UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

**DAVI EBERT BOBSIN - 00246764**

# TESTES PRÁTICOS DE SISTEMAS DE VISÃO PARA SISTEMAS EMBARCADOS

Porto Alegre
2020

**DAVI EBERT BOBSIN - 00246764**

# TESTES PRÁTICOS DE SISTEMAS DE VISÃO PARA SISTEMAS EMBARCADOS

Trabalho de Conclusão de Curso (TCC-CCA) apresentado à COMGRAD-CCA da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de *Bacharel em Engenharia de Controle e Automação* .

ORIENTADOR:

Prof. Dr. Marcelo Götz

Porto Alegre
2020

**DAVI EBERT BOBSIN - 00246764**

# TESTES PRÁTICOS DE SISTEMAS DE VISÃO PARA SISTEMAS EMBARCADOS

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção dos créditos da Disciplina de TCC do curso *Engenharia de Controle e Automação* e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Marcelo Götz, UFRGS
Doutor pela Universität Paderborn, Alemanha

Banca Examinadora:

Prof. Dr. Marcelo Götz, UFRGS
Doutor pela Universität Paderborn, Alemanha

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS
Doutor pela Universidade Federal de Minas Gerais, Brasil

Prof. Dr. Valner João Brusamarello, UFRGS
Doutor pela University of Virginia, Estados Unidos

_____
Marcelo Götz
Coordenador de Curso
Engenharia de Controle e Automação

Porto Alegre, novembro de 2020.

# DEDICATÓRIA

Dedico este trabalho primeiramente aos meus professores de contrato vitalício: meus pais. Este trabalho e todo o percurso só foram possíveis e suportáveis pelo seu apoio incondicional.

Dedico também a minha irmã que, dentre outras coisas, me inspirou a trilhar a jornada que se encerra neste trabalho.

E por último, dedico aos meus amigos e colegas do Instituto Senai e ZF, que despertaram em mim o interesse pelo tema.

# AGRADECIMENTOS

Por este trabalho ser uma realidade, agradeço ao meu orientador e aos meus revisores e amigos Alex Treviso, Émerson de Barros, Leonardo Nogueira, Guilherme Linck e em especial a Igor Diesel, que compartilhou boa parte dos trabalhos e momentos durante a faculdade.

Aos meus professores, principalmente por me instigarem ao olhar crítico.

Aos meus colegas e amigos do CEECA, RSRacing UFRGS e LAROSE, por compartilharem estes projetos comigo e ampliarem os meus horizontes durante a faculdade.

Aos meus amigos de Bateria Minotrago, por me proporcionarem leveza e alegria durante a faculdade.

# RESUMO

Muitas aplicações modernas de sistemas embarcados envolvem a utilização de câmeras de vídeo como sensor. A possibilidade de se capturar informações contidas no ambiente e o crescente desenvolvimento na área de extração de informações através de imagens tornam as câmeras escolhas atraentes para percepção autônoma.

Algoritmos de visão dinâmicos, ou seja, que empregam a informação temporal para extrair a métrica desejada, assumem um valor constante como base para cálculos utilizando amostragem e sincronização. Além disso, o valor absoluto das cores para *pixels* e a resposta à incidência luminosa são distintos em diferentes sensores.

Este trabalho propõe, de foma ampla, uma metodologia de testes que pode ser utilizada para avaliar características relevantes em câmeras de vídeo, trazer mais robustez para os sistemas e auxiliar no desenvolvimento de sistemas de visão.

De forma específica, os testes propõem a medição da frequência de captura, resposta do sensor à incidência luminosa e defasagem entre instantes de captura em um sistema multicamera.

O conceito geral é, ao criar um ambiente controlado para o sensor, garantir uma entrada de dados conhecida. Então, ao processar a imagem recebida pela câmera, faz-se possível avaliação da saída do sistema de visão. Assim sendo, os testes propostos podem ser vistos como validações que abrangem um sistema completo de captura, do momento de percepção do ambiente até as imagens estarem disponíveis para processamento.

O trabalho também descreve a implementação prática da metodologia proposta, a fim de exemplificar e comprovar os premissas através de testes.

A precisão obtida com a prova de conceito apresentada se mostrou limitada aos componentes e equipamentos utilizados. Mesmo assim, os conceitos propostos para cada teste puderam ser comprovados e através dos resultados a metodologia proposta pôde ser validada.

**Palavras-chave: Controle e Automação, Visão Computacional, Processamento de Imagem, Sistemas Embarcados, Engenharia Elétrica, Microcontrolador, Linux.**

# ABSTRACT

Many modern embedded systems' applications make use of video cameras as a sensor. The possibility of capturing information contained in the environment along with growing development in the computer vision field makes cameras an attractive choice for autonomous perception.

Dynamic vision algorithms, i.e., which employ temporal information to extract the desired metric, assume a constant value as the basis for calculations using sampling and synchronization. In addition, the absolute value of colors for *pixels* and the response to light incidence are distinct among different sensors.

This work proposes a testing methodology that can be used to evaluate relevant characteristics in video cameras, bring more robustness to the systems, and assist in vision systems development.

Specifically, the tests propose the measurement of the capture frequency, sensor response due to light incidence, and delay between capture moments in a multi-camera system.

The general concept is, by creating a controlled environment for the sensor, ensure a known data input. Then, when processing the image received by the camera, it is possible to evaluate the output of the vision system. Thus, the proposed tests can be seen as validations covering a complete capture system, from the moment the environment is perceived until the images are available for processing.

The work also describes the practical implementation of the proposed methodology, in order to exemplify and prove the premises through tests.

The accuracy obtained with the presented proof of concept was bounded to the components and equipment limits. Even so, the proposed concepts for each test could be proved and through the results, the proposed methodology could be validated.

**Keywords: Control and Automation, Computer Vision, Image Processing, Embedded Systems, Electrical Engineering, Microcontroller, Linux.**

# SUMÁRIO

# LISTA DE ILUSTRAÇÕES

# LISTA DE TABELAS

# LISTA DE ABREVIATURAS

| | |
|---|---|
| API | Application Proggraming Interface |
| CSI | Camera Serial Interface |
| FPS | Frames per Second |
| HSV | Hue, Saturation, Value |
| RGB | Red, Green, Blue |
| UART | Universal Asynchronous Receiver/Transmitter |

# 1 INTRODUÇÃO

Com o desenvolvimento de câmeras de vídeo digitais, a extração de informações utilizando sistemas processados se tornou possível. E com o rápido crescimento da tecnologia microeletrônica, câmeras com resoluções cada vez maiores e com menor ruído foram sendo desenvolvidas. Isso tornou cada vez mais comum a utilização de câmera como sensor (AUTOPILOTREVIEW, s.d.).

Hoje em dia, câmeras de vídeo tem sido vastamente utilizadas em sistemas autônomos. Muitos algoritmos atuais utilizam as informações contidas nos *frames* para tomada de decisões nas áreas de produção industrial, robótica e direção autônoma.

Tomando a área automotiva como exemplo, sistemas baseados em vídeo podem substituir espelhos traseiros de um carro (SCHMIDT et al., 2016), estimar o movimento (ULLMAN; BRENNER, 1979), ser utilizado em sistemas de alerta de colisão frontal (FCW, do inglês *Front Collision Warning*) (LIN et al., 2012), extração de informação visual (DE LA ESCALERA et al., 2004) e em sistemas de direção autônoma (HILLEL et al., 2014).

Estes algoritmos utilizam valores absolutos e pré-definidos para frequência de amostragem. Além disso, nos sistemas com mais de uma câmera não são levados em consideração a diferença entre os instantes de captura entre as câmeras. Então, com algoritmos e sistemas embarcados cada vez mais complexos, é importante garantir que o sistema de visão irá se comportar como pré-estabelecido, que os dados sejam íntegros e as diferenças instantâneas entre as câmeras não sejam prejudiciais ao resultado extraído.

Por estes motivos, o presente trabalho propõe uma metodologia de testes que avalia características como frequência de amostragem e sincronismo entre câmeras. Tais características são relevantes principalmente para sistemas embarcados, onde uma frequência de amostragem mais precisa pode ser útil para se economizarem recursos de processamento, normalmente mais limitados nesse tipo de sistema. E como visão estéreo e/ou multicâmera vem sendo utilizados em sistemas embarcados devido ao acréscimo de informações providas em tais estruturas, o sincronismo entre as câmeras se torna um parâmetro relevante em tais sistemas.

Além disso, aplicações que realizam a leitura de parâmetros físicos através da cor ou iluminação se apoiam em um modelo que relacione bem a incidência luminosa na câmera e o valor digital obtido pelo sensor. Por isso, uma metodologia de teste para esta relação também é proposta.

O conceito geral é fazer uso de um sistema controlado que irá gerar sinais luminosos bem definidos e conhecidos. Uma câmera deverá então fazer a leitura e o processador do sistema embarcado deverá analisar os dados recebidos. Os sinais a serem gerados podem ser vistos como a entrada do sistema de teste, bem como os *frames* recebidos já processados podem ser vistos como a saída. As informações obtidas com os testes podem ser definidas como a relação entre a entrada controlada e a saída obtida.

Portanto, o objetivo do trabalho é postular e provar o funcionamento de teste de frequência de amostragem, teste de sincronismo entre câmeras e método de levantamento de modelo teórico de relação entre incidência luminosa e valor do *pixel*. O teste de frequência amostral cujo valor representa a quantidade de *frames* recebidos em um intervalo de tempo, tem seu resultado expresso em $Hz$ (escolha da unidade justificada no texto). O teste de sincronismo de câmera tem como resultado o intervalo entre a captura das duas câmeras, medido em *ms*. E por fim, o método para definição de relação entre luminosidade e valor do *pixel* é realizado através do tempo de acionamento de uma fonte luminosa, portanto o modelo possui unidade $contagem/s$.

A estrutura do trabalho contém uma revisão da literatura, reservada ao Capítulo 2, fundamentando conceitos que serão importantes ao longo do texto. O Capítulo 3 estabelece a metodologia de testes para câmeras de video, correspondendo ao cerne do trabalho. Com o objetivo de se executar uma prova de conceito para o método proposto, foi necessário implementar algumas ferramentas e sistemas auxiliares. O desenvolvimento de tais ferramentas é descrito no Capítulo 4. Os resultados da aplicação do método proposto, utilizando a prova de conceito, estão brevemente descritos no Capítulo 5. Uma análise mais rigorosa de tais resultados, bem como conclusões que avaliam a validade dos testes propostos encerram o trabalho no Capítulo 6.

# 2 REVISÃO DA LITERATURA

## 2.1 Câmeras digitais

Câmeras podem ser definidas como dispositivos capazes de capturar e gravar a variação de intensidade da luz. Nas primeiras abordagens, as capturas e as gravações eram realizadas por meio de reações químicas no filme fotográfico, e o seu armazenamento de dados utilizava sistemas analógicos similares aos disquetes. Atualmente, as câmeras capturam as imagens convertendo a luz em sinais elétricos, e são armazenadas como sinais elétricos em dispositivos de silício, como memórias *flash* (NAKAMURA, 2017).

Para possibilitar a medição de intensidade de luz em forma de sinais elétricos, é necessária a utilização de algum dispositivo que tenha as suas propriedades físicas alteradas pela interação com a luz. Estes componentes são definidos como fotosensores ou fotodetectores, e medem a intensidade da radiação incidente, que pode ser expressa em fluxo de prótons (DALLA BETTA, 2011).

Um conjunto destes sensores de luz é capaz de formar uma imagem, traduzindo a luz presente, em uma cena, em um sinal elétrico. Estes conjuntos são chamados de sensores de imagens estáticas (*steady-state image sensors*) ou simplesmente *imagers* (NAKAMURA, 2017).

### 2.1.1 CCD

Com o objetivo inicial de criar um dispositivo de memória capaz de mover cargas pela superfície de um semicondutor, projeto hoje em dia chamado de *bubble memory*, Willar Sterling Boyle e George Elwood Smith criaram o dispositivo de carga acoplada (do inglês *Charged Coupled Device*, CCD). Posteriormente, fazendo uso desta tecnologia, as primeiras câmeras eletrônicas foram criadas (CRESSLER, 2016).

