UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

JULIANO MORATO FRANZ

STC: Smart Travelers Camera

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Engeneering

Advisor: Profª. Dra. Luciana Nedel

Porto Alegre
December 18th 2014

"Do not judge me by my successes. Judge me by how many times I fell down and got back up again." - Nelson Mandela

ACKNOWLEDGMENTS

Firstly, I would like to thank my advisor Luciana Nedel, from the Federal University of Rio Grande do Sul, for all the support and guidance during all my undergraduate studies. Also, my sincere thanks for all the professors and students of the GVI Lab.

In the personal department, I would like thank my family, friends and classmates for the support during stressful times. To the last ones, a warm thank you for those long hours of study and projects. I am also thankful for the support of my girlfriend Amanda, who found a way to keep me in balance.

ABSTRACT

When people travel, they often encounter problems and difficulties along the way. A very common issue is to ask for the help of a stranger to take a photograph on a touristic spot. Some of the results of this short collaboration are not as previously imagined. This work introduces a new camera application for smartphones that aims to help tourists to take good pictures during travels. The proposed system consists of an application capable of aiding a third-party to photograph the camera owner. It does so by guiding the photographer to position the camera in the right place that was once imagined by the owner. The application current targets are Android devices, and it relies on the phone inertial sensors and computer vision algorithms. Two different different studies were conducted with a total of 60 participants. The results shown that most of the participants liked the photograph taken with the application and would use the application during their travels.

Keywords: Photography. android. image processing. openCV. interial sensors.

RESUMO

Quando as pessoas viajam, muitas vezes encontram algumas dificuldades ao longo do caminho. Um problema muito comum é o de fazer fotografias em algum ponto turístico com a ajuda de um estranho uma vez que grande parte dos resultados desta curta colaboração não saem como o esperado. Neste trabalho, é introduzido um novo aplicativo de câmera fotográfica que tem como objetivo ajudar turistas durante as suas viagens. O sistema proposto consiste em uma aplicação capaz de ajudar um terceiro a fotografar o proprietário da câmera em pouco tempo. O aplicativo opera orientando o fotógrafo a posicionar a câmera no lugar previamente definido e como imaginado pelo proprietário. A aplicação foi desenvolvida para dispositivos Android e faz usos algoritmos de medição inercial e de visão computacional.

Palavras-chave: fotografia, processamento de imagem, openCV, sensores inerciais.

LIST OF ABBREVIATIONS AND ACRONYMS

STC	Smart Travelers Camera
UI	User Interface
DOF	Degrees of Freedom
PnS	Point-and-Shoot
IMU	Inertial Measurement Unit

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

We live in a digital era where most of us carry one or more cameras at all times and because of that we intrinsically create tons of images. But because most people did not study the basics of image composition they are not able to produce excellent images.

We carry this issue with us during our travels. Even if we know how to proper compose a photograph, sometimes we have to rely on another person. And their compositions problems end up affecting our images.

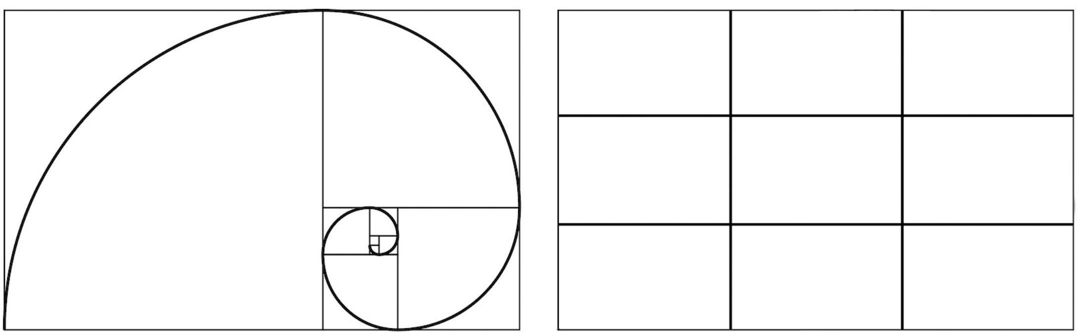
1.1 Motivation

Image composing is one of the most fundamental techniques used in any type of art that creates imagery. In photography, people are firstly introduced to two simple (on the surface at least) concepts: the golden ratio, that assumes that the image composition should follow the the golden ratio, and the rule of thirds, that is a simplification of the golden ratio an assumes that the every image have four major points of interest (Figure 1.1). More advanced photographers even study techniques from famous painters such as DaVinci and Rembrandt.

Travelers often need assistance from strangers to aid them in taking pictures of themselves. The main issue is that the traveler have no knowledge of what are the abilities of that stranger in particular. Again, the lack of proper know how may occur and the final result shall not capture the real intent of the traveler.

Eliminating the easy to correct problems, *e.g. out of focus, thumb in front of the lens* one could assume that the real issue with travel photography is composition. Maybe we could assume that in the selfie era people would search for education on how to take better pictures. However in most cases that is not true mainly because people can now, with digital cameras, see if the result is acceptable just after the button is pressed. this is exactly the opposite that used to occur with film cameras where the photographer would take more time thinking before

Figure 1.1: Two basic composing techniques: the Golden Ratio and the Rule of Thirds



Source: Created by the author.

shotting the photograph, because he would only see if the result was good weeks after.

In short, there are several reasons why bad travel photographs exists. Most people are in a hurry and when someone asks them for this favor he or she often try to do it as quickly as possible. Another class of strangers is the ones that have time, but do not know how to frame a photo correctly. Rare are the ones that know what they are doing, but that does not mean that the result will be satisfying because people just think different. Even the lucky ones that find someone with time and skill to help them, might not be able to explain what he or she wants because of language barriers.

1.2 Goals

The work presented here proposes an attempt to solve the problem with travel photography regarding composition. It assumes that by removing the responsibility to think on how the photo should be framed away from the stranger, the final image could be significantly improved. It is so because the only person who has real interest in the final photograph is the one being photographed therefore the responsibility of framing the image should be the theirs.

1.3 Contribution

In this work, we introduce a novel camera application that aims to minimize human error when shooting photographs for someone. It was created as camera a application for Android that relies on computer vision algorithms and the current orientation of the cell phone and was baptized as Smart Travelers Camera (STC). Smart because it does not rely on the knowledge of the person that is using it. The software assumes that whoever is operating the camera barely has any photographic experience, but have sufficient intimacy with modern cellphones and is capable of following simple on-screen information.

With the use of STC the responsibility of imagining how the final image should be and how it should be framed is entirely of the person that have the most interest in it. The application relieves the stranger from the need of thinking about the final image as he only needs to point the lens forward. Of course that the composition problem is not solved but it is assumed that whoever owns the application will have at least a minor level of knowledge on how to frame a picture correctly.

1.4 Structure of this work

The rest of this work is organized as follows: Related Work of image processing and orientation sensing applied to photography are presented in Chapter 2. The STC concept design and implementation are presented in Chapter 3. The design and sample characterization of both studies are shown in Chapter 4. Chapter 5 exposes the final results for both studies and finally

Chapter 6 ends this work showing its main contributions and presenting suggestions for future work.

2 RELATED WORK

This kind of application is new in the literature mainly because digital cameras are closed platforms, thus making it hard to develop and test new applications in the field. This problem was broadly discussed by Levoy (LEVOY, 2010). He talks about the current industry situation and, also, shows some successful prototypes that can be used as an alternative to off the shelf cameras. Adams et al.(ADAMS et al., 2010) developed the Frankencamera, shown in Figure 2.1 that shows up as a more robust hardware and software solution for computational photography development that uses off the shelf components. This camera is an alternative to commercial cameras because both hardware and software are opensource. However, the Frankencamera developments is currently halted.

Figure 2.1: The Frankencamera (Camera 2.0) developed in Stanford Universisty

Source: (ADAMS et al., 2010)

Some of the techniques that are going to be used here are not new and were also used in retro-photography algorithms in the past. Bae et al. (BAE; AGARWALA; DURAND, 2010) proposed a solution to recreate photographs from the past using a reference photogrpah and image processing. In their work, they shown that it is not possible to recreate a photograph only relying on naive composition, i.e. trying to recompose an image just by looking at the old one, there is a need for computational assistance. In Figure 2.2 we can see one of their test results image. Their approach, however, needed a time costly calibration phase and also needed the use of a computer attached to the camera to handle the image processing. That whole process limits the user freedom and requires more time to recreate the photograph. Their solution would need around thirty minutes to recreate the photograph, which in the travel photography scenario is not acceptable.

