

# Identificação de frequência transmitida através de dispositivos que utilizam VLC para aplicação em localização *indoor*<sup>1</sup>

Daniele de Sá Freitas, Gustavo Henrique Santana de Souza, Ney Corrêa da Silva Neto & Pedro Henrique Bernardes Almeida

**Abstract**—This document has the purpose of presenting a proposal to identify frequency transmitted through devices that uses VLC (Visible Light Communication) to apply in *indoor* location systems.

**Index Terms**—Convey information, VLC, location, *Indoor* location.

**Resumo**—Este trabalho tem como finalidade apresentar uma proposta de identificação de frequência transmitida através de dispositivos que utilizam a comunicação VLC (*Visible Light Communication*) para aplicação em sistemas de localização *indoor*, ou seja, localização em ambiente interno.

**Palavras chave**—Transmitir informação, VLC, localização, localização *indoor*.

## I. INTRODUÇÃO

A tecnologia surgiu para auxiliar a vida da sociedade, e com o passar do tempo, é possível ver a agilidade que foi concebida aos seus usuários. Com o objetivo de combater diversos tipos de problemas ou incômodos em muitas áreas existentes uma das aplicações da tecnologia utilizada é o GPS (*Global Positioning System*), que surgiu como solução para a problematização de se localizar em lugares desconhecidos.

Apesar de ser uma tecnologia recente, o GPS é um serviço de localização eficiente que permite aos seus usuários encontrar caminhos de um lugar para o outro, obtendo informações sobre as imediações e facilitando o deslocamento para qualquer comércio, escritório, hospital, pontos turísticos, cidades ou outros locais. É uma forma de navegação que se concretizou através dos aplicativos móveis, e trouxe simplicidade, praticidade e acessibilidade àqueles que o utilizam [1], tornando-se uma tecnologia indispensável nos dias atuais.

Contudo, mesmo depois de passar por muitas mudanças para que a tecnologia GPS se tornasse mais precisa e confiável, essa só é viável quando utilizada em localização *outdoor*, ou seja, em ambientes externos [2].

Foi então que se iniciou o estudo da possibilidade de

desenvolvimento de uma tecnologia que auxiliasse na localização *indoor*, com o objetivo de orientar a locomoção das pessoas em ambientes internos, como supermercados, edifícios, faculdades, aeroportos, shopping centers e outros. Isto é, em qualquer local fechado, e que pudesse trazer mais conforto e agilidade para o dia-a-dia das pessoas.

A localização *indoor* pode ser apresentada através de várias aplicações tecnológicas existentes, porém o foco deste trabalho é desenvolver uma solução para identificação de frequências transmitidas por dispositivos que utilizando uma técnica de comunicação conhecida como VLC (sigla em inglês para *Visible Light Communication*). Dentro dessa perspectiva, a finalidade do trabalho é mostrar a implementação de um sistema que transmita dados através da luz, e os envie para um aplicativo móvel instalado no smartphone dos usuários que identifica a frequência transmitida e associa à localização correspondente. Esta solução tem potencial para atingir grande parte da população, visto que os smartphones estão presentes nas mãos de 62% dos brasileiros [3].

Neste trabalho, além de abordar outros estudos relacionados e projetos desenvolvidos como possíveis soluções de localização *indoor*, também serão abordadas as tecnologias e ferramentas utilizadas para a criação do mesmo, bem como detalhes importantes do processo de estudo, desenvolvimento e funcionamento. O tópico II trata de assuntos que mostram o princípio de funcionamento de algumas soluções já encontradas, e que motivaram o estudo e a pesquisa para criação de todo sistema proposto. Em sequência, no tópico III, são descritas todas as ferramentas e tecnologias utilizadas para desenvolver o trabalho e o tópico IV explica todo o seu funcionamento. E por fim, no tópico V são apresentadas as conclusões encontradas no desenvolvimento da solução proposta.

## II. TRABALHOS RELACIONADOS

Já existem trabalhos e pesquisas que mostram o desenvolvimento da tecnologia auxiliando a localização *indoor* de diversas formas e para diversos objetivos. Uma das pesquisas que ajudou no progresso deste trabalho, foi a [4] onde é estudado tecnologias que podem fornecer posicionamentos com alta precisão dentro de locais, e além disso, possam ser implementadas em dispositivos móveis que possuem recursos limitados. O autor apresenta a técnica de

<sup>1</sup> Trabalho de Conclusão de Curso apresentado ao Instituto Nacional de Telecomunicações como parte dos requisitos para obtenção do Grau de Bacharel em Engenharia da Computação. Orientador: Prof. Evandro. Trabalho aprovado em xx/xx/2017.

posicionamento *Visible Light Positioning* (VLP) que utiliza da tecnologia VLC para posicionamento *indoor*, onde o VLP depende da comunicação via luz (VLC) para transmitir as informações de localização para os receptores, que utilizam dos sensores de luz ou da câmera para receber essas informações. Segundo [4], este método já teria mais capacidade para atingir uma precisão maior do que se utilizasse a tecnologia da rede sem fio Wi-Fi [4].

Além dessas características, em [5] afirma-se que o VLC tem capacidade para superar a maioria dos obstáculos que outras tecnologias, como o GPS, teriam para desenvolver a mesma solução, e ainda que possui um consumo de energia mais eficiente. Os autores propõem no trabalho o desenvolvimento de um algoritmo que proporciona um sistema preciso de localização interno usando LEDs, que são responsáveis por transmitir as coordenadas, um sensor de imagem, que recebe esse sinal transmitido pelo LED e um acelerômetro, que mede a velocidade do movimento [5].