O princípio por traz desta tecnologia é a criação de um *array* de capacitores condutores-isolantes-semicondutores que, quando aplicado a tensão correta nas camadas de condutores, possibilitam o movimento de uma região com um mínimo potencial elétrico. Desta forma, pela movimentação da região com mínimo potencial, as cargas elétricas também são movimentadas na superfície do semicondutor (BOYLE; SMITH, 1970).

### 2.1.2 CMOS

Quando o CCD era a principal tecnologia utilizada para captura digital de imagens, outro sistema estava sendo proposto para sensoriamento de imagem, cujo conceito era uma matriz/mosaico de fotodetectores (SCHUSTER; STRULL, 1966). Os componentes CMOS (*Complementary metal-oxide semiconductor*) provém de um sistema de chaveamento eletrônico capaz de amostragem da luz e armazenamento com a capacitância do sensor.

Apesar de ser uma tecnologia mais recente, apresentava uma maior quantidade de ruído quando comparada com o CCD (NAKAMURA, 2017), devido à chamada corrente parasita escura, intrinsecamente presente em fotodiodos e ao ruído durante a leitura de cada célula da matriz (HSU et al., 2004). Quando sistemas system-on-chip (embarcados) de câmeras foram propostos, unindo matrizes de sensores, sistemas de endereçamento, amplificadores e processadores lógicos em um chip, o ruído da tecnologia CMOS pôde ser manejado, transformando a tecnologia CMOS interessante para câmeras, permitindo o controle sobre exposição e triggered synchronization (WANG et al., 1991).

Com sensores ativos usando tecnologia de pinned photodiode (PPD) combinada com processamento lógico *on-chip*, hoje em dia o CMOS apresenta menos ruído até mesmo que sensores de imagem CCD (KOZLOWSKI; LUO; TOMASINI, 1999).

### 2.1.3 Propriedades da Câmera Digital

#### *2.1.3.1 Tempo de exposição*

O tempo de exposição é o período no qual os fotodetectores são expostos à luz, permitindo a captura. Nas primeiras câmeras esse período era controlado por um obturador que fechava e bloqueava a chegada da luz no sensor. Atualmente, é possível controlar diretamente a alimentação dos sensores.

#### *2.1.3.2 Frequência de captura*

Este parâmetro de vídeo é calculado pelo número de frames capturados em um determinado período de tempo, comumente usado na unidade de frames por segundo. Em uma câmera digital diversos fatores podem mudar a frequência de captura, entre eles: os tempos de limpeza, escaneamento e envio dos dados de cada sensor; o tempo de exposição de cada frame; limitações do protocolo usado; clock usado pela câmera;

### 2.1.4 Processamento digital de cores

Pode-se encontrar três tipos de receptores de luz no olho humano. Estes receptores são cones com diferentes tamanhos que permitem a percepção de diferentes frequências. O processamento desses sinais pelo cérebro é definido como a percepção de uma cor (NAKAMURA, 2017).

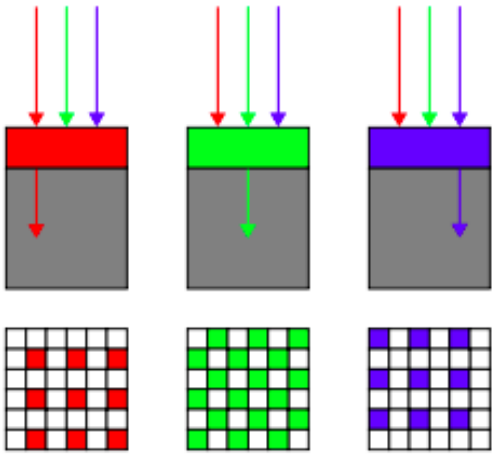
Com o intuito de desenvolver um sistema padronizado que defina o que é uma cor de forma quantitativa, a CIE (International Commission on Illumination) criou o que é chamado de modelo de aparência de cores (CAM) (MORONEY et al., 2002). Nesses documentos foram estabelecidos padrões e descritos os seguintes conceitos:

- Brilho - a quantidade de luz percebida em um estimulo,
- Matiz ou angulo da matiz - Quão similar o estimulo é de um padrão definido de cores.
- Saturação - Quão colorido o sinal é em relação ao seu próprio brilho.

Depois de desenvolver um dispositivo para capturar a luz que representa uma cena, foi necessário representar as cores vistas por humanos através destes dispositivos. Então, usando lentes, mascarou-se a frequência da luz sendo possível filtrar a intensidade correspondente à cor desejada. Com esta técnica desenvolvida, foi proposto um padrão de filtro que é conhecido até hoje como filtro Bayer(BAYER, 1976).

Na Figura 1 estão representados os sensores com seus filtros e abaixo a organização destes em uma matriz formando um filtro Bayer.

**Figura 1:** *Ilustração da filtragem das componentes com filtro de Bayer.*



Fonte: Wikipedia.

Para especificar, criar e visualizar as cores nas câmeras digitais e em diferentes dispositivos, existem métodos chamados modelos de cores. Usando estes métodos as cores são representadas por ênuplos de números, em geral três ou quatro valores para cada cor. A combinação dos valores numéricos de um modelo resulta em um conjunto de cores que é chamado espaço de cores (FAIRCHILD, 2005). Em seguida serão apresentado três dos mais tradicionais e usados modelos de cores, são eles RGB, HSV e YUV.

#### *2.1.4.1 RGB*

O RGB é um modelo de cores aditivo, no qual diferentes intensidades das cores primárias, vermelho, verde e azul, são adicionadas para produzir outras cores. O nome do modelo é construído com as iniciais das cores usadas por ele. Este modelo é usado em diferentes dispositivos de entrada como câmeras digitais, câmeras de vídeo e *scanners*, além de dispositivos de saída como tecnologias de *display* LCD, plasma e OLED (HIRSCH, 2004).

O RGB é dependente dos dispositivos, devido à diferenças intrínsecas em cada fotosensor, e por isso normalmente é necessário que haja algum gerenciamento de cores para que diferentes dispositivos reproduzam as mesmas cores.

Os *pixels* no modelo RGB são compostos pela mescla de diferentes intensidades das cores verde, vermelho e azul. Estas intensidades podem variar em diferentes intervalos dependendo da aplicação. Em computadores, o mais comum é que cada cor varie entre 0 e 255, sendo armazenadas em 8-bits. Em dispositivos de mais alta tecnologia, é comum que cada cor seja representada por um número maior de bits, podendo chegar a valores como 64-bits, aumentando assim o número de cores que podem ser criadas ou visualizadas.

#### *2.1.4.2 HSV*

Uma alternativa ao modelo RGB, é o modelo HSV no qual os ênuplos de números armazenam as intensidades matiz, saturação e valor. Como explanado anteriormente, a escala de matiz representa a similaridade da cor em questão com uma das cores primárias ou uma combinação delas, esta escala tem inicio em 0, representando vermelho, passa

por 120º, representando verde, por 240º, representando azul e se encerra em vermelho novamente em 360º. Como a escala sugere, o espaço de cores do HSV possui uma geometria cilíndrica onde o matiz é a dimensão angular. A saturação seria a variação do raio do cilindro, especificando a qualidade do matiz pelo grau de mesclagem deste com a cor branca e por fim o valor representa o brilho da cor e geometricamente seria a altura do cilindro HSV (FAIRCHILD, 2005).

O fato de o modelo HSV separar cor de luminosidade faz com que ele seja usado com frequência em visão computacional se mostrando mais relevante que modelos como o RGB onde os valores de todos os parâmetros estão fortemente atrelados à luminosidade incidindo sobre o objeto calculado (CHENG et al., 2001).

#### *2.1.4.3 YUV*

O modelo de cores YUV é definido em termos de um componente de luminância (Y) e dois elementos de crominância (UV). Este modelo é principalmente usado na transmissão de imagens e possui a característica de mascarar de forma mais eficiente à percepção humana erros de transmissão e compressão.

O modelo surgiu com o desafio de se transmitir o sinal de televisão em cores e que fosse compatível com o modelo anterior, preto-e-branco. O modelo preto-e-branco usava o componente luma, também nomeado como Y' e que é um correlato perceptual da luminância. Assim, para transmitir cores foram adicionados os dois novos componentes, U e V (MALLER, 2003).

## 2.2 Código de Gray

Criado por Frank Gray, este sistema de código binário busca representar valores de forma que somente um bit varie entre números consecutivos. Esta codificação foi inventada com o intuito de diminuir o consumo de energia e a produção de ruídos em circuitos lógicos rudimentares montados com válvulas termoiônicas e dispositivos eletromecânicos (DORAN, 2007).

Atualmente este sistema de codificação é usado quando a mudança do valor de apenas um bit é um fator facilitador para a aplicação, dessa forma acaba-se criando uma tabela de relação entre os códigos e números representados.

## 2.3 RaspberryPi

RaspberryPi é uma plataforma portátil para desenvolvimento, e conta com um processador Quad core 1.2 GHz Broadcom BCM2837 e 1gb de memória RAM. Pode-se dizer que a Raspberry Pi é um computador inteiro em uma placada de desenvolvimento única e com capacidade de rodar sistemas operacionais Linux, ou até Windows 10 IoT, o qual é uma versão para sistemas embardados (UPTON; HALFACREE, 2014).

Uma das principais características da plataforma é a sua ampla capacidade de conectividade. Dentre seus componentes para esta finalidade, pode-se citar:

- Duas portas USB 2.0 para conexão de periféricos e dispositivos de armazenamento;
- Uma porta RCA para conexão com televisores;
- Uma porta HDMI (*High Definition Multi-media Interface*) permite conexão com televisores de alta definição;

- Suporte para DSI (*display Serial Interface*)
- 26 pinos de GPIO (*(general purpose input and output)* arranjadas em duas portas com 13 pinos cada;
- Suporte para CSI (*camera serial interface*)

Os pinos do GPIO podem ser configurados como entrada ou saída, contendo também suporte para protocolos embarcados, como *serial peripheral interface* (SPI) e *Inter-Integrated Circuit* ($I^2C$), sendo esta a principal interface de comunicação com outros componentes e circuitos (MAKSIMOVIĆ et al., 2014). Contudo, para a aplicação apresentada neste trabalho, uma das características importantes da RaspberryPi é a conexão CSI (*camera serial interface*, que exclui a necessidade de utilização de uma porta USB para a câmera. A interface CSI introduz um hardware dedicado aumentando a performance de aplicações que façam uso de câmeras e facilitando a comunicação com as mesmas.

## 2.4 Light Emition Diodes (LEDs)

Diodos emissores de luz são dispositivos semicondutores, que surgiram na década de 60 em sua primeira versão comercial emitindo luz vermelha (THORSETH, 2011). A passagem de corrente em um LED é permitida apenas em um sentidos, assim como em diodos tradicionais, sendo um dispositivo formado por uma junção P-N. Na junção, o lado P contém lacunas enquanto o lado N contém excesso de elétrons (cargas negativas). Quando polarizado diretamente, o movimento das cargas e das lacunas resulta na emissão de fótons (DENBAARS, 1997).

Novas cores de LEDs surgiram na década de 90, como azul e verde (THORSETH, 2011). A luz emitida por um LED é monocromática e o comprimento de onda gerado está relacionado ao material utilizado no semicondutor. Gálio, alumínio, arsênio, fósforo, nitrogênio, índio, e a combinação desses elementos no semicondutor, altera o comprimento de onda da luz gerada (DENBAARS, 1997). O fato de luz produzida não ser resultado de incandescência, mas sim do movimento das lacunas e elétrons, proporciona para os LEDs uma característica desejável em muitas aplicações, alta velocidade de chaveamento do dispositivo (THORSETH, 2011).

# 3 METODOLOGIA

O presente trabalho propõe metodologias para testagem de câmeras, que visam explicitar as limitações e as características do sensor e do sistema de visão como um todo. Os testes, melhor definidos nas próximas subseções, são os seguintes:

- Ajuste de modelo matemático para relacionar incidência luminosa com valor de *pixel*;
- Teste de frequência de amostragem;
- Teste de sincronismo entre câmeras em sistema multivisão.