2.1 Image processing

Feature detectors and extractors are used since the beginning of computer vision. Detectors are often categorized in a few different types such as edge, corner and blob. Each one of the detectors has their own subset of applications. Once the detected subset is created it is possible to try to extract information of them with a feature description algorithm.

SIFT (LOWE, 1999) is an algorithm to detect and describe local features in images, and was largely used mainly because it has the capability of finding keypoints that are invariant to location, scale and rotation. Mikolajczyk et al. (MIKOLAJCZYK; SCHMID, 2005) tested SIFT against other methods of descriptors including the author's method and SIFT outperformed most of the algorithms tested.

Another common feature descriptor is the Speeded Up Robust Features (SURF) (BAY et al., 2008), which is partially based on SIFT. SURF is several times faster and more accurate then SIFT because it uses a type of blob detector as basis.

However, both SIFT and SURF are not intended for use in mobile applications because of their computational requirement as described by Rublee et al. (RUBLEE et al., 2011) and Yang et al. (YANG; CHENG, 2012).

In this paper, the proposed application make use of another feature extractor: Oriented FAST and Rotated BRIEF (ORB) introduced by Rublee et al.(RUBLEE et al., 2011) as an alternative for both SIFT and SURF: ORB is a great alternative to SIFT and SURF in computation cost, matching performance and mainly because it is free of nasty patents. ORB also is more friendly on mobile devices, thus using less energy and saving battery life.

2.2 Orientation sensing

Inertial measurement units (IMU) today are tiny and almost every wearable electronic or mobile device such as cellphones and cameras have them installed. They often have 6 degrees of freedom, but today most of the devices are designed with 9 DOF units. Figure 2.3 shows a 6 DOF IMU breakout board. Those units usually have a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer. Woodman (WOODMAN; WOODMAN; WOODMAN, 2007) goes through the different types of devices and talks about how they can act together to track movements.

This combined act is known as sensor fusion and is widely used in mobile devices. Sensor fusion is mostly used to track head movements in modern head-mounted displays as used by Reus et al. (REUS et al., 2013) and in some cases is used to help estimating camera position in the real world.

In the proposed application sensor fusion is used to estimate the current camera orientation relative to a reference frame. Similar work was described by Hol et al. (HOL et al., 2006)

Figure 2.2: Example of an image from the past reconstructed with Bae's technique and by a naive approach

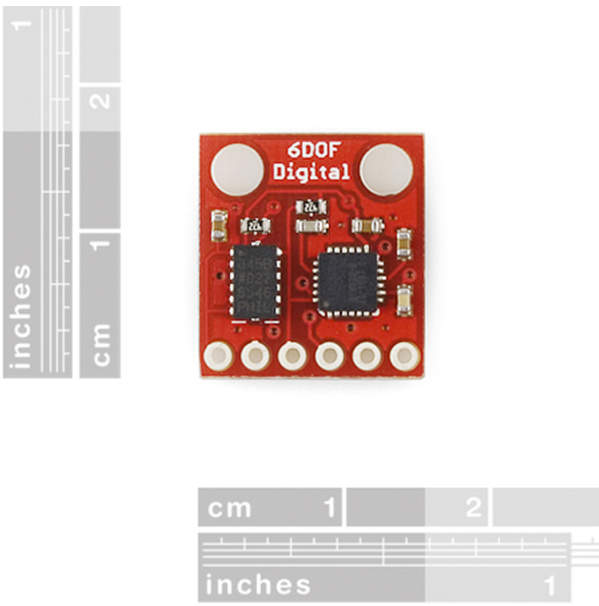
(a) The reference photograph from the past

(b) A photograph taken using Bae's algorithm

(c) A photograph taken with a naive approach.

Source: (BAE; AGARWALA; DURAND, 2010)

Figure 2.3: 6 DOF IMU breakout board containing two ICs, one Accelerometer and one Gyroscope



Source: Sparkfun SKU: SEN-1021

3 SMART TOURIST CAMERA

Here it is introduced the STC, a camera application that has the propose of aiding tourists with photos abroad. STC aims to solve the problem that occurs when someone needs the assistance of a third party to take photographs for them. This situation is very common when the tourist wants to be part of the photograph.

The software is designed with three base principles: it has to complete its task fast, it has to be easy to use and it has to have an easy to learn interface. All those ideas are used to minimize any issues with the stranger.

3.1 System Design

The STC main objective is to avoid the need for the strange to think on how the photograph should be framed. By doing so the software take most of the responsibility away from the stranger.

STC consists of an application that can be deployed on any phone or even embedded in modern compact cameras. The minimum requirement is that they must have an IMU with at least 6 DOF and a processor capable of running computer vision algorithms. In the scope of this work the software was implemented on an Android mobile phone.

The software works by using a reference photo and a frame of reference that was set by the camera owner. Once the owner sets the reference frame, the phone can be delivered to anyone in the crowd and all this person has to do is to follow the on-screen instructions. By doing this, the concept of a disposable camera, that was pretty common in the film days, is brought back, because the person that is operating will use it only once thus requiring a good interface.

The application works as follows: first the tourist goes to where he wants the camera to be and take a reference photograph from the scene. Then, he delivers the camera to the third-party, here called as a user. The user then points the camera forward and follows the on-screen instruction. Once the camera detects that it is close enough to the reference photo it automatically takes a series of photographs.

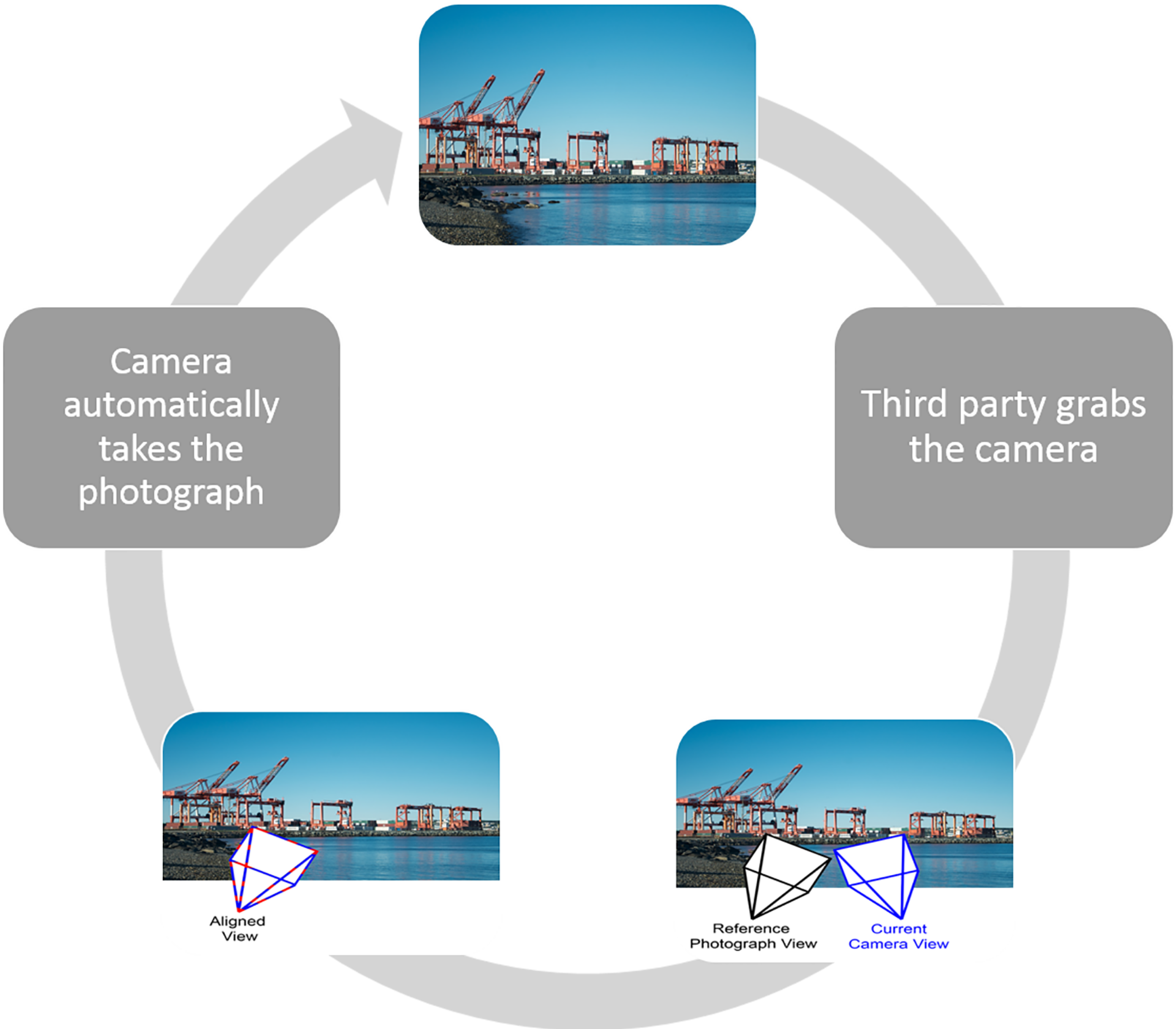
Figure 3.1 illustrates the overview of the application environment.