No mercado já existem alguns trabalhos, como aplicativos e produtos, que já foram lançados com o objetivo de solucionar o problema de localização *indoor*. Dentre eles encontram-se:

#### A. Indoor Maps

É um aplicativo que visa auxiliar a localização dentro de ambientes internos e é o pioneiro na tecnologia de Sistemas de Posicionamento *Indoor* (IPS) no Brasil. Foi criado pela empresa EKØdesign&interactive em 2013, e utilizou, dentre várias tecnologias, a da empresa francesa POLE STAR, que é pioneira mundial em IPS. O aplicativo pode ser instalado nas plataformas iOS e Android, possui uma precisão de 2 a 5 metros e permite localização em múltiplos andares sabendo informar ao usuário quando ocorre mudança de piso. Além disso, é possível inserir no mapa pontos de interesse, indicação de rotas e serviços de emergências [6]. A Figura 1 ilustra o produto descrito.

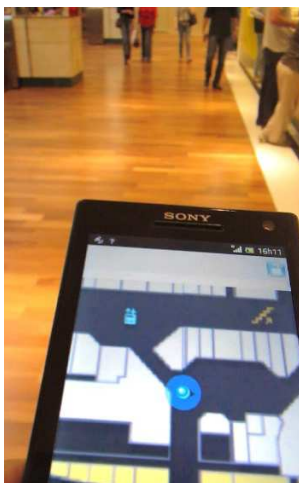


Figura 1 – Aplicativo *Indoor Maps*

#### B. Lumicast

É uma tecnologia da Qualcomm criada em abril de 2016, e que permite aos clientes uma experiência eficiente de posicionamento *indoor*. Utiliza as lâmpadas LED para

transmitir os sinais com a tecnologia VLC para um aplicativo instalado no smartphone dos clientes, que através da câmera consegue captar o sinal transmitido. Ele possui uma precisão de centímetros e a vantagem de utilizar da própria infraestrutura de iluminação do local [7]. A Figura 2 ilustra o aplicativo Lumicast.

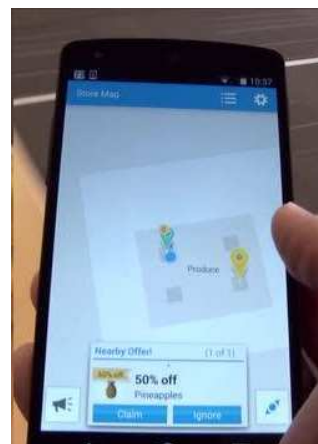


Figura 2 – Aplicativo Lumicast

#### C. Sistema de Posicionamento Indoor da Philips

Este sistema da Philips possui o mesmo princípio do produto da Qualcomm citado anteriormente. A Philips incorporou nas lâmpadas LED a tecnologia VLC, trazendo duas grandes vantagens. Primeiro, possibilitou a economia de energia para o proprietário em relação aos antigos sistemas de iluminação que utilizavam lâmpadas incandescentes ou fluorescentes, e segundo, viabilizou a tecnologia sem ter que interferir na estrutura física do local. O sistema conta com um SDK (da sigla em inglês, *Software Development Kit*) para iOS e Android e serviço em nuvem, que são usados para incorporar a capacidade de posicionamento aos aplicativos móveis [1]. A Figura 3 mostra o aplicativo móvel do sistema de posicionamento *indoor* da Philips.

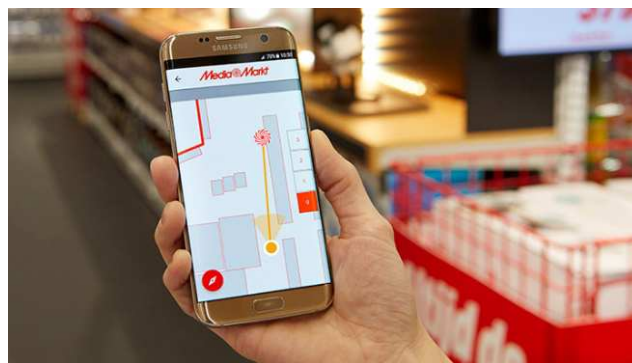


Figura 3 – Aplicativo Android do sistema de localização *indoor* da Philips

### III. FERRAMENTAS E TECNOLOGIAS UTILIZADAS

Neste tópico serão abordadas as ferramentas e tecnologias utilizadas no desenvolvimento do trabalho.

## A. Software

### 1) Android

O Android é um sistema operacional *open source*, com Kernel Linux, atualmente mantido pela Google. Ele é um dos sistemas operacionais mais utilizados no mundo, principalmente em *smartphones*, *tablets*, *tv's* e dispositivos *wearables* [8]. Atualmente encontra-se na versão 8.0 [9] e a sua versão mínima necessária para o funcionamento do projeto é a 5.0.

### 2) Arduino IDE

A plataforma Arduino, descrita no item B.1), fornece um ambiente de desenvolvimento integrado, o Arduino IDE (da sigla em inglês, *Integrated Development Enviroment*). O Arduino IDE é um software de código aberto e gratuito para os principais sistemas operacionais como Windows e Linux. É uma aplicação multiplataforma, e foi