A metodologia apresentada tratará dos conceitos e procedimentos de forma ampla e se abstrairá de qualquer ferramenta específica, uma vez que a proposta do trabalho é descrever os métodos para medição e não a implementação das medições em si.

## 3.1 Ambiente controlado

Uma vez que a informação a ser lida e avaliada será a luminosidade provida por fontes controladas, se faz necessário a mitigação de possíveis ruídos luminosos externos que possam prejudicar a leitura do sistema de visão sendo testado. Por isso, deve ser feito uso de envólucro que permita o mínimo de penetração de luz. Isso pode ser obtido na prática por uma caixa fechada de material opaco que permita um distanciamento mínimo dos componentes evitando a reflexão dos materiais internos à caixa.

O distanciamento mínimo pode ser empírico de forma que todas as fontes luminosas acionáveis sejam contempladas pela imagem produzida pelo sensor e sem reflexos perceptíveis em *frames* (nas laterais da caixa e possíveis estruturas internas).

Além da construção física de um ambiente escuro, é necessário o uso de equipamento que garanta uma iluminação constante durante períodos de tempo que possam ser especificados e modificados em tempo de execução durante as etapas de calibração e testes. Por desejar-se avaliar o desempenho de um sistema de visão para um caso de aplicação prática, o tempo de abertura da câmera é considerado fixo em valor conhecido. Para que se possa variar a quantidade de luz acumulada por um sensor sem a necessidade de controlar o tempo de integração da câmera, controla-se então o tempo de acionamento da fonte luminosa, e assumindo que este acionamento é rápido o suficiente para tal dinâmica ser desprezível essa ferramenta permite realizar uma inferência do valor lido esperado.

## 3.2 Análise Pré-teste

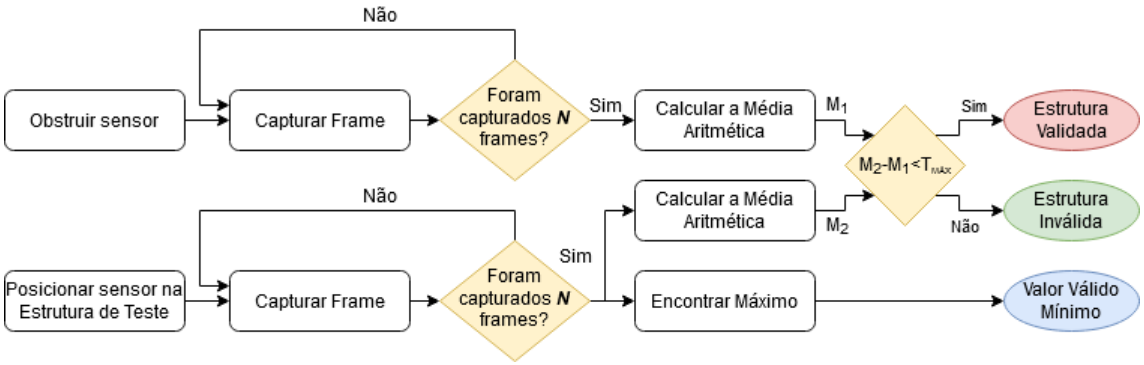
Nesta etapa inicial, procura-se recolher os parâmetros e especificações relevantes da câmera e do ambiente para o desenvolvimento dos testes.

### 3.2.1 Parâmetros do ambiente controlado

O primeiro parâmetro a ser analisado é o quão escuro o ambiente de testes é. O nível de escuridão pode ser definido como proporcional ao valor máximo da luminosidade em um *frame*. Para o interior do ambiente de testes, pode ser medido em contagens (que variam entre 0 e $2^n$), em imagens capturadas com a caixa fechada. Para se estabelecer que o ambiente não permite ruídos luminosos externos, se estabelece um valor máximo para a incidência externa. Uma forma de se estabelecer esse limite é obstruindo a lente com alguma tampa ou proteção opaca para a mesma e verificando nessas condições qual é o valor máximo obtido. Por se fazer presente uma carga elétrica mínima no sensor, mesmo quando na ausência de luz (cor preta), este valor pode e provavelmente será diferente de zero devido a ruídos intrínsecos nos fotosensores ou amplificadores. Esta classe de ruído é chamada corrente escura (TAKAYANAGI et al., 2001).

Então, visando testar o ambiente escuro a ser utilizado nos testes, o método demonstrado pelo fluxograma disposto na Figura 2 será utilizado.

**Figura 2:** *Fluxograma para teste do ambiente.*




Fonte: Elaborado pelo autor.

O primeiro conjunto de $N$ capturas deve ser feito com a lente obstruída e o segundo, com a câmera já posicionada no ambiente fechado onde serão realizados os testes. As métricas utilizadas para análise foram definidas como o valor máximo absoluto e média. O máximo representa um valor de ruído que pode voltar a acometer um dos testes a ser performado, por isso pode ser considerado como nível mínimo a partir do qual os valores lidos são válidos. A média tem o objetivo de ser utilizada como métrica para a luminosidade do *frame* e deve ser utilizada para a comparação entre os dois casos. Assumindo a resolução da imagem muito maior do que os valores numéricos dos *pixels*, a média tende a ser nula, por isso não se traduz em uma informação relevante sobre o ruído. Porém, é uma métrica útil para se avaliar a luminosidade dentro do ambiente de testes. No caso de um acréscimo de uma contagem em todos os *pixels*, a média será acrescida também de uma unidade, tornando possível se definir um limite de luminosidade $T_{MAX}$.

A variável $N$ deve ser definida para a realização do teste. Para a escolha de $N$, deve-se levar em consideração dois fatores: tempo de teste e quantidade de amostragem. Estes valores devem ser comparados com quantidades desejadas dentro do contexto de análise.

Se a mesma se tratar de um único experimento para caracterizar o sensor, o tempo não é um fator tão relevante quanto a melhor caracterização do ruído. Se este teste irá compor um grupo de testes a ser realizado repetidamente, o tempo se torna o fator possivelmente mais relevante.

A título de exemplo, tomaremos $N = 100$ *frames*. Em 100 amostras, uma variação de uma contagem no resultado de uma das leituras representa 1%. Se for levado em consideração a intensidade máxima lida na imagem para se avaliar o ruído, esse valor usualmente já é menor que a quantidade percentual de ruído preto. E para serem capturados 100 *frames* em 30 FPS, o tempo do teste é de aproximadamente 3 segundos.

### 3.2.2 Parâmetros da câmera

Como os testes se utilizarão das características relacionadas com a exposição, cores e a frequência de amostragem nominal, essas são informações necessárias. Tais informações visam ainda verificar se os testes são válidos para o sensor escolhido e podem ser obtidas no *datasheet* do sensor ou manual da câmera.

Relacionado à exposição, sensores CMOS possuem a configuração de tempo de integração/exposição em linhas. Um contador interno à câmera é responsável por indicar qual linha da matriz de sensores deve ser lida. Este contador é incrementado em um intervalo de tempo constante. Por isso, convenientemente os sensores empregam a quantidade de linhas percorridas como unidade de tempo. E esta unidade é utilizada para descrever o tempo de integração. Este normalmente é um parâmetro ajustável. Quando fixo, deverá ser encontrado na especificação do sensor. Além disso, parte dos sensores possuem o autoajuste de exposição, o qual pode ser desativado através do envio de comandos de configuração para a câmera.

Quanto à percepção das cores, usualmente os amplificadores presentes em cada um dos fotosensores possuem um ganho comum ou global e também ganhos ajustáveis por grupos de sensores com um mesmo filtro. Isso significa que para uma câmera que utiliza filtro Bayer (BAYER, 1976), sendo esse um mosaico de um filtro vermelho, um azul e dois verdes para compor cada *pixel*, será encontrado um ajuste de ganho para a cor vermelha, um para azul e dois ajustes para verde.

Também referente aos ganhos, alguns chips implementam um ajuste automático de ganhos. Isso é realizado através de uma verificação estatística das cores na imagem coletada feita pelo próprio *hardware* da câmera, que configura o ganho em cada uma das componentes.

A sensibilização do sensor, normalmente descrita e ajustada em um valor na escala de sensibilidade normalizada pela ISO (*International Standardization Organization*), é encontrada nas configurações ou especificações da câmera.

Alguns sensores possuem ainda o *Auto-White Balance*, que alteram em tempo de captura esses ganhos de forma automática, balanceando as componentes buscando um valor de branco padrão.

É importante antes de realizar os testes estar ciente da existência ou não de todos os ajustes automáticos citados para o sensor selecionado, que podem interferir ou ainda invalidar os dados lidos. Para os testes propostos, todos os ajustes automáticos serão desativados.

A frequência de amostragem de uma câmera está relacionada com a resolução da região de interesse, o espaçamento entre linhas definido como tempo em branco (ou *blank time*), a frequência de *clock* da câmera e o tempo de integração. (C-CAM, s.d.)

Conhecendo-se estas características e parâmetros da câmera, é possível ajustar e

configurar os parâmetros dos testes a fim de aumentar a precisão dos resultados ou permitir uma convergência mais rápida.

Além de otimizar o resultado dos testes, obtendo-se tais informações é possível realizar a construção de um modelo teórico para simular valores a serem lidos na execução dos testes. O objetivo final do modelo teórico deve ser gerar *frames* para que os mesmos possam ser comparados com os obtidos durante a execução dos testes, ou ainda sinais para verificação dos valores absolutos em um ou mais *pixels*.

### 3.2.3 Calibração espacial

Uma vez que cada fonte luminosa representa uma informação da referência, o primeiro passo é encontrar, dentro da matriz de valores do *frame*, onde estão posicionadas cada uma destas fontes.

O procedimento projetado para esta calibração consiste nos seguintes passos:

1. Acionar uma única fonte de luz com alimentação constante e capturar N *frames* do ambiente escuro controlado.

2. Com os *frames* capturados, realizar um processamento de imagem para filtrar e criar uma máscara binária com as regiões da imagem que possuem cores próximas às emitidas pela fonte de luz.

3. Utilizando um algoritmo de clusterização, encontram-se agrupamentos de *pixels* nesta máscara de cor.

4. Utilizando *thresholds* de quantidade mínima de pixeis agrupados que podem representar uma fonte luminosa, verificar se existe algum *cluster* que possa representar a fonte luminosa e se esse *cluster* é único.

5. Caso a fonte luminosa não tenha sido encontrada, aumentar a emissão da fonte luminosa. Caso mais de uma fonte luminosa tenha sido encontrada, diminuir o intervalo de coloração para criar a máscara. Se a fonte luminosa foi encontrada, repetir o procedimento para a próxima fonte.

## 3.3 Teste de parâmetros da Câmera

Sabendo que o valor numérico dos *pixels* é uma leitura digital da carga acumulada em cada um dos fotosensores, é possível criar uma relação entre incidência luminosa e valor em contagens (entre 0 e $2^n$ bit). Por se tratar de um sensor com tecnologia CMOS, pode-se aproximar a característica da curva de acumulo de carga semelhante à um capacitor. E por isso, espera-se que a relação entre incidência e valor lido tenha forma exponencial.

Para realizar o teste, propõe-se fixar o período de exposição e variar o tempo de incidência luminosa do LED. Para garantir que o período do pulso luminoso esteja englobado no tempo de exposição, utilizam-se as fontes auxiliares também gerando pulsos de forma intervalada, permitindo uma janela de incidência da luz principal que será lida. Um dado é considerado válido se as bordas dessa janela, formadas pelas outras fontes, puderem ser lidas.