3.1.1 Pose estimation

In order to achieve the desired result, it is necessary to know where the camera is pointing at and compare it to where it should be. Therefore, there is a need to create a link between the camera's inertial measure units (IMU) and its current view of the real world. This technique has already been used in several applications involving mobile devices and virtual reality environments such as described by Hol (HOL et al., 2006).

The bound between camera and the IMU is done in two different steps to ease on the user in-

Figure 3.1: STC overview



Source: Created by the author.

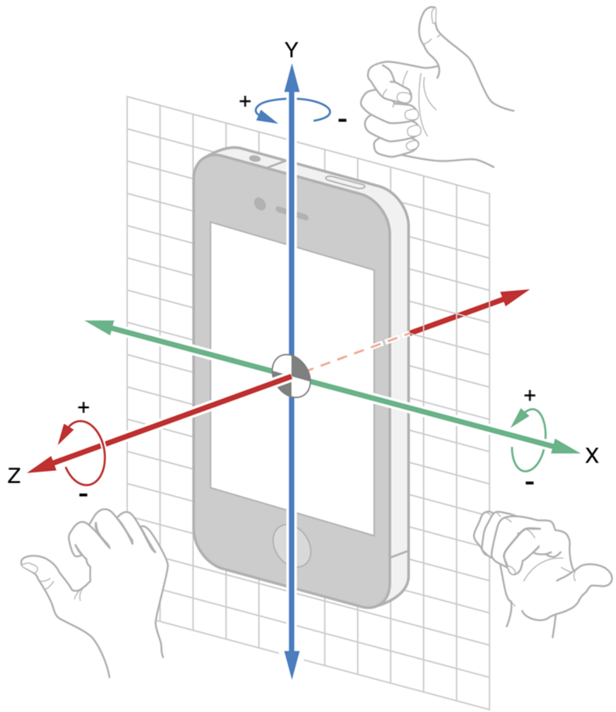
teraction. First the user will be guided to correct the rotation of the camera by checking the IMU data and comparing it against the reference frame. Once the proper orientation is achieved, the user is then guided to fix the camera translation. The camera calculates the necessary translation transformation comparing the actual camera view and the one from the reference photograph.

3.1.1.1 Camera Rotation

The first step on reconstructing the reference photo is to correct the camera rotation. That is changing the camera's roll, pitch and yaw (Figure 3.2) to set it according to the reference model. At every frame, the camera gets a quaternion that represents the camera orientation. With that, it will render an image of the camera state on the screen. All the user have to do is to align that representation with a ghost camera that shows where the real camera needs to be.

With a sensor fusion technique, that uses all of the available IMU sensors (accelerometer, gyroscope and magnetometer), it is possible to obtain the quaternion data for every frame. Quaternion are preferred over Euler angles because they are faster to compute and do not suffer from ambiguities and gimbal locks.

Figure 3.2: Device's rotation axes



Source: Apple IOS SDK Manual.

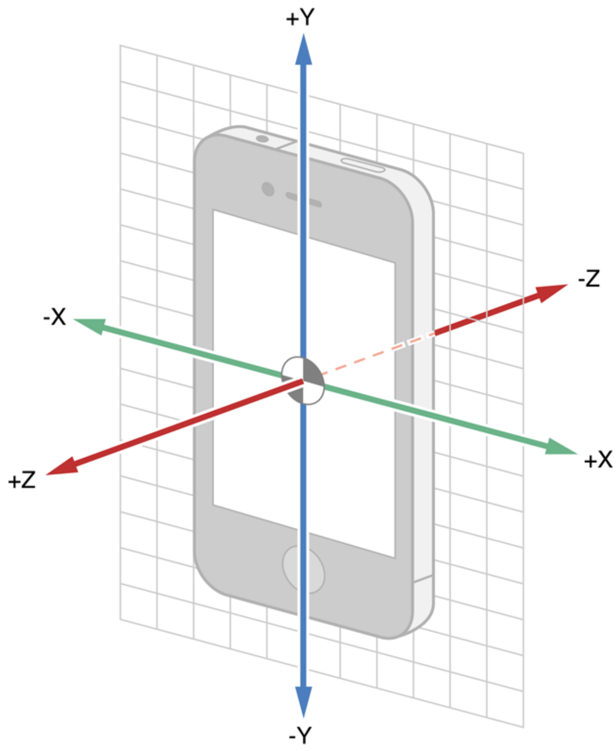
3.1.1.2 Camera translation

Once the rotation axes are correct, the application can guide the user to fix the translation axis as seen in Figure 3.3. To calculate the amount of correction necessary for the translation axes the camera will, at every frame, extract keypoints from the image and match them upon the reference photo ones. With that matching it is possible to calculate the translation difference between both images.

3.2 User Interface

The fast interaction with the application is only possible if the UI is simple enough to be used for someone with no training on it. This work differs from Bae's work (BAE; AGARWALA; DURAND, 2010) specially on the interface. His work had a complex interface that required

Figure 3.3: Device’s translation axes



Source: Apple IOS SDK Manual.

time to use (approximatively 30 minutes) where our has a simple one that requires no more then 30 seconds.

The application has some similarities to the disposable cameras of the film days. That is so because once the third-party takes the photo he or she will probably not use it again. From his point of view, the camera is a single use device. Because of that, the application interface has a tremendous weight on the overall system especially because there is no training phase for the user.

The user interface was designed with the goal of being as simple as possible. It has no buttons, and there are only two kinds of instructions for the user: rotate and translate. The interface also tries to minimize external distraction by fading the current camera image. That is expected to work because the user opinion about the final result is not to be taken into consideration.

3.2.1 Shake avoidance

Sometimes the image produced by cameras, especially in low light situations, suffer from blur resultant from camera shaking. This problem can also affect our algorithm because the photographer is always moving the camera towards the correct placement. In order to reduce or avoid camera blur, once the camera is in the correct position the application takes a burst of photographs instead of a single one.

The intent to use the burst of images is only as an attempt to avoid image blur. However, blur can still occur because the user is continuously moving the camera. In future work, it would be interesting to add a post-processing stage that tries to minimize or remove image blur. Because STC already has the IMU data at the time of the photograph, it could save this information and use it in the deblurring stage as used by Joshi et al. (JOSHI et al., 2010).

An example of the de-blurring algorithm (JOSHI et al., 2010) is shown in Figure 3.4 where on the left the blurred image is shown and on the right, the corrected one.

3.3 System Implementation

The application was implemented in both Java an C++ using the JNI as a gate between both programming languages. OpenCV was used as the image processing core. The target device was a Google Nexus 5 running Android KitKat, however it could be ported to the IOS platform.

Figure 3.5 shows the system's block top-down architecture starting from the Android SDK.

