICHE, R. Using Geometric Corners to Build a 2D Mosaic from a Set of Images. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 16., San Juan, Puerto Rico, June 1997. **Proceedings…** San Juan, 1997. p. 420-425.

APPENDIX A – TESTBED'S ELECTRICAL SCHEMATICS

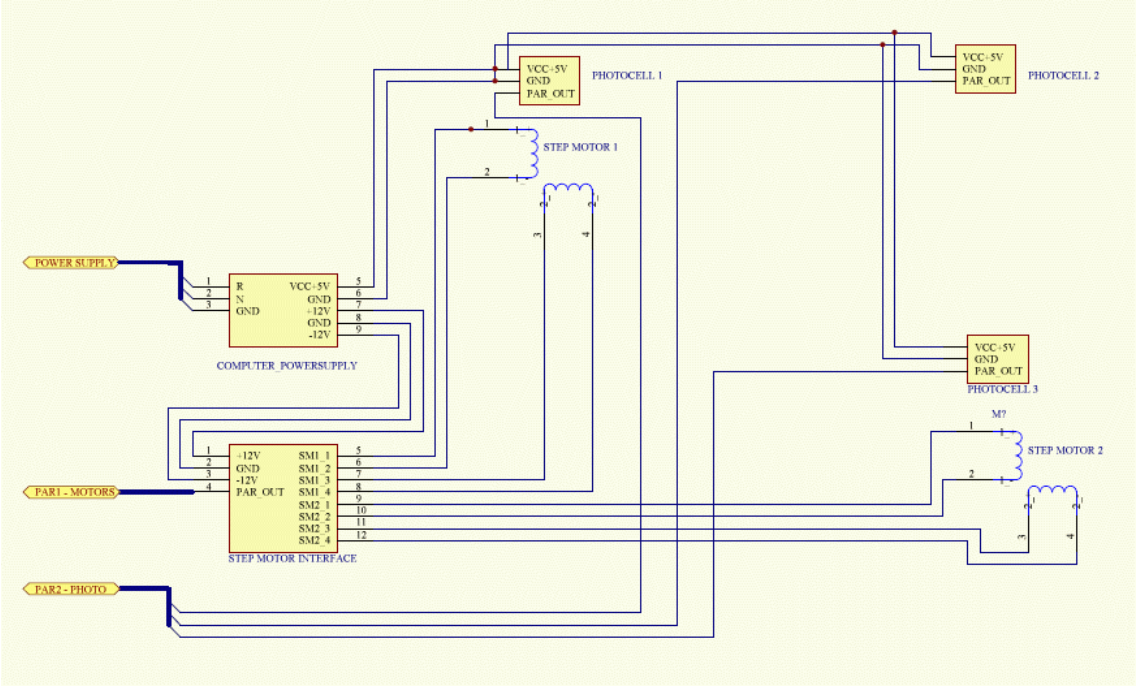

Figure 113 – Electrical Schematic of Evaluation Testbed.

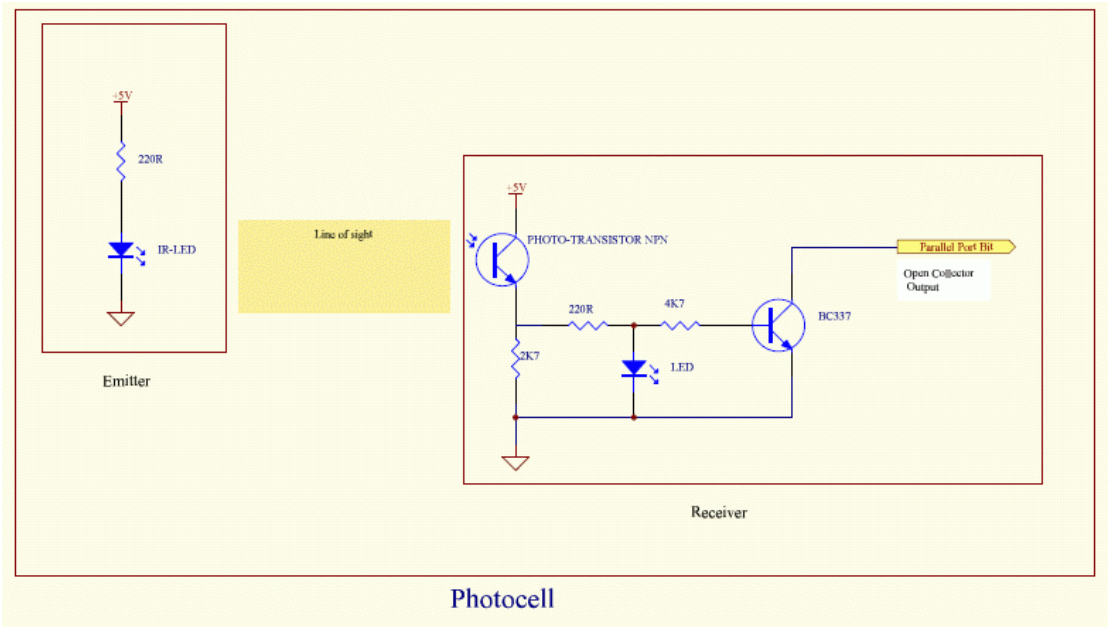

Figure 114 – Electrical Schematic of Photocells in Evaluation Testbed.