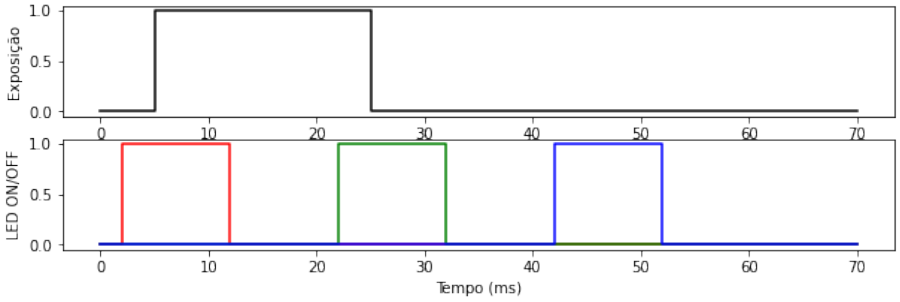
Então, realizando a leitura das três fontes luminosas, serão avaliados as componentes de *Value* para cada uma das fontes. O valor obtido para as fontes auxiliares será utilizado através de um *threshold* para verificar se a leitura da fonte principal é válida. O valor da fonte principal é o valor sendo analisado, com a qual se fará a correlação de exposição e

valor absoluto do pixel. A fonte principal é definida como a que se encontra entre outras duas fontes, pelo menos, chamadas de auxiliares.

O *frame* lido é considerado inválido se uma das fontes de luz auxiliares não tiver atingido valor de intensidade mínimo (pode ser empírico, de forma a garantir que o LED esteja ligado). No exemplo das Figura 3 e Figura 4, são ilustrados sinais de acionamento de LEDs (utilizados como fontes de luz) bem como o sinal de exposição, valor 1 representando o sensor sensibilizado, capturando a luminosidade e 0 o sensor dessensibilizado. Neste caso, o LED verde é o definido como principal e os outros dois são definidos como auxiliares.

Para o caso demonstrado na Figura 3, percebe-se que o tempo de acionamento do LED principal (verde) não é completamente englobado pelo tempo de exposição.

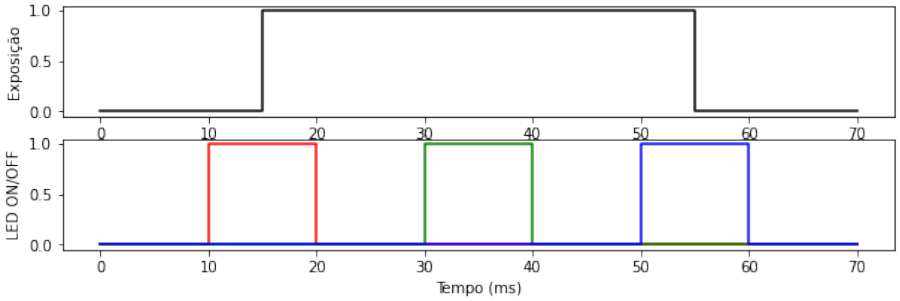
**Figura 3:** *Exemplo de sinais em frame inválido.*




Fonte: Elaborado pelo autor.

Para o caso demonstrado na Figura 4, o tempo de exposição engloba os LEDs auxiliares e consequentemente, o LED principal. Assim, o requisito de ser possível ler um valor mínimo vindo dos LEDs auxiliares garante que o período do LED principal foi lido em sua totalidade.

**Figura 4:** *Exemplo de sinais em frame válido.*




Fonte: Elaborado pelo autor.

Os períodos de acionamento (valores controlados) podem ser considerados como a entrada do experimento e a leitura das contagens obtida para os *pixels* pode ser considerada a saída. A correlação entre as duas grandezas é feita pelo ajuste da curva entrada $x$ saída por uma curva semelhante à uma curva de acúmulo carga (exponencial negativa), feita através de um processo de minimização de uma função custo entre pontos reais encontrados e ajuste.

## 3.4 Teste de frequência de amostragem

O conceito aqui é testar a frequência exata de amostragem. A frequência é uma grandeza que pode ser medida pela variação do número de *frames* obtidos em um período de tempo definido. Portanto, a precisão da medição está associada ao equipamento que irá fazer a relação entre o *frame* capturado e o instante temporal em que o mesmo foi capturado. Por isso, propõe-se a utilização de um sistema controlado com períodos bem definidos que será utilizado de referência. Utilizando as fontes luminosas como emissores de sinais binários para a câmera através da luminosidade, a câmera será exposta a essa contagem que será utilizada como referência para saber a ordem e o momento que o sensor capturou o devido *frame*.

Tendo-se um período de tempo como base, como neste caso a contagem binária, após a leitura dos *frames* é possível recuperar a contagem gerada. Além disso, utilizando o intervalo de base gerado, por ser um equipamento controlado, é possível saber em que momento cada contagem foi emitida. Juntando as duas informações da contagem recebida pelo sensor e em que momento ela foi emitida, é possível inferir qual a frequência de amostragem da câmera. Nota-se que a precisão está também relacionada com a quantidade de amostras, uma vez que será mais fácil encontrar diferenças significativas na contagem em um número maior de amostras.

## 3.5 Teste de sincronismo

Utilizando o mesmo princípio de geração de pulsos luminosos com período controlado, é possível avaliar a diferença entre a leitura de múltiplas câmeras. O princípio é a comparação entre as contagens lidas através dos *frames*.

O procedimento é exibir a contagem através das fontes luminosas em uma frequência maior do que a de amostragem das câmeras. Se a frequência for muito próxima da de captura das câmeras, podem ser necessárias várias amostras para que se consiga ter diferença significativa entre os *frames* obtidos pelas diferentes câmeras. Por outro lado, se a frequência de contagem for muito rápida, pode ser difícil avaliar o valor real da contagem sendo exibida pelas fontes luminosas. Isso se deve ao fato de que a janela de exposição estará sendo exposta a mais de um valor de contagem. A integração da incidência de mais de uma contagem pode tornar o valor indefinível.

Por isso, para se escolher um período para os pulsos luminosos, leva-se em consideração como limite superior a utilização de período igual ao de amostragem e como limite inferior o menor período para que se possa definir a medida de luminosidade. Em termos numéricos, pode-se definir o limite inferior fazendo uso da relação entre luminosidade e leitura (que pode ser testado como descrito na Seção 3.3). Utilizando uma exposição fixa nas câmeras de modo que os valores lidos não saturem, o período mínimo é virtualmente igual à precisão da relação entre tempo de exposição e valor lido. Abaixo desta precisão, não se pode definir quando uma fonte representa uma contagem ou não. Ou ainda, se o valor é logicamente verdadeiro ou falso.

Se as câmeras não tiverem mecanismo de sincronismo entre elas, outro ponto relevante a se analisar antes da realização do teste é a diferença entre a frequência de amostragem das câmeras. Uma vez que se sabe essa diferença, pode-se inferir qual será a taxa de aumento do assincronismo entre as câmeras.

# 4 DESENVOLVIMENTO

Para validar a metodologia proposta, uma prova de conceito foi construída, implementando as ferramentas necessárias descritas na Seção 3. A presente seção tem como objetivo descrever as etapas do desenvolvimento desta prova de conceito.

## 4.1 Ambiente controlado

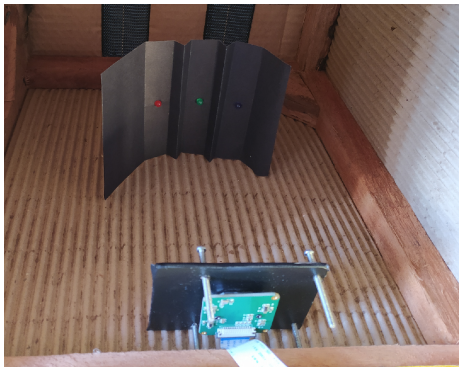
Na Subseção 3.2, define-se que o ambiente onde os testes serão realizados deve ser escuro, para evitar ruído luminoso nas leituras da câmera. Por isso, para a prova de conceito será construída uma caixa escura, na qual a câmera testada bem como as fonte de luz estarão contidas.

Para a construção da caixa escura, se utilizou uma estrutura cúbica de madeira já existente que possui forro de papelão. Uma tampa de papelão também foi utilizada para garantir que nenhuma luz entre na caixa.

A primeira estrutura de referência que se imaginou foi um conjunto de três LEDs de cores distintas. Para manter os LEDs fixos e impedir que a luminosidade de um dos LEDs influencie no outro, foi construída uma estrutura com aletas entre os emissores que impede o efeito de *cross-talk* (a incidência luminosa de um LED refletir em outro). O ajuste da posição dos elementos foi feito de forma empírica, e foi verificado que todas as fontes de luz estão contempladas na imagem capturada, como pode ser visto na Figure 5b que exibe a perspectiva da imagem produzida pela câmera. A Figura 5a mostra a construção da estrutura utilizada.

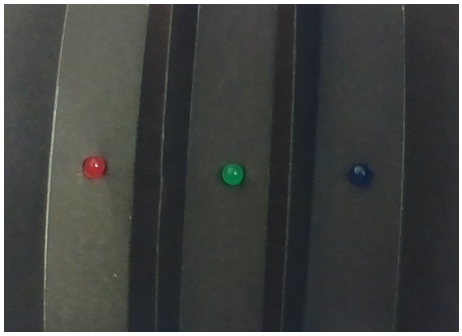
**Figura 5:** *Imagens do ambiente de teste montado.*

**(a)** *Caixa utilizada para ambiente sem infiltração de luz.*



Fonte: Elaborado pelo autor.

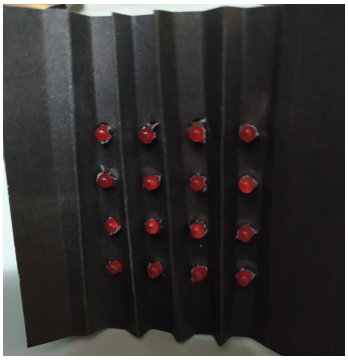
**(b)** *Perspectiva da câmera do sistema montado.*



Fonte: Elaborado pelo autor.

Para servir de referência de contagem nos testes que utilizam um contador em frente às câmeras, foi desenvolvida também uma matriz de LEDs. A escolha da utilização de uma matriz se dá pela grande quantidade de combinações, permitindo um maior ciclo de contagem do que com a estrutura de três LEDs. Isso possibilita um maior número de amostras em um mesmo teste e diminui também a possibilidade de redundância nos valores lidos. Tal matriz de LEDs já montada pode ser vista na Figura 6.

**Figura 6:** *Matriz de LED construída.*



Fonte: Elaborado pelo autor.

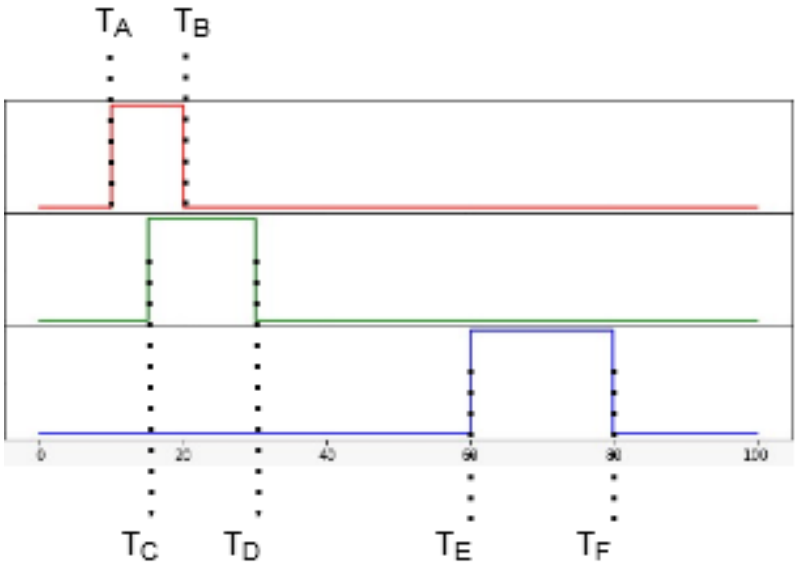
Os resultados a análise do teste de validação da caixa escura construída podem ser vistos na Seção 5.