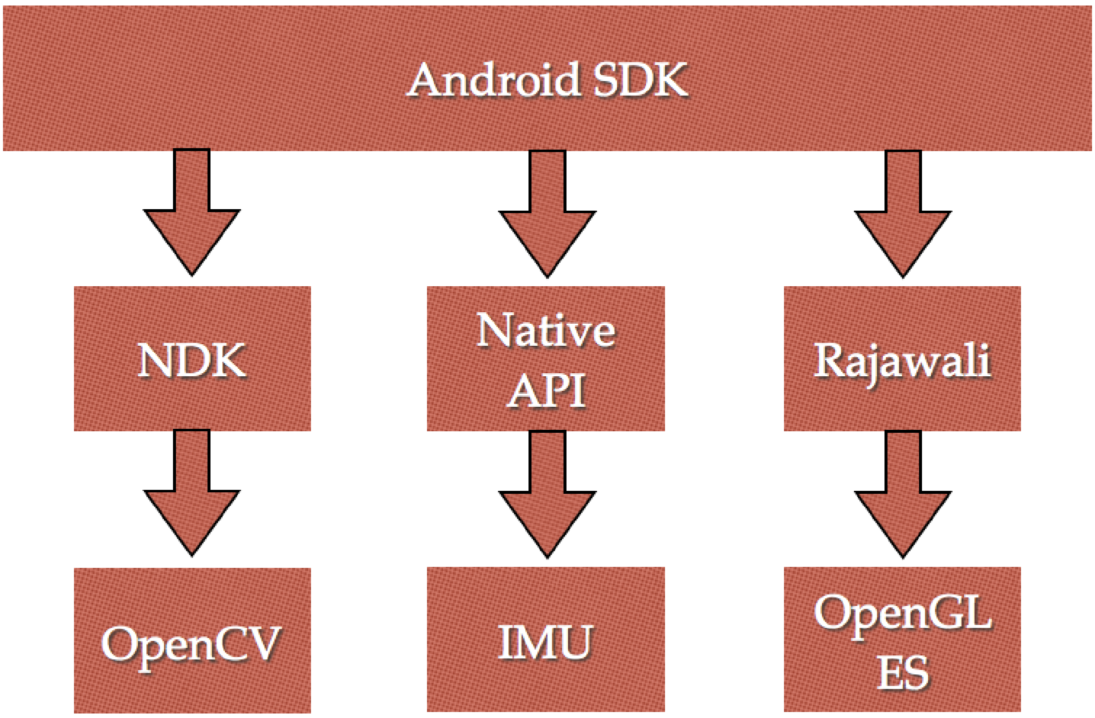
3.3.1 Nexus 5

The Google device was chosen as the target device because at the time of the implementation it was the current flagship mobile device available in the market. Table 3.1 shows the phone specifications

Figure 3.4: Example of image de-bluring using IMU based post-processing

Source: (JOSHI et al., 2010)

Figure 3.5: The architecture of STC



Source: Created by the author.

Table 3.1: Nexus 5's main specifications

	Nexus 5
Display	4.95 inches True HD IPS+ with 1080x1920 pixels
Main Camera	8 MP, 3264 x 2448 pixels, autofocus, optical image stabilization
OS	Android OS 4.4 (KitKat)
Chipset	Qualcomm MSM8974 Snapdragon 800
CPU	Quad-core 2.3 GHz Krait 400
GPU	Adreno 330
Sensors	Accelerometer, Gyroscope, Proximity, Compass, Barometer

Source: Created by the author.

3.3.2 Rotation Sensing

To estimate the current orientation of the camera relative to the real world it was used a fusion between all of the phone IMU sensors.

A wrapper around the Android Sensor Manager API was created to implement the Sensor Event Listener calls. The API provides a special virtual sensor called Rotation Vector. This sensor is a combination of all available motion sensors present on the phone. It represents the orientation of the device as a combination of an angle and an axis, in which the device has rotated through an angle θ around an axis (x, y, or z). The refresh value of the sensors was set to the default value of $1000us$.

With the information extracted form the Rotation Vector it was possible to create a function that compares if the rotation from the reference is equal to the one at the current frame. By doing so the first problem of the pose estimation was solved.

3.3.3 Computer Vision

To solve the camera position among the screen place the software uses image processing algorithms with OpenCV. At the moment that the camera takes the reference photograph, the library extracts keypoints, using the ORB algorithm, from the image, as seen in Figure 3.6. After the keypoints are extracted they are saved with the reference frame from the rotation sensors.

The chosen feature extractor and descriptor, ORB, is currently the state of the art for this type of task on mobile devices. A side from having a good resistance to noise on the images it is also processor and consequently battery friendly. ORB also runs faster then traditional extractors on mobile devices. Table 3.2 extracted from Yang's work (YANG; CHENG, 2012) shows how faster is ORB against SURF.

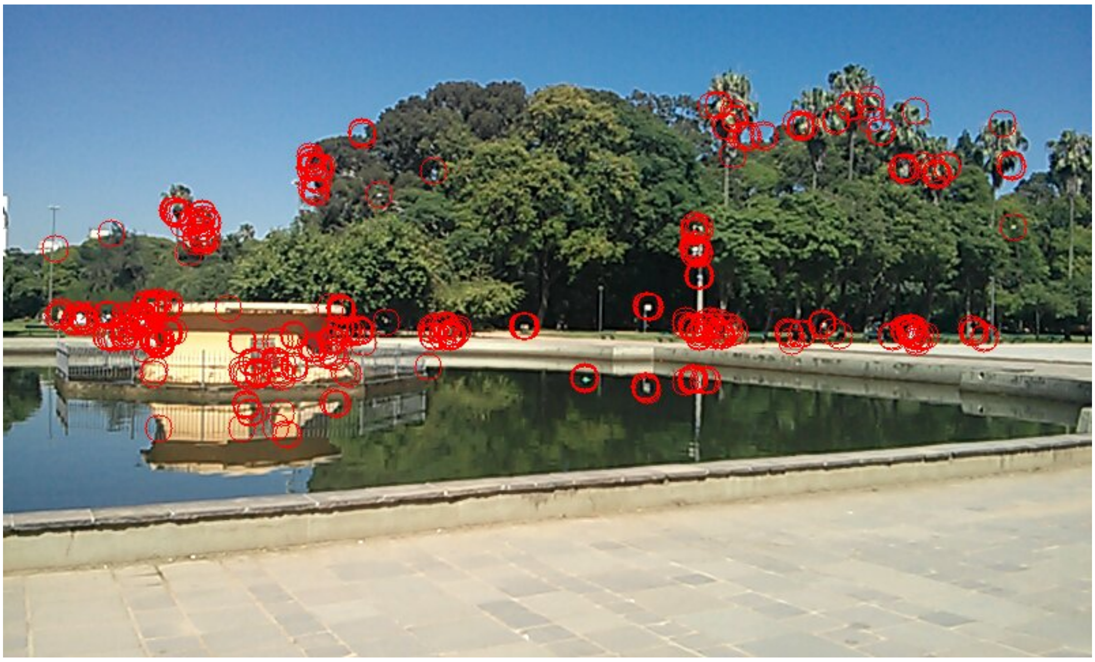
When the stranger begins its interaction with the application the software does exactly the

Table 3.2: Table comparing ORB and SURF times

	Phone (ms)	PC (ms)
ORB	170	40
SURF	2156	143

Source: (YANG; CHENG, 2012)

Figure 3.6: Image showing the detected keypoints from the reference photograph



Source: Created by the author.

same keypoint detection at each camera frame. With the new keypoints extracted the application tries to match them with the ones from the reference photograph that were saved.

They keypoints are matched using a brute force algorithm that takes the descriptor of one feature in first image and matches with all other features in second image. The match with the smallest distance is then chosen.

Once the keypoints are matched, it is possible to calculate a distance vector from the reference ones to the actual frame ones. This vector is created by taking an average of all vectors between each keypoint matches.

With that vector, the UI then renders the information for the user to where he or she should move the camera. Figure 3.7 shows each individual vector for the keypoint matches currently detected.

At the moment when the magnitude of distance vector is smaller then a defined threshold the user is asked to hold the camera still for a couple seconds and the camera is triggered. The

Figure 3.7: The matched keypoints. Red are the ones from the reference photograph and green, the ones from the current frame.

Source: Created by the author.

application flow is shown in Figure 3.8

Currently the minimal amount of matched keypoints for the application to work is set to one. Nonetheless, with a small number of matches, the application can oddly behave leading to a greater time span before the automatic photograph. Further studies should access a minimum threshold however we did not had problems during the user study regarding this matter.

3.3.4 User Interface

The user interface is one of the most important elements of the application because of the need of a quick understanding on how the software works. It is divided in two different screens: the first one where the camera owner sets the reference photograph and the one where the stranger interacts with the application.

Both screens were kept as simple as possible. The first has only one button and the current camera view (Figure 3.9). To set the reference image, the owner has only to point the lens to the right place and hit the button. This interaction is equal to a PnS.

The second screen (Figure 3.10) is more crucial. It has to be simple enough for someone to understand it in a few seconds. The first important component is that it has no button that the stranger needs to click.

Another factor is that it tries to minimize the world influence on the stranger. It does so by dimming most of the camera view leaving only the center of the image clear. By doing that it guides the user to focus only on the information shown in the center.

The last element is the one that the stranger interacts with. It's a centralized ghost camera that does not move and an arrow pointing to a more solid camera. What the user have to do is move the camera towards the direction that the arrow shows, placing the solid camera in the middle of the screen.