## 4.2 Unidade de Acionamento

Além dos três LEDs utilizados como fonte luminosa, a unidade de acionamento é composta também por um microcontrolador Teensy 3.2. A escolha pela utilização de tal recurso se deve à possibilidade de implementação de um sistema de temporização bem definido, utilizando o mecanismo de interrupção presente nessa classe de microcontroladores. Além disso, a velocidade de *clock* na placa de desenvolvimento permite a criação de uma função de interrupção em um intervalo suficientemente curto para chaveamento dos LEDs em períodos menores do que o tempo de exposição da câmera, permitindo a execução dos testes. A placa possui um *clock* de 72 $MHz$, por isso, o tempo de execução de uma instrução é de aproximadamente 14 $ns$. E portanto, utilizando tal microcontrolador são possíveis chaveamentos em períodos na ordem de microssegundos.

A programação do microcontrolador foi feita utilizando a linguagem C. O valor referente a um intervalo de tempo base em microssegundos é armazenado em uma variável convencionada $T_{STEP}$. Em múltiplos deste intervalo são definidos os períodos de chaveamento dos LEDs. Assim, Uma variável $T_{TOTAL}$ define em quantos $T_{STEP}$ o ciclo de acionamento será reiniciado. As variáveis $T_A$,$T_C$ e $T_E$ definem em que múltiplo de $T_{STEP}$ os LEDs serão alimentados e $T_B$, $T_D$ e $T_F$ definem em qual contagem os LEDs serão desligados. Para melhor entendimento, a Figura 7 demonstra um ciclo de acionamento.

**Figura 7:** *Gráfico ilustrando o papel de cada variável nos acionamentos.*




Fonte: Elaborado pelo autor.

Para ser possível alterar os intervalos de acionamento em tempo de execução, foi implementada uma interface de comunicação UART que pode receber do usuário comandos para verificação e configuração de qualquer uma das variáveis de temporização citadas anteriormente. Com a construção destes elementos, pode-se garantir que as medições terão pouca influência de iluminação externa, o tempo de chaveamento dos LEDs é pelo menos uma ordem de grandeza menor que o tempo de amostragem e se terá controle sobre as variáveis de temporização durante o tempo de execução dos testes.

## 4.3 Unidade de Processamento

Na unidade de processamento, a prova de conceito requer um software que possibilite:

- Capturar todos *frames* sendo enviados por uma câmera de vídeo;
- Criar máscaras para cada uma das cores das fontes luminosas, a fim de segmentá-las;
- Inferir a posição de cada uma das fontes luminosas, bem como uma região de avaliação;
- Avaliar os valores de luminosidade dentro de cada uma das regiões encontradas.

### 4.3.1 Captura de frames

Se fez necessário implementar uma aplicação que fosse capaz de capturar todos os *frames* recebidos pelo processador. Para isso, utilizou-se da API V4L2 para Linux (SCHIMEK, s.d.), permitindo que assim que cada *frame* transmitido pela câmera fosse salvo em um buffer e, assim que a transferência estivesse concluída, o espaço de memória fosse disponibilizado para utilização a nível de aplicação. Isso garante que todos os *frames* serão recebidos e armazenados em *buffer*, ficando a cargo da aplicação descartar tais *frames* na necessidade de uma aplicação mais veloz ou que consuma menos recursos de máquina. A aplicação desenvolvida pode ser dividida em três etapas: inicialização, captura

e encerramento. Todos os procedimentos de configuração da câmera ou dos *buffers* é feita através da API, que permite uma maneira direta de enviar e receber dados dos *drivers*.

A inicialização da câmera é a etapa de configuração dos parâmetros do sensor e dos espaços de memória que serão utilizados pelos *drivers* para salvar os *frames*. A API Linux para captura de vídeo ainda conta com um sistema de filas que é configurado nesta mesma etapa. As configurações relevantes desta etapa são:

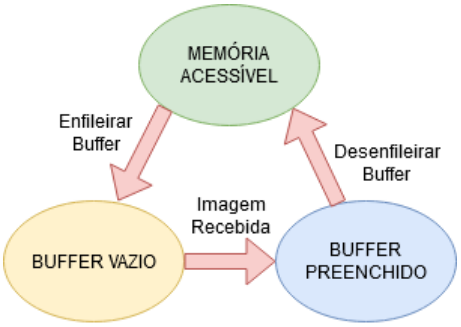
- Resolução: Foi escolhida a resolução máxima recomendada da câmera, utilizando a maior região de interesse possível em aplicações a serem desenvolvidas com essa câmera;
- Formato dos *pixels*: pelo fato do chip possuir um pré-processador interno na câmera, dentre as possibilidades disponibilizadas pelo próprio chip da câmera, está o formato RGB. Neste caso, o processo de *demosaic* já é realizado pela câmera, por isso, por economia de esforços e recursos do lado do processador, foi utilizado este formato para todos os testes;
- Exposição: a exposição foi implementada no modo manual, para que não ocorram ajustes automáticos que possam manipular os dados sendo coletados. Além disso, o valor absoluto da exposição pode ser configurado através de um arquivo de configuração passado juntamente ao executável da aplicação.
- Balanço de branco: o ajuste de *Auto White-Balance* foi desativado, impedindo que os ganhos de cada uma das cores seja ajustado pelo próprio sensor em tempo de execução;
- *Buffers* e Filas: a aplicação foi desenvolvida de modo que a quantidade de *buffers* possa ser especificada no arquivo de configuração. A aplicação lê esta quantidade e faz a requisição ao *driver* de espaços de memórias e posições na fila para os mesmos. Se a requisição for bem sucedida, os espaços dedicados à cada um dos *buffers* são retornados, bem como o seu tamanho.

Após realizar todas as configurações, a etapa de inicialização é completa iniciando o *streaming* de video.

Por padrão para as aplicações Linux, utilizado também neste projeto, os *buffers* são organizados de forma cíclica. Os *buffers* são enfileirados, fazendo com que os mesmos estejam disponíveis para armazenar informações recebidas da câmera. Quando as informações são recebidas, os *frames* preenchem os membros da fila. Para processar as informações em espaço de memória disponível para a aplicação, o *buffer* é desenfileirado e pode ser processado. Assim que um *buffer* estiver disponível para ser preenchido por um novo *frame*, ele pode ser colocado novamente na fila.

A etapa de captura então verifica se algum *frame* está disponível para ser processado. Em caso afirmativo, o respectivo *buffer* é desenfileirado, o *frame* é processado e após o processamento o *buffer* é recolocado na fila. É importante perceber que se a retirada dos *frames* da fila for mais lenta do que a recepção dos mesmos, todos os *buffers* serão preenchidos e os novos *frames* sendo recebidos serão descartados. Outro ponto relevante da implementação é que a função de processamento chamada pela etapa de captura é feita por uma função separada definida em tempo de compilação. Isso permitiu que se pudesse utilizar essa mesma estrutura de captura de *frames* para os diferentes testes, alterando-se apenas a função de processamento.

**Figura 8:** *Representação dos diferentes estados e eventos para Buffers de vídeo V4L2.*




Fonte: Elaborado pelo autor.

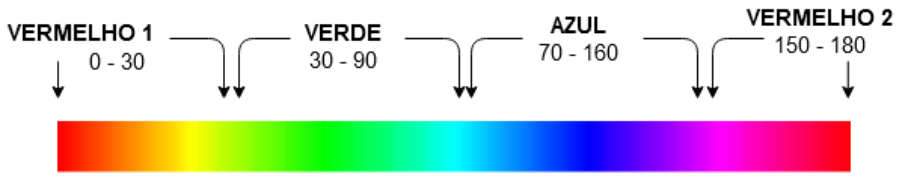
A etapa de encerramento é basicamente encerrar a transmissão, ou *streaming* de vídeo, e liberar a memória do processador utilizada pelos *buffers*.

### 4.3.2 Máscara de cores

Para que a aplicação de processamento de imagem fosse capaz de realizar os testes propostos, fez-se necessário a utilização de algoritmo capaz de filtrar as zonas da imagem que possuem a mesma coloração. E para se agruparem *pixeis* com coloração semelhante, a utilização de uma projeção HSV se mostra conveniente. Diferentemente da projeção RGB, onde cada uma das três cores básicas possui uma intensidade, o sistema HSV proporciona um único canal responsável pela chamada matiz, que caracteriza a cor do pixel.

Assim, ao se procurarem por regiões de *pixels* com LEDs de uma cor específica, se realiza uma conversão do *frame* em espaço RGB para o sistema HSV e então se realiza uma comparação para verificar se a componente H de cada um dos *pixels* pertence à um determinado intervalo de coloração, como por exemplo vermelho. Como a implementação da conversão para HSV foi realizada com a variação entre 0 e 180, os intervalos utilizados para filtragem são demonstrados na escala demonstrada na Figura 9.

**Figura 9:** *Indicação dos conjuntos de valores Hue utilizados para cada filtro.*




Fonte: Elaborado pelo autor.

Após esta comparação, uma máscara binária é gerada, com a mesma resolução do *frame* original, porém com um único canal cujo valor contido (verdadeiro ou falso) indica se o *pixel* do *frame* original está dentro do intervalo analisado.

### 4.3.3 Algoritmo para obtenção de posição dos LEDs

Para a calibração espacial, como chamada a etapa pré-testes responsável por atribuir coordenadas para cada uma das fontes luminosas, se faz necessário obter a posição de cada um dos LEDs. Como a filtragem de cores pode ser feita através do processo descrito

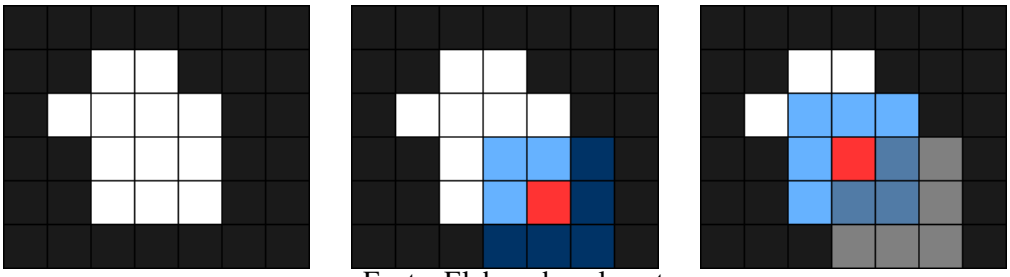
na subseção anterior, onde a saída é uma máscara (ou matriz verdade), foi implementado algoritmo para encontrar os LEDs a partir da máscara gerada.

Uma vez que a máscara obtida é uma matriz binária, o algoritmo trata de encontrar agrupamentos de *pixels* que possuem o valor não-nulo. Uma representação de um desses agrupamentos pode ser visto na Figura 19a. Para isso, o algoritmo verifica se todos os vizinhos de um pixel de referência possuem valor não-nulo. E isso é feito de forma recursiva, portanto, a verificação dos vizinhos de um pixel de referência (como na Figura 10b) irá chamar a verificação para cada um de seus vizinhos (Figura 10c) e assim sucessivamente, até que todos *pixels* em uma região sejam encontrados.

O algoritmo armazena a quantidade de *pixels* no *cluster*, para que se possa ter uma dimensão do tamanho do mesmo. E também salva as coordenadas espaciais de onde o *cluster* está localizado dentro do *frame*.

**Figura 10:** *Exemplo de cluster dentro de matriz de máscara.*

**(a)** *Exemplo de cluster em uma matriz de máscara.* **(b)** *Verificação de pixels vizinhos a partir de referência.* **(c)** *Verificação de pixels vizinhos a partir de outra verificação.*



Fonte: Elaborado pelo autor.

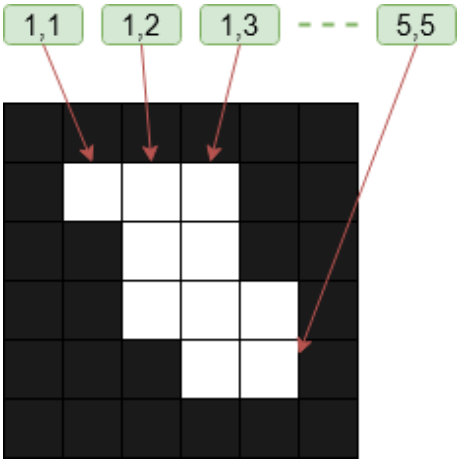
### 4.3.4 Gerando uma *Lookup Table*

Imaginando-se a utilização de uma câmera gravando em 60 frames por segundo, o período para recebimento de um novo frame é de aproximadamente $16ms$. Realizando testes iniciais com os algoritmos implementados, se verificou que para realizar a conversão, máscara e clusterização no frame inteiro, a aplicação despende uma quantia de tempo maior do que o recebimento de frames, no cenário descrito. Por isso, foi elaborado uma estrutura de *lookup tables* para as regiões onde os LEDs são encontrados.

A utilização de *lookup tables* economiza o tempo de processamento para encontrar os LEDs em cada *frame* recebido. Uma etapa de calibração irá encontrar os LEDs e criar uma tabela com as posições de interesse. As etapas de avaliação das intensidades luminosas podem realizar as operações de avaliação apenas nas regiões contidas nas tabelas. O ponto negativo é que os LEDs são considerados com posição fixa e qualquer alteração na posição dos LEDs, as coordenadas de medição serão inválidas e as medidas terão valores que não correspondem aos LEDs.

Primeiramente, os clusters, que representam a posição dos agrupamentos de interesse no frame, são descritos como caixas e quantidade de *pixels*. Ou seja, uma coordenada representando o vértice superior esquerdo e outra representando o vértice inferior direito. Como os LEDs são circulares, ao simplificar os clusters como caixas, alguns *pixels* que não pertencem à região do LED serão incluídos. Por isso, ao gerar uma *lookup table* para os *pixels* que devem ser avaliados, os pontos são criados a partir de uma geometria circular. A aresta da caixa com menor dimensão é utilizada como diâmetro do círculo e o centro do círculo coincide com o centro da caixa.

**Figura 11:** *Ilustração do funcionamento de uma lookup table pra um cluster.*




Fonte: Elaborado pelo autor.

A grande diminuição no número de pixels pôde ser validado com a criação de uma *lookup table* para cada LED. No caso em que a imagem inteira era processada, eram avaliados 921600 *pixels*. Para o caso testado a quantidade de *pixels* processados passou a ser menos de 3500.

## 4.4 Implementação dos testes

Com as ferramentas desenvolvidas, os testes propostos no presente trabalho puderam ser implementados de forma automática, no sentido de que necessitam apenas da sua execução e as etapas de calibração, leitura de valores e avaliação de resultados serão realizadas de forma autônoma.

O processamento dos dados e geração dos resultados é feito utilizando a linguagem Python, principalmente por ser uma linguagem interpretada. O lado negativo da utilização de Python é a possível necessidade de instalação de pacotes adicionais e um custo de processamento maior do que linguagens compiladas. Porém como a avaliação de resultados não precisa ser necessariamente realizada no sistema embarcado e não se tem uma necessidade de otimização dos algoritmos, a escolha se manteve.

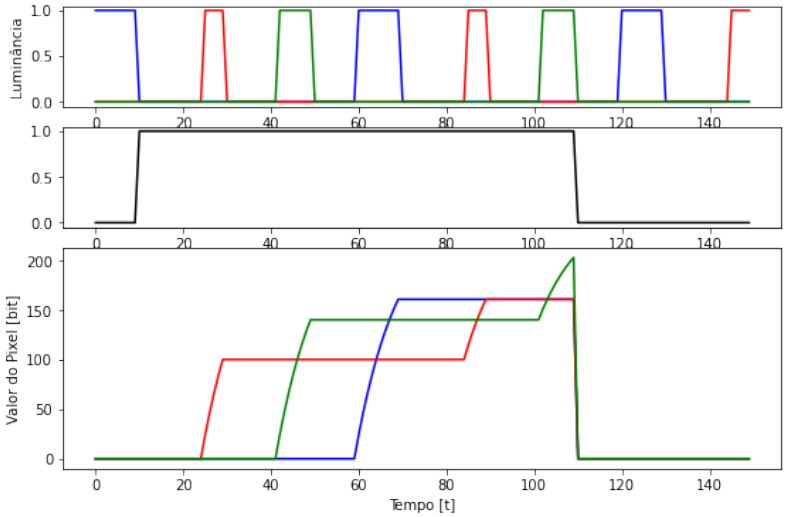
Então as ferramentas para os testes que foram desenvolvidas são: aplicação de captura e processamento compilada para o sistema embarcado, unidade de acionamento que se comunica através de UART, manipulação dos resultados através de script python. Para que fosse possível a integração de todas essas ferramentas, criou-se um script para o sistema operacional, que é capaz de utilizar todas estas ferramentas e assim realizou-se a sequencia de execução para cada um dos testes.

## 4.5 Simulação

Uma maneira de avaliar se os dados obtidos são válidos e razoáveis é através da simulação de dados. Por isso, foi desenvolvido um *script* que gera sinais virtuais de luminosidade dos LEDs e, conhecendo-se a função de relação entre incidência e valor do pixel, ao realizar a integração da luminosidade de cada um dos LEDs, simula-se o valor aproximado que será lido pelo processador. Esta simulação para um único ponto em

função do tempo pode ser vista através dos gráficos da Figura 12.

**Figura 12:** *Simulação da cor em um ponto, obturador e tempo de integração.*




Fonte: Elaborado pelo autor.

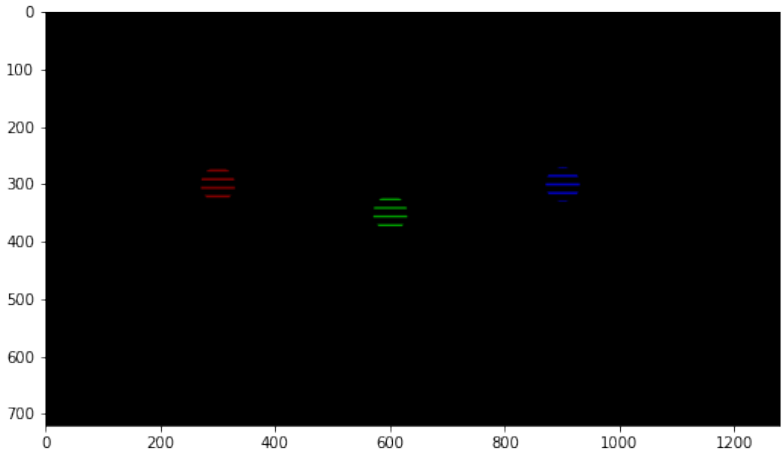
Uma forma de se entender o procedimento utilizado para tais cálculos é dividir cada os canais (R,G e B). Para cada de instante de tempo, cada LED apresentará uma incidência luminosa. Portanto, cada um desses instantes pode ser representado por um *frame* monocromático que representa a intensidade da cor no instante. Assim, um único *frame* pode ser representado em um plano cartesiano X,Y.

Utilizando um terceiro eixo que representa o tempo, é possível descrever a intensidade de um LED para todos os instantes do teste. O valor final lido para um pixel posicionado em $x, y$ durante um tempo de exposição estabelecido pode ser encontrado integrando a reta entre os pontos $x, y, t_{INICIAL}$ e $x, y, t_{FINAL}$.

Foi utilizado então o modelo matricial descrito para cada um dos três canais. Uma vez que a exposição à luz em sensores CMOS é feita linha a linha, foi adicionada ainda uma variável representando a diferença de tempo de leitura das linhas do sensor.

Com essas informações e com a posição e raio dos LEDs, é possível inferir como será o *frame* lido nas condições sendo testadas. Na Figura 13, é possível verificar um *frame* gerado pelo simulador.

**Figura 13:** *Simulação de um frame.*




Fonte: Elaborado pelo autor.

# 5 RESULTADOS

Utilizando as ferramentas e componentes desenvolvidos e relatados na seção anterior e seguindo a metodologia proposta na Seção 3, foram realizados testes para validação dos conceitos inicialmente supostos. Os resultados, bem como uma breve análise dos mesmos, estão descritos na presente seção.

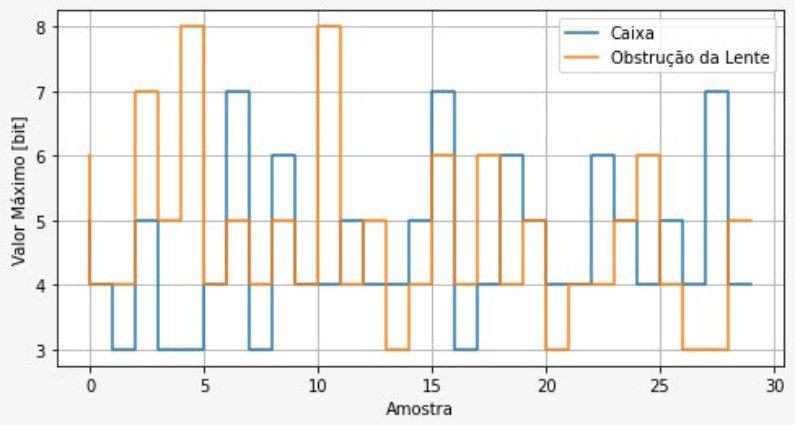
## 5.1 Análise da Caixa Escura

Antes da realização dos demais testes que procuram caracterizar o sistema de visão, faz-se necessária a análise da caixa escura, para validar que a mesma foi construída de forma a cumprir a função de não permitir fontes externas de luz que podem prejudicar os testes. A metodologia seguida é a descrita na Subseção 3.2.

Para que o tempo de captura de todos os *frames* necessários para o teste fosse de um segundo, um valor de $N = 30$ foi escolhido.

Como pode ser visto na Figura 14, o máximo valor obtido na caixa escura foi de 8 contagens. Portanto, o mínimo valor válido para os próximos testes será considerado de 8 contagens.

**Figura 14:** *Valores máximos lidos no teste de black current.*

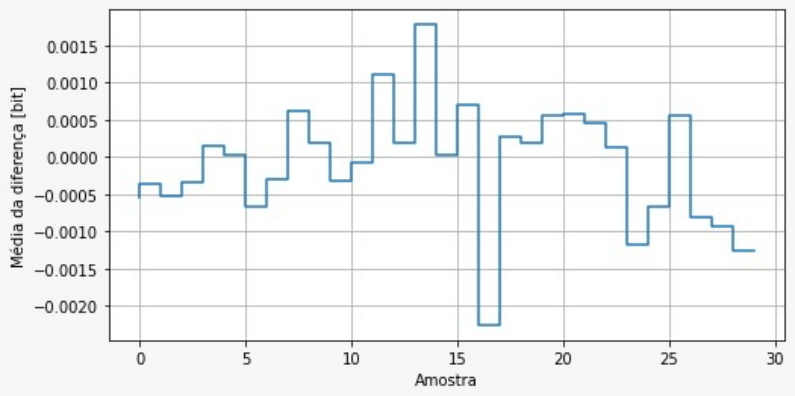



Fonte: Elaborado pelo autor.

O máximo incremento de luz desejado para cada um dos *pixels* é de uma contagem. Ou seja, acima desse valor será considerado que a caixa está permitindo a entrada de luz externa. Definindo o conjunto de dados como sendo o valor de todos os *pixels* em uma imagem, a diferença entre a média no caso da medição na caixa escura e a média da medição na referência não poderia ultrapassar uma contagem.

O incremento na média dos *frames* capturados em ambiente escuro e na referência foi de menos de uma contagem, como pode ser visto na Figura 15.

**Figura 15:** *Diferença entre valor médio da referência e valor médio da caixa escura.*




Fonte: Elaborado pelo autor.

Deve se observar que, por ser uma prova de conceito, os critérios de aceitação não exigiram alta precisão. E, portanto, para as condições previamente estabelecidas, a caixa pôde ser considerada válida para realização dos próximos testes.

## 5.2 Modelo de Relação

Para se criar um modelo que relaciona valor do *pixel* com a incidência luminosa, foi utilizada a metodologia descrita na Subseção 3.3.