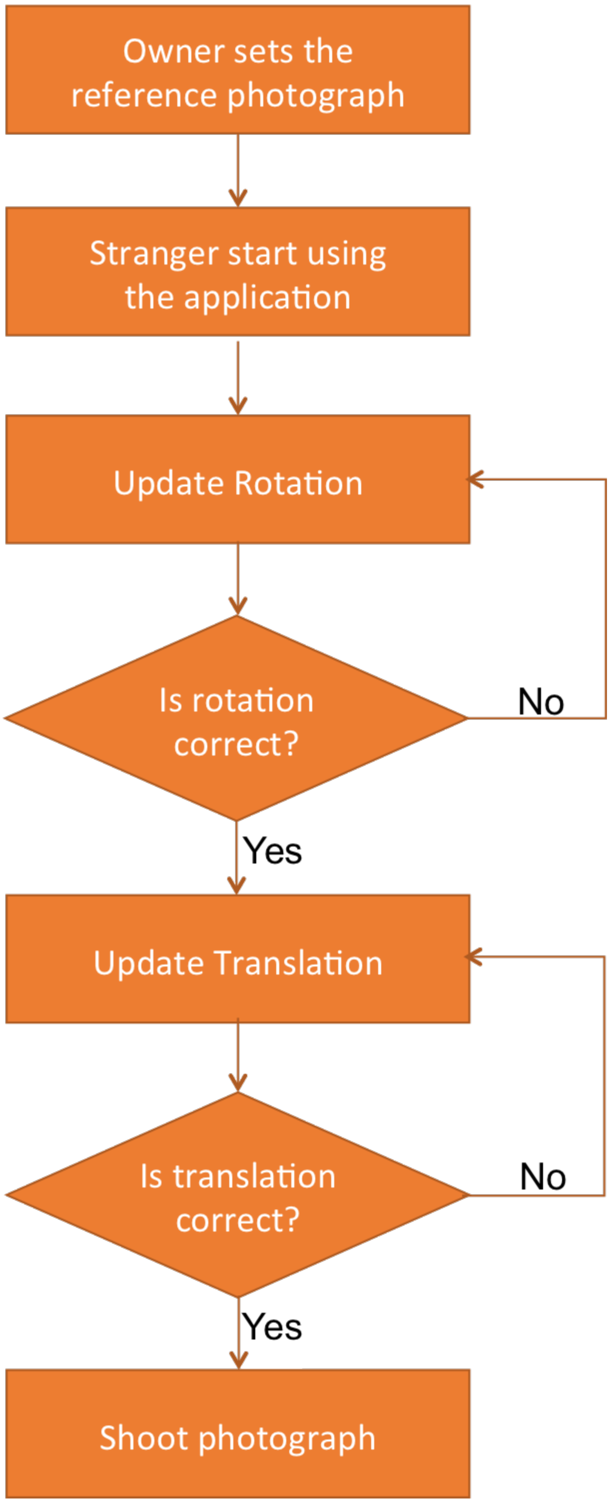
To aid the user on how much he has to translate the camera the blue arrow changes its size according to magnitude of the distance vector. Figure 3.11 shows the UI with a big arrow (when compared to Figure 3.10) indicating that the user is far from the correct placement.

At this stage, all the screen is dimmed black, a message asking for the user to hold the camera steady is shown and the interaction is over.

This proposed interface compared to Bae's, show in Figure 3.12, has fewer elements that the user needs to focus on. Our proposed interface also makes use of a 3D environment which creates a more natural interaction regarding to rotations and translations in the real world.

The UI elements are drawn in OpenGLES 2.0 using the Rajawali engine to simplify development. The use of OpenGL for the interface grants that it will be as responsive as possible because of the OpenGL refresh rate. Standard UI elements from the Android API do not provide a fast refresh rate.

Figure 3.8: The application flow chart from the moment where the owner sets the reference image to when the final image is shot



Source: Created by the author.

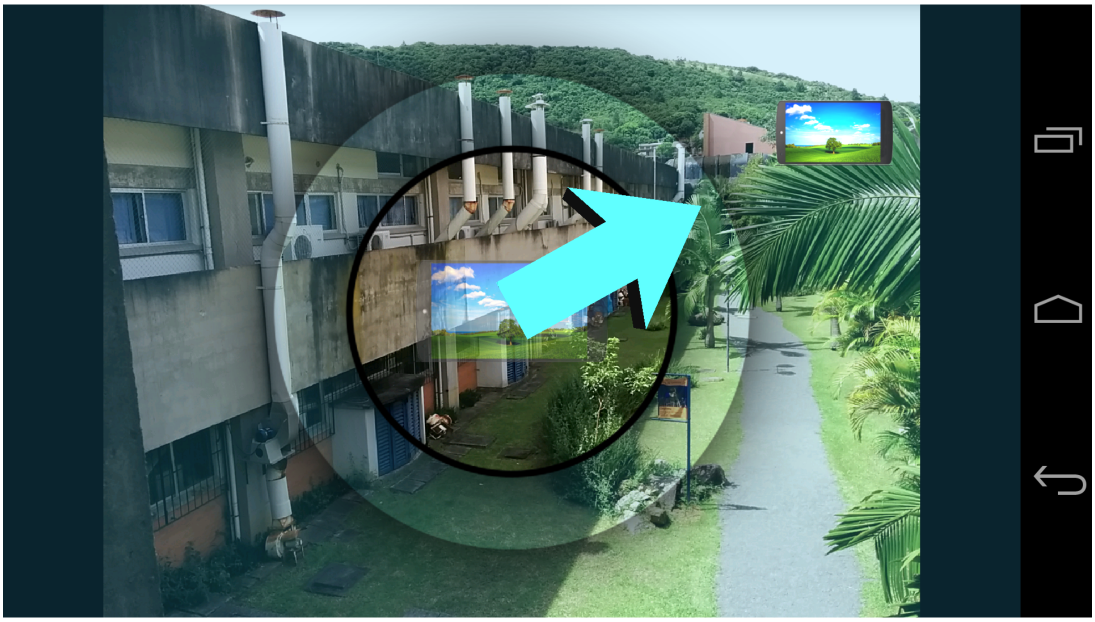
Figure 3.9: The UI that the camera owner sees

Source: Created by the author.

Figure 3.10: The UI that the stranger interact with

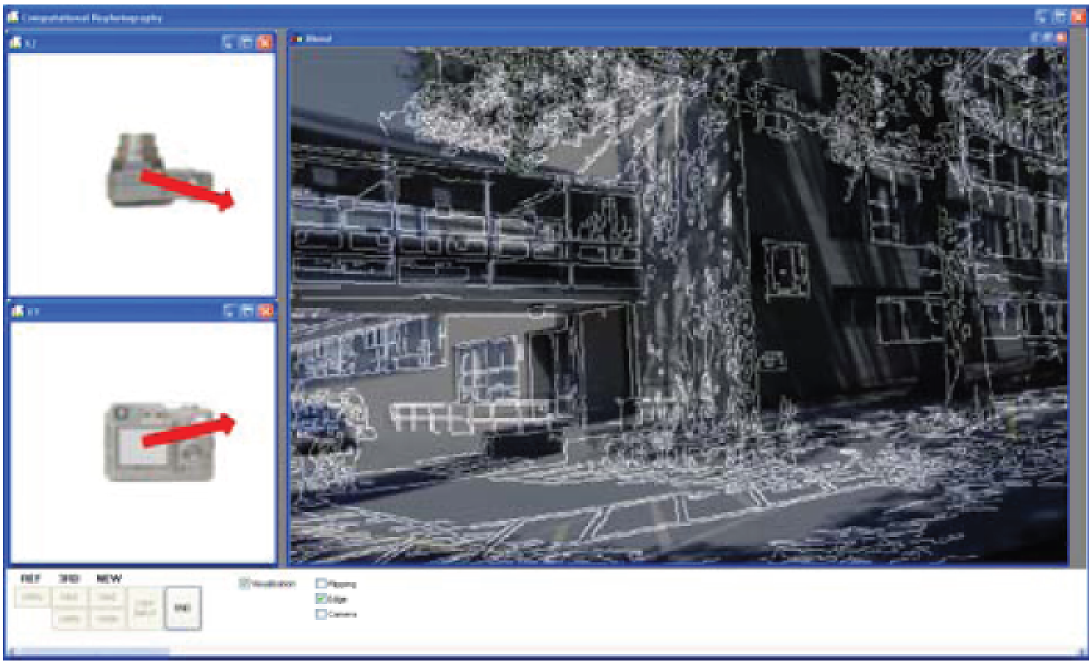
Source: Created by the author.

Figure 3.11: Image of stranger UI with a big guiding arrow



Source: Created by the author.

Figure 3.12: Bae's work user interface



Source: (BAE; AGARWALA; DURAND, 2010)

4 EXPERIMENTAL EVALUATION

The proposed study in this work is based on probability sampling where the target population are men and women from the age of 15 to 55 years old. The participants are selected trough stratified sampling in parks around town and at the university campus.

4.1 Study Design A

This case study was designed in order to evaluate three aspects: how fast and easy it is to use the proposed interface and how similar is the reference photograph to the one taken by the application. In order to measure the first part, it is recorded the time span between the moment where the participant starts interacting with the application and the moment when the photograph is taken. We evaluated the second based on if the user did or did not completed the task,

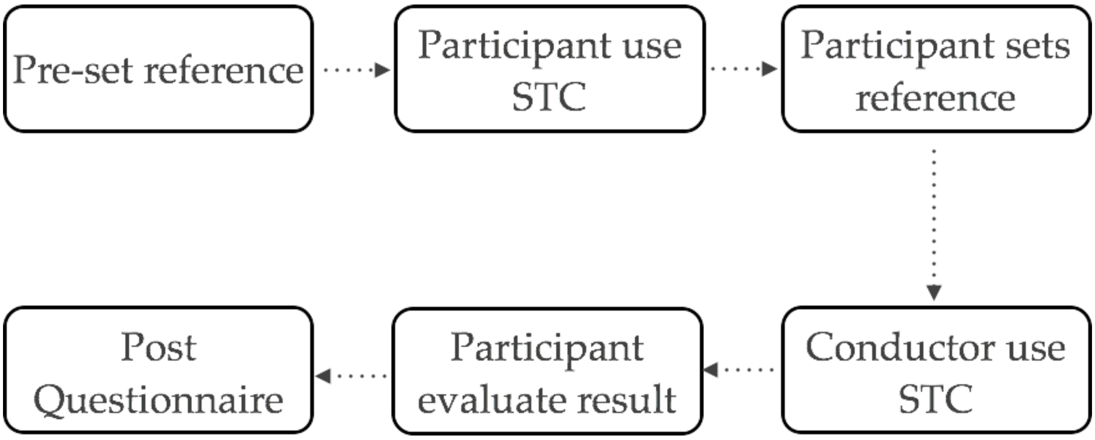
For the third aspect, the participant is asked to imagine and take a reference photograph where the only constraint is that it has to be in landscape orientation to minimize the number of variables. After this the final image and the reference one are shown and the participant is required to answer using a 5 point Likert scale, how similar the final image is compared with the one he imagined. Figure 4.1 illustrates the flow of this study.

In this study the participants where asked to interact with the application before being informed of what was the application objective. It was only told them what the application was for after they completed both tasks.

Subjects were asked to freely participate on the study in either the university campus or touristic places around the city of Porto Alegre. Before the test begin they were asked to answer a small questionnaire (Appendix A) to asset the sample data.

- Sample size: 40
- Average age: 26.1 σ +/- 7.86 years
- 20% worked with informatics
- 55% Women and 45% Men

Figure 4.1: Diagram demonstrating the flow of study A



Source: Created by the author.

4.2 Study Design B

For this second study, different participants were selected and introduced to the following scenario: *You are a tourist visiting the city of Porto Alegre. Lets assume that you want a photo taken from you here at this park. You are going to need the assistance of a stranger to shoot it for you. Assume that another person (another participant) is that stranger. First imagine the photograph without you on it and take a reference image. Then go to the place where you imagined you would be and the stranger will try to replicate the reference photograph using the application.*

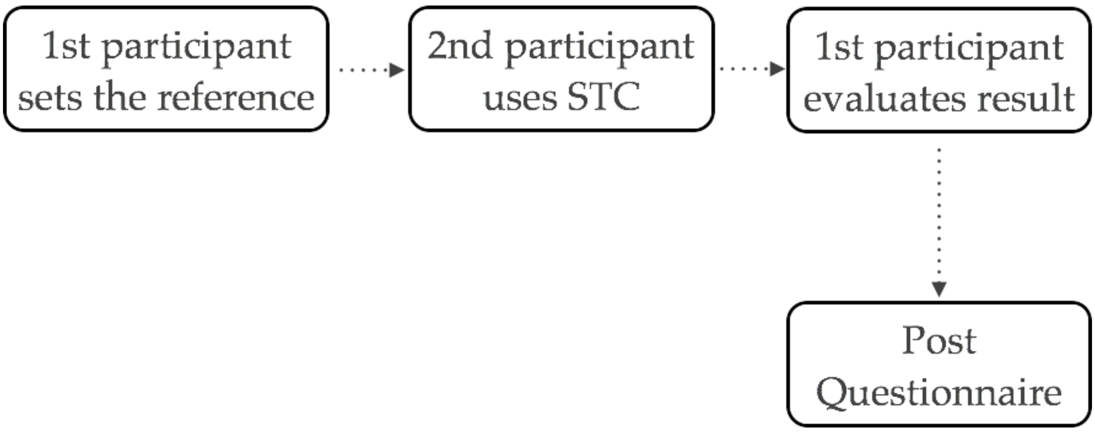
Once the image was taken, the first participant was asked to evaluate, using a 5 point Likert scale, if the final photograph is similar to the one they have previously imagined. Figure 4.2 illustrates the flow for this study.

- Sample size: 20
- Average age: 29.05 σ +/- 9.54 years
- 0% worked with informatics
- 45% Women and 55% Men

4.3 Final remarks

Although both tests share some similarities, in the study A the participants were not required to think on a photograph where they need to be therefore giving them more options. Because of that the study A worked more as a stress test to the application while the study B served as a field test.

Figure 4.2: Diagram demonstrating the flow of study B



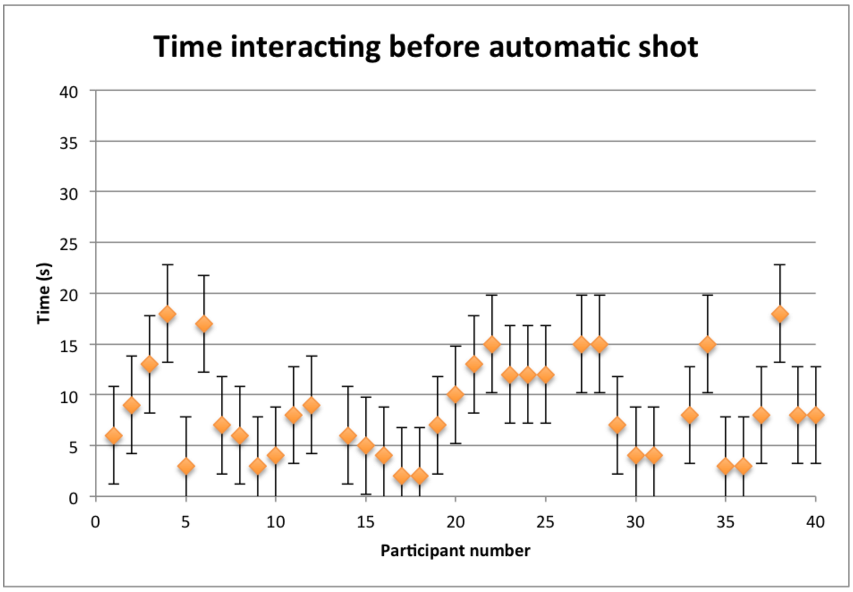
Source: Created by the author.

5 RESULTS

5.1 Study A

By the end of this study we extracted an average time of 8.62 seconds before the automatic shot was taken with a σ =4.79 after removing three outliers. Chart 5.1 shows the time of each participant and the standard deviation. All the users from study A where able to complete the study. When asked about their expertise with photographic cameras the participants answers fitted a normal distribution as seen in chart 5.2 with a tendency to the *No Expertise* side.

Figure 5.1: Chart showing the time each participant took to shoot the automatic photograph

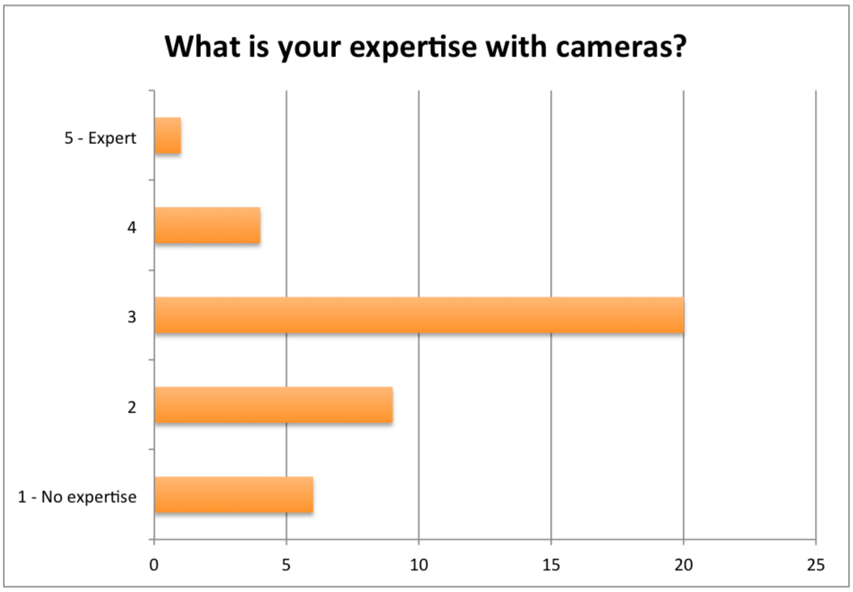


Source: Created by the author.

For the post task questionnaire questions, 90% of the participants answered that the reference image and the final image where similar (ratings 4 and 5 of the scale) as seen in chart 5.3. To the second question, 87.5% of the participants answered that they would use the application (ratings 4 and 5) during one of their travels if the application was available. This result is shown in chart 5.4.

In Figure 5.5 we can see one of the results from study A. Figure 5.5a shows the reference image with the extracted keypoints and 5.5b shows the final image.

Figure 5.2: Chart with the answers for the first question of the pre-questionnaire.



Source: Created by the author.

Figure 5.5: Example of pair of reference and final photographs from a test session

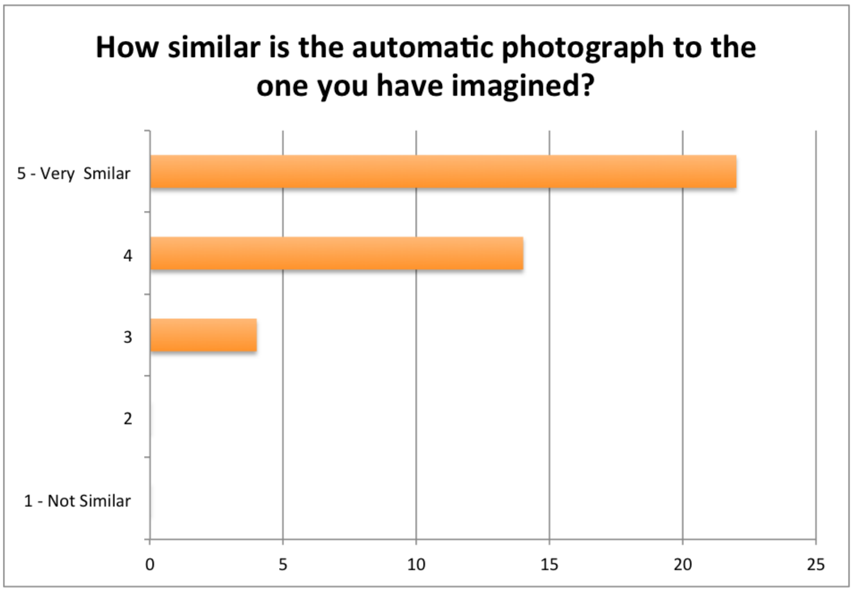
(a) Example of a reference photograph. Red circles are the extracted keypoints

(b) The final image taken automatically by a participant using STC

Source: Created by the author.

Because the participants were free to choose any scenario to shoot the photograph, the application worked with some images that were not its focus such as Figure 5.6. However, the application was able to take the automatic photograph nevertheless.

Figure 5.3: Chart with the answers for the second question of the questionnaire.



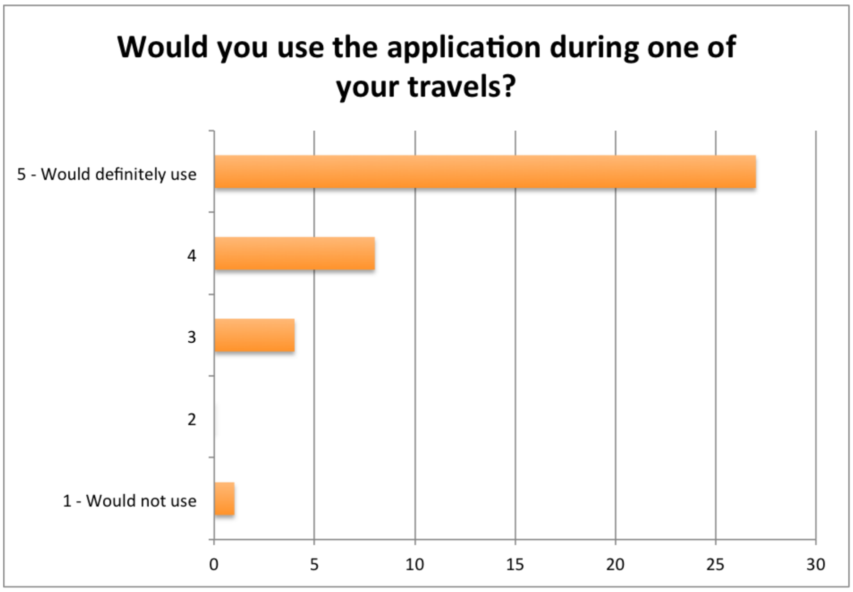
Source: Created by the author.

5.2 Study B

The results for the second study are similar to the ones from study A. The major difference between both studies was the use of two participants and the fact that now the photograph that ought to be taken was restricted to one where the first participant needed to appear.

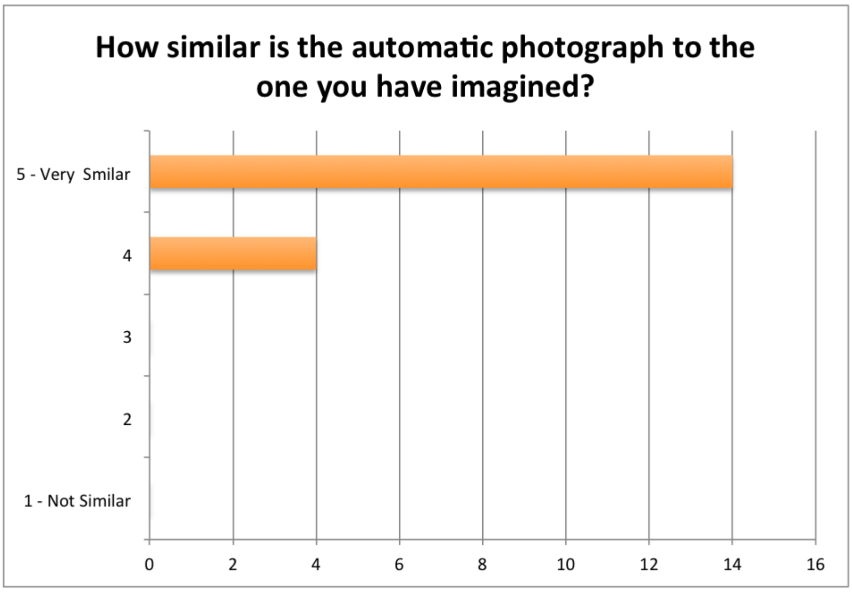
The chart 5.8 shows that 100% of the participants responded that the final photograph was similar to the one that they had imagined (ratings 4 and 5). 80% of the participants also answered that they would use the application if it was available during their travels as shown in chart 5.7.

Figure 5.4: Chart with the answers for the third question of the questionnaire.



Source: Created by the author.

Figure 5.7: Chart with the answers for the first question of the questionnaire.

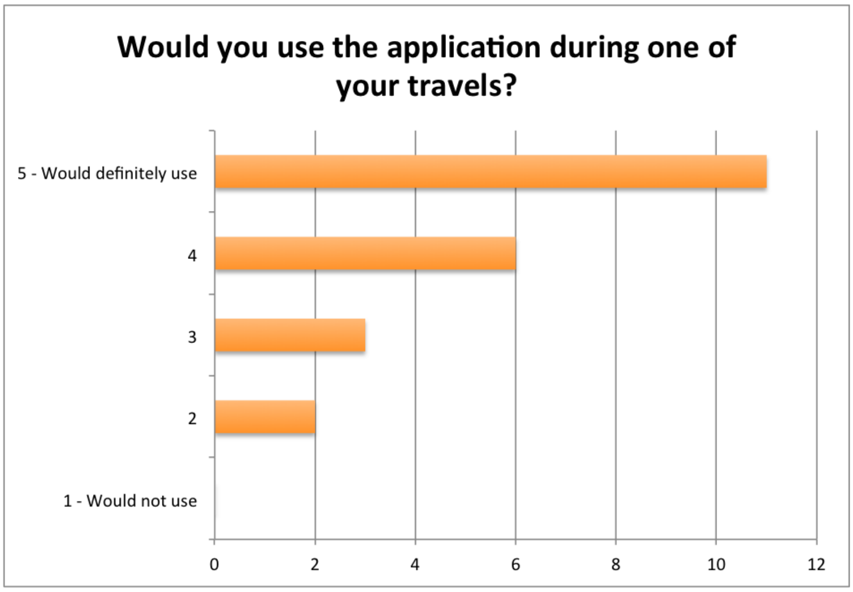


Source: Created by the author.

Figure 5.6: A photograph shot by a participant in study A that is not part of the scope of images for the application

Source: Created by the author.

Figure 5.8: Chart with the answers for the second question of the questionnaire.



Source: Created by the author.

Below in Figures 5.9, 5.10, 5.11, 5.12 we can see some of the images taken during this study, on the left side are the reference photographs and on the right the final ones.

Figure 5.9: Image pair for participant #1

Source: Created by the author.

Figure 5.10: Image pair for participant #6

Source: Created by the author.

Figure 5.11: Image pair for participant #8

Source: Created by the author.

Figure 5.12: Image pair for participant #10

Source: Created by the author.

6 CONCLUSIONS AND FUTURE WORK

By the end of this work we were able to propose, implement and test a new application that could solve one of the problems people encounter during their travels. With the advent of STC, now it is possible for someone to ask for a stranger to shoot them without worrying about how that stranger will frame the image.

During the tests most of the participants were surprised by the existence of such application and were very receptive to its concept. Participants were also surprised by how simple it was to make it work. With that in mind, it is possible to assume that there is a lack for such application in the market and that if released to public STC will have a good acceptance.

One reason for that thought is that when asked to comment about their experience with photographs taken from strangers, all of them shared the same memories quite a few frustrating photographs. The major complain was that even if they did not liked the photograph they usually felt uncomfortable to ask for that stranger to shoot it again. The ones that actually did asked also commented that sometimes even with a couple of different shots they could not get the final result that they were expecting.

6.1 Future Work

Although STC goal was reached, there are still adjustments that need to be done before its final release. The main reason is that the objective of this work was to propose and test this new concept.

One of the future adjustments is to create a method for the camera owner to set an area of interest so the camera can measure light correctly. The reason for that feature is to avoid problems with back-lightning where the subject is not exposed correctly. Such a problem can be seen in Figures 6.1 and 6.2.

Figure 6.1: Final image of user #16 where it is possible to see exposure problems.

Source: Created by the author.

Figure 6.2: Final image of user #9 that shows exposure problems with back-lightning.

Source: Created by the author.

Some participants also had problems keeping the frame properly aligned with the horizon.

Figure 6.3: Final image taken by some user during Study A. Note that the image is slightly rotated.

Source: Created by the author.

This issue created a couple of images that where slightly rotated such as Figure 6.3. To avoid this issue it would be interesting to add an artificial horizon line to the screen. However that feature needs to be tested in order to prove that will not make the interface harder to use.

6.1.1 Study Design C

There is also a third study scheduled. On this last study the participants will be divided in two groups. Both groups will shoot pictures on the exact same spot. Participants from the first group will be asked to shoot a photograph using a PnS camera without any restriciton or instruction. The other group, will be asked to shoot a photograph using our application with the reference image already set.

Then, all the images will be sent to a third independent group. This group will be asked to rank the images and chose the best five images without knowing which are the ones that were taken by the application or by a PnS camera.

REFERENCES

ADAMS, A. et al. The frankencamera: An experimental platform for computational photography. In: **ACM SIGGRAPH 2010 Papers**. New York, NY, USA: ACM, 2010. (SIGGRAPH '10), p. 29:1–29:12. ISBN 978-1-4503-0210-4. Available from Internet: <http://doi.acm.org/10.1145/1833349.1778766>.

BAE, S.; AGARWALA, A.; DURAND, F. Computational rephotography. **ACM Trans. Graph.**, ACM, New York, NY, USA, v. 29, n. 3, p. 24:1–24:15, jul. 2010. ISSN 0730-0301. Available from Internet: <http://doi.acm.org/10.1145/1805964.1805968>.

BAY, H. et al. Speeded-up robust features (surf). **Comput. Vis. Image Underst.**, Elsevier Science Inc., New York, NY, USA, v. 110, n. 3, p. 346–359, jun. 2008. ISSN 1077-3142. Available from Internet: <http://dx.doi.org/10.1016/j.cviu.2007.09.014>.

HOL, J. D. et al. Sensor fusion for augmented reality. In: **FUSION**. IEEE, 2006. p. 1–6. Available from Internet: <http://dblp.uni-trier.de/db/conf/fusion/fusion2006.html#HolSGS06>.

JOSHI, N. et al. Image deblurring using inertial measurement sensors. In: **ACM SIGGRAPH 2010 Papers**. New York, NY, USA: ACM, 2010. (SIGGRAPH '10), p. 30:1–30:9. ISBN 978-1-4503-0210-4. Available from Internet: <http://doi.acm.org/10.1145/1833349.1778767>.

LEVOY, M. Experimental platforms for computational photography. **IEEE Computer Graphics and Applications**, v. 30, n. 5, p. 81–87, 2010. Available from Internet: <http://dblp.uni-trier.de/db/journals/cga/cga30.html#Levoy10>.

LOWE, D. G. Object recognition from local scale-invariant features. In: **Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2**. Washington, DC, USA: IEEE Computer Society, 1999. (ICCV '99), p. 1150–. ISBN 0-7695-0164-8. Available from Internet: <http://dl.acm.org/citation.cfm?id=850924.851523>.

MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE Computer Society, Washington, DC, USA, v. 27, n. 10, p. 1615–1630, oct. 2005. ISSN 0162-8828. Available from Internet: <http://dx.doi.org/10.1109/TPAMI.2005.188>.

REUS, V. et al. Correcting user's head and body orientation using a comfort pose function. In: **Proceedings of the 2013 XV Symposium on Virtual and Augmented Reality**. Washington, DC, USA: IEEE Computer Society, 2013. (SVR '13), p. 167–174. ISBN 978-0-7695-5001-5. Available from Internet: <http://dx.doi.org/10.1109/SVR.2013.33>.

RUBLEE, E. et al. Orb: An efficient alternative to sift or surf. In: **Proceedings of the 2011 International Conference on Computer Vision**. Washington, DC, USA: IEEE Computer Society, 2011. (ICCV '11), p. 2564–2571. ISBN 978-1-4577-1101-5. Available from Internet: <http://dx.doi.org/10.1109/ICCV.2011.6126544>.

WOODMAN, O. J.; WOODMAN, C. O. J.; WOODMAN, O. J. **An introduction to inertial navigation**. 2007.

YANG, X.; CHENG, K.-T. T. Accelerating surf detector on mobile devices. In: **Proceedings of the 20th ACM International Conference on Multimedia**. New York, NY, USA: ACM, 2012. (MM '12), p. 569–578. ISBN 978-1-4503-1089-5. Available from Internet: <http://doi.acm.org/10.1145/2393347.2393427>.

APENDIX A

Questionário STC

* Required

Número *

Idade *

Sexo *

Trabalha com desenvolvimento de Software/Hardware? *

- SIm
- Não

Experiência com máquinas fotográficas *

	1	2	3	4	5	
Pouco experiente	◯	◯	◯	◯	◯	Muito experiente

A foto automática ficou parecida com a que você imaginou? *

	1	2	3	4	5	
Pouco parecida	◯	◯	◯	◯	◯	Muito parecida

Tentaria utilizar o aplicativo em alguma viagem?

	1	2	3	4	5	
Pouco provável	◯	◯	◯	◯	◯	Muito provável

« Back   Submit

Never submit passwords through Google Forms.