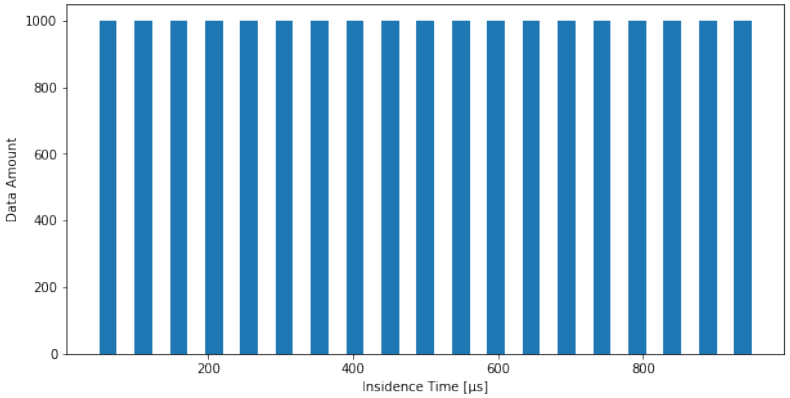
Como se deseja utilizar a variação do tempo em que o LED fica aceso como variável de controle, deseja-se que o tempo de incremento seja consideravelmente menor que o tempo de integração. Porém pela limitação do microcontrolador utilizado, as interrupções do sistema devem ser intervaladas de forma longa o suficiente para que a interrupção possa ser processada sem que outra interrupção seja disparada. Como o tempo mínimo para chamada de interrupção é de ordem de microssegundos, empiricamente foi encontrado um tempo mínimo para garantir regularidade do sistema em $50\mu s$.

Cada teste capturou 100 *frames* e foram realizados um total de 25 vezes. Os testes variaram o tempo de incidência de $1\mu s$ a $60\mu s$. Como discutido na subseção anterior, mesmo sem incidência de luz a leitura pode apresentar um erro, por isso o valor do *pixel* para o tempo 0 foi ignorado.

Após a coleta dos dados, os *frames* válidos foram selecionados. Em seguida, os dados foram extraídos e pôde-se verificar uma grande quantidade de pontos na zona de saturação. Os pontos na região de saturação foram descartados para a etapa de minimização.

Foi também realizado um emparelhamento dos dados para ser possível utilizar um algoritmo de mínimos quadrados sem que uma das regiões esteja melhor ajustada que as demais por uma questão de densidade de dados. No total foram usados 1000 pontos para cada variação de incidência de luz. O histograma dos dados balanceados pode ser visto na Figura 16.

**Figura 16:** *Histograma das amostras para o teste de frequência.*




Fonte: Elaborado pelo autor.

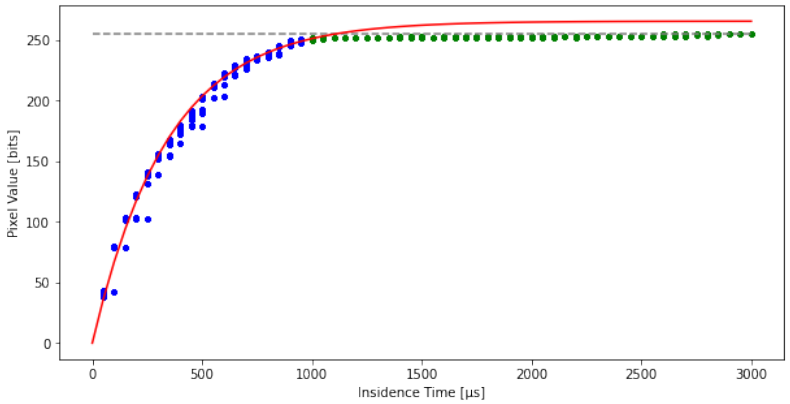
Por serem construídos com a estrutura CMOS, cada fotosensor pode ser analisado como um transistor. No caso de transistores CMOS, o *gate* é construído de forma a acumular cargas utilizando a energia potencial do campo elétrico, análogo a um capacitor. Considerando que a resposta do sistema tem esta característica de acúmulo de carga devida à capacitância nos fotosensores, a equação a ser minimizada será

$$V(t) = A(1 - e^{-\tau t}),$$

onde V é o valor do *pixel*, $\tau$ é a constante de tempo da curva de acúmulo e t é o período de incidência.

Utilizando a biblioteca *scipy* de python para o cálculo de mínimos quadrados, os valores obtidos para as constantes foram $A = 265,68$ e $\tau = 0.00292$. A curva ajustada pode ser vista em vermelho na Figura 17. Os pontos em azul representam os pontos utilizados para o ajuste e em verde se encontram os pontos descartados.

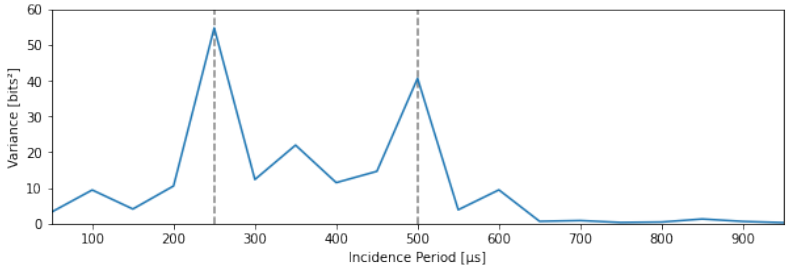
**Figura 17:** *Simulação de um frame.*




Fonte: Elaborado pelo autor.

Outro ponto relevante a ser analisado é a variância dos dados. Para cada intervalo de incidência, se calculou a variância para os dados utilizados. A Figura 18 apresenta essa variância. Assumindo que a frequência de captura é próxima de $30Hz$, a $5^a$ e $6^a$ harmônicas representam $515,62\mu s$ e $257,81\mu s$. Nota-se que os picos se encontram justamente próximos à esses valores, onde um efeito de *aliasing* é esperado para as amostras coletadas.

**Figura 18:** *Simulação de um frame.*




Fonte: Elaborado pelo autor.

## 5.3 Leitura de frequência

Utilizando-se a metodologia proposta na seção anterior, através dos LEDs da matriz, uma contagem na ordem de código *gray* é exibida. A câmera captura uma imagem que, processada, permite identificar qual código está sendo exibido. Conhecendo o código presente na imagem, é possível verificar o índice de tal código dentro da sequência do código *gray*.

Foram lidos no total $N = 45$ para realização do teste, ou seja, 45 códigos. Após receber as imagens e realizar a calibração da posição de cada um dos LEDs, foi possível recuperar a informação do índice do código sendo enviado. A informação contida nos LEDs foi considerada binária. O resultado verdadeiro ou falso para cada LED foi avaliado através de um valor de *threshold* ajustado em $T = 15$. Este valor foi escolhido por ser, aproximadamente, o dobro do mínimo valor válido (vide subseção anterior).

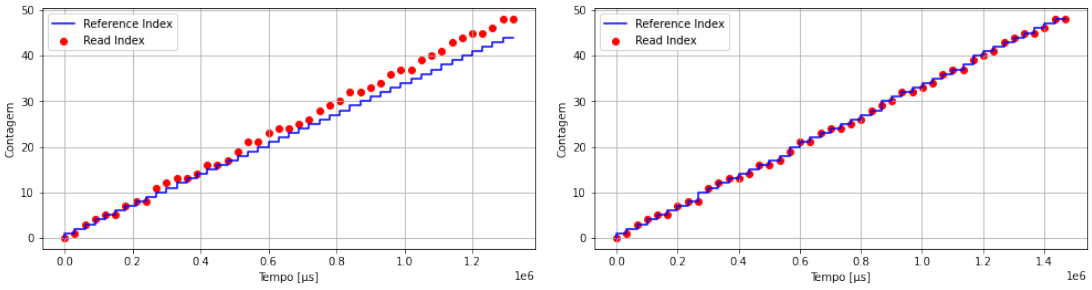
Sabendo que a contagem tem passo fixo e que a cada passo o valor é incrementado em 1, pode-se procurar uma relação entre os valores enviados e os lidos.

Neste caso, são conhecidos os índices lidos, os índices que foram emitidos e se conhece o passo fixo da função que controla os LEDs. Assim, em um caso onde se procure minimizar a curva entre o lido e o gerado, a única variável a ser ajustada é o tamanho do passo da captura para que os dados capturados coincidam com a função referência.

Para isso, foi implementado um laço de repetição com custo quadrático da diferença entre o emitido e o esperado. Por se tratar de uma função não linear, tal função teve de ser implementada para abranger o caso específico. O ajuste das funções pode ser visto na Figura 19b.

**Figura 19:** *Exemplo de cluster dentro de matriz de máscara.*

**(a)** *Dados lidos e contagem de referência.* **(b)** *Dados ajustados à contagem de referência.*




Fonte: Elaborado pelo autor.

Portanto, com um período de passo de referência $T_{STEP} = 30000\mu s$, o valor obtido iterativamente foi 33392 $\mu s$. Utilizando o passo de referência e dividindo em partes iguais para todos os índices, um erro intrinsecamente ligado ao tamanho do passo pode ser descrito como $E = T_{STEP}/N = 666,66\ \mu s$. Este resultado representa então uma leitura de $29.95 \pm 1.20\ Hz$. Em seu manual, por padrão a câmera informa uma taxa de amostragem de 30 FPS, ou seja, o resultado obtido se mostrou coerente com o esperado.

Cabe aqui uma observação que a unidade de *frames* por segundo é análoga à frequência de amostragem neste caso, optou-se apenas pela representação em $Hz$ por ser mais coerente num contexto de números decimais, diferentemente da utilização de valor FPS, comumente utilizado apenas com números inteiros.

## 5.4 Sincronismo entre câmeras

O teste de sincronismo de câmera se baseia também em extrair a informação de uma contagem através do processamento de imagens da matriz de LED , como apresentado na Seção 3.5.

A sequência utilizada foi $C = 2^i$, variando-se $i$ entre 0 e 15. Em binário, este procedimento pode ser visto como um *shift bit*.

É interessante que o tempo entre cada contagem seja o menor possível para garantir uma precisão maior na leitura do sincronismo. Porém, chaveando o LED muito rapidamente, tem-se uma menor incidência luminosa por LED. Isso faz com que o valor lido para um LED fique mais próximo do mínimo valor válido, que foi definido como máximo ruído encontrado. Assim, em condições com o chaveamento muito rápido a leitura do valor (ligado ou desligado) se torna indefinida, uma vez que se está avaliando a mesma grandeza que a quantidade de ruído presente.

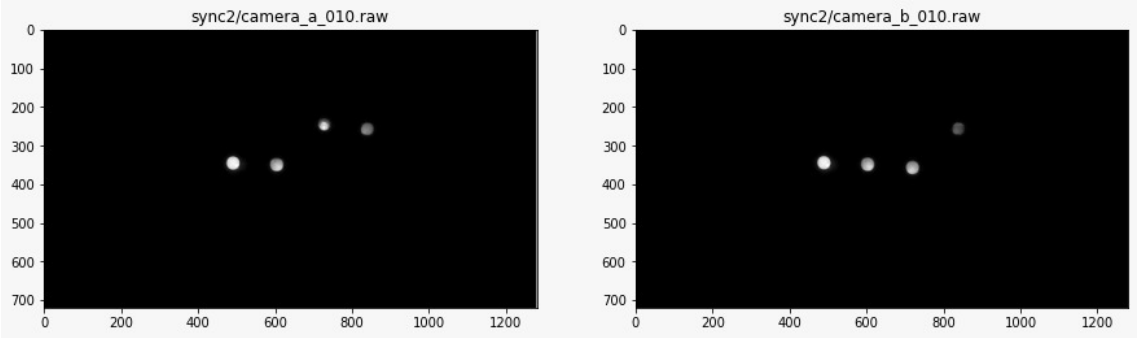
Por isso, optou-se por, em detrimento da precisão, realizar uma leitura com valores bem definidos, e o valor de passo entre amostras foi definido como $T_{STEP} = 1ms$.

Pelo fato da interface de câmeras da placa Raspberry não possuir entrada para duas câmeras em protocolo CSI, para a prova de conceito foi utilizada uma única câmera fazendo duas coletas diferentes. Para tal, o sistema configura a câmera e os LEDs e, sincronizando ambas, inicia tanto captura quando a contagem.

A diferença de tempo entre dois testes realizados se dá pelos atrasos de comunicação entre os componentes e ainda atrasos no sistema operacional que não são determinísticos e variam de execução para execução. A partir deste ponto, o texto irá relevar o fato de ser apenas uma câmera realizando as coletas de dados duas vezes e considerará duas câmeras (Câmera 1 e Câmera 2) por maior simplicidade de análise.

A diferença entre as contagens pode ser vista de na Figura 20. Cada imagem representa um dos dois grupos de teste. É possível verificar uma diferença de tempo entre as imagens.

**Figura 20:** *Canal vermelho filtrado para as imagens capturadas pela câmera.*



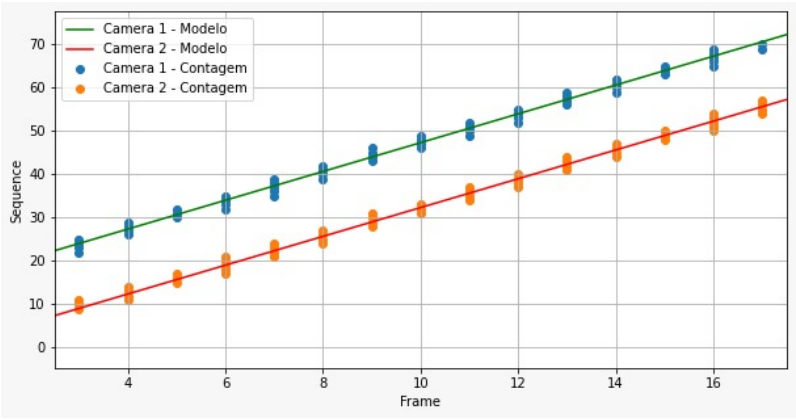

Fonte: Elaborado pelo autor.

É importante salientar que apenas um LED é acionado por vez pelo sistema e os demais permanecem desligados. Porém, através das imagens, quatro LEDs parecem estar acionados simultaneamente. Isso é explicado pelo tempo de integração que, por ser maior que o passo do *shift* ($T_{STEP}$), acaba integrando o valor de mais de uma contagem em um mesmo *frame*. Portanto, para estas imagens, o tempo de integração provavelmente é algo próximo de $4T_{STEP}$.

Extraindo a informação de qual a posição dos LEDs ligados nos *frames*, foi possível inferir o momento dentro do ciclo de *shift* que foi capturado. Porém como mais de um LED aparenta estar ligado em uma mesma imagem quando na verdade um único LED é aceso por contagem, não se tem certeza sobre o instante exato e sim sobre uma faixa de contagens capturados. Porém realizando o mesmo procedimento diversas vezes se obtém informação para avaliar a progressão dessa faixa como um todo.

Realizou-se então uma regressão linear através de mínimos quadrados para encontrar uma reta que represente bem a progressão das faixas de contagem capturadas nas amostras. O resultado pode ser visto no gráfico exposto na Figura 21.

**Figura 21:** *Simulação de um frame.*




Fonte: Elaborado pelo autor.

Para se amortizar o erro pela quantidade amostral, uma minimização que engloba todas as amostras de uma câmera foi utilizada.

A equação para reta $ax+b$ foi utilizada para minimização. Os resultados obtidos foram $a_0 = 3.328$, $a_1 = 3.309$, $b_0 = 14.01$ e $b_1 = 1.262$. Fazendo a consideração de

que as inclinações são iguais (o que deveria ocorrer na teoria, uma vez que a velocidade de incremento é a mesma nos dois casos), o *offset* entre a leitura das câmeras pode ser inferido pela constante $b$ da equação da reta. O valor de *offset* entre as capturas pode foi tomado então por $(b_0 - b_1) * T_{STEP}$, com a precisão máxima de $T_{STEP}$. Dessa maneira, $T_{OFFSET} = 12,748 \pm 1\ ms$.

O valor de *offset* representa a diferença temporal perante uma mesma referência visual. Ou seja, para cada instante capturado pela Câmera 1, após $T_{OFFSET}$ foi capturada uma imagem para a Câmera 2.

# 6 CONCLUSÃO

O presente trabalho teve a proposta de criar uma metodologia simples de avaliação das características e propriedades da captura de video realizada em um sistemas de visão embarcado. Além disso, através do levantamento de parâmetros do sensor, como no caso da relação entre valor digital e luminosidade, os testes realizados ainda trazem novas possibilidades para testes mais complexos e precisos, bem como implementações de sistemas de visão que se utilizam de tais relações.

Com a implementação dos testes através da prova de conceito, foi possível verificar que as métricas propostas foram obtidas, dentro das precisões especificadas nos resultados.

Como o teste de frequência de amostragem demonstrou uma menor variação do que a medição realizada diretamente no sistema operacional embarcado, que neste caso é estocástico, conclui-se que o mesmo é válido e satisfatório.

Diferente de outros testes de sincronismo de câmeras, onde se utilizam os sinais elétricos gerados pela câmera ou ainda as interrupções nos *drivers* de entrada de vídeo, o teste proposto e implementado utiliza os próprios *frames* para a medição de sincronismo. Neste caso, sincronismo pode ser definido como a proximidade entre os instantes capturados do ambiente por diferentes câmeras e, por trazer resultados conclusivos quanto à ordem de grandeza desta proximidade, considerou-se o teste satisfatório.

Foi possível verificar também que um modelo matemático que correlaciona incidência luminosa e valor do pixel pode ser gerado através do procedimento proposto.

Por ser útil em diferentes contextos do desenvolvimento de sistemas de visão, a precisão obtida na implementação prática pode ou não ser satisfatórios e/ou relevantes, dependendo da aplicação e necessidade dos testes.

Além do inicialmente proposto, mostrou-se útil a utilização da simulação de *frames* capturados, como referência teórica para os resultados obtidos.

De modo geral, para o desenvolvimento de algoritmos que utilizam as cores vindas do sensor e temporizações disponibilizadas pelo sistema operacional ou mesmo providas pela câmera, considera-se que os testes podem trazer robustez ao possibilitar uma análise crítica das condições de operação do sistema. Para o desenvolvimento de sensores e câmeras, a metodologia pode ser útil para possibilitar informações técnicas mais amplas sobre a câmera, porém a precisão obtida pela prova de conceito possivelmente será insatisfatória, uma vez que a frequência de *clock* e a sensibilidade elétrica dos sensores normalmente estão em ordens de grandeza menores do que a precisão obtida. Se tratando do desenvolvimento de uma solução embarcada, as metodologias e testes trazem uma possibilidade de avaliação simplificada e prática para as métricas abrangidas, sem a necessidade de medições indiretas para avaliar o desempenho do sistema.

Sendo a metodologia em si o produto final do trabalho, notam-se já possíveis aspectos a serem aprofundados e detalhados para futuras especificações de novos testes. Além

disso, do ponto de vista prático algumas escolhas de componentes e ferramentas podem ser aprimoradas para futuras implementações.

# APÊNDICE A - OUTROS EDITORES

Os alunos podem utilizar qualquer editor de texto, devendo apenas tomar o cuidado de atêr-se as normas ABNT tal qual apresentado neste documento (e.g. sistema LaTeX).

# REFERÊNCIAS


AUTOPILOTREVIEW. *LiDAR vs. Cameras for Self Driving Cars – What's Best?* [S.l.: s.n.]. https://www.autopilotreview.com/lidar-vs-cameras-self-driving-cars/. Accessed: 17-11-2020.

BAYER, B. E. *Color imaging array*. [S.l.]: Google Patents, jul. 1976. US Patent 3,971,065.

BOYLE, W. S.; SMITH, G. E. Charge coupled semiconductor devices. *The Bell System Technical Journal*, v. 49, n. 4, p. 587–593, 1970.

C-CAM. *Frame-rate Calculation*. [S.l.: s.n.]. http://www.c-cam.be/doc/Archive/FrameRates.pdf. Accessed: 28-10-2020.

CHENG, H.-D. et al. Color image segmentation: Advances and prospects. *Pattern Recognition*, p. 2259–2281, 2001.

CRESSLER, J. D. *Silicon Earth: Introduction to Microelectronics and Nanotechnology*. [S.l.]: CRC Press, 2016.

DALLA BETTA, G.-F. *Advances in photodiodes*. [S.l.]: BoD–Books on Demand, 2011.

DE LA ESCALERA, A. et al. Visual sign information extraction and identification by deformable models for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, v. 5, n. 2, p. 57–68, 2004. DOI: 10.1109/TITS.2004.828173.

DENBAARS, S. Gallium-nitride-based materials for blue to ultraviolet optoelectronics devices. *Proceedings of the IEEE*, IEEE, v. 85, n. 11, p. 1740–1749, 1997.

DORAN, R. W. The Gray Code. v. 13, n. 11, p. 1573–1597, 28 nov. 2007. |http://www.jucs.org/jucs$_1$3$_1$1/$the_g$r

FAIRCHILD, M. D. *Color Appearance Models*. [S.l.]: Addison-Wesley, 2005.

HILLEL, A. BAR et al. Recent progress in road and lane detection: A survey. *Machine Vision and Applications*, v. 25, abr. 2014. DOI: 10.1007/s00138-011-0404-2.

HIRSCH, R. *Exploring Colour Photography: A Complete Guide*. [S.l.]: Laurence King Publishing, 2004.

HSU, T. et al. The Mechanism and Evaluation of Hot-Carrier-Induced Performance Degradation in 0.18- *mu* m CMOS Image Sensor. *Electron Device Letters, IEEE*, v. 25, p. 427–429, jul. 2004. DOI: 10.1109/LED.2004.829000.

KOZLOWSKI, L. J.; LUO, J.; TOMASINI, A. *Performance limits in visible and infrared imager sensors*. [S.l.: s.n.], 1999. p. 867–870.

LIN, H.-Y. et al. Lane departure and front collision warning using a single camera. In: p. 64–69. ISBN 978-1-4673-5083-9. DOI: 10.1109/ISPACS.2012.6473454.

MAKSIMOVIĆ, M. et al. Raspberry Pi as Internet of things hardware: performances and constraints. *design issues*, v. 3, n. 8, 2014.

MALLER, J. RGB and YUV Color. *FXScript Reference*, 2003.

SOCIETY FOR IMAGING SCIENCE e TECHNOLOGY, 1, [s.l.]. *The CIECAM02 color appearance model*. [S.l.: s.n.], 2002. v. 2002. p. 23–27.

MURATA, H. *Light emitting diode lamp*. [S.l.]: Google Patents, jun. 1990. US Patent 4,935,665.

NAKAMURA, J. *Image sensors and signal processing for digital still cameras*. [S.l.]: CRC press, 2017.

SCHIMEK, M. H. *Video for Linux Two API Specification*. [S.l.: s.n.]. `https://www.linuxtv.org/downloads/legacy/video4linux/API/V4L2_API/spec-single/v4l2.html`. Accessed: 17-11-2020.

SCHMIDT, E. A. et al. Camera-Monitor Systems as a Replacement for Exterior Mirrors in Cars and Trucks. In: *Handbook of Camera Monitor Systems: The Automotive Mirror-Replacement Technology based on ISO 16505*. Edição: Anestis Terzis. Cham: Springer International Publishing, 2016. p. 369–435.

SCHUSTER, M.; STRULL, G. A monolithic mosaic of photon sensors for solid-state imaging applications. *IEEE Transactions on Electron Devices*, IEEE, n. 12, p. 907–912, 1966.

[S.l.]. *A low dark current stacked CMOS-APS for charged particle imaging*. [S.l.: s.n.], 2001. p. 24.2.1–24.2.4. DOI: `10.1109/IEDM.2001.979566`.

THORSETH, A. Characterization, modeling, and optimization of light-emitting diode systems. *Department of Photonics Engineering*, p. 135, 2011.

ULLMAN, S.; BRENNER, S. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, v. 203, n. 1153, p. 405–426, 1979. DOI: `10.1098/rspb.1979.0006`.

UPTON, E.; HALFACREE, G. *Raspberry Pi user guide*. [S.l.]: John Wiley & Sons, 2014.

WANG, G. et al. CMOS video cameras. *Euro ASIC'91*, p. 100–101, 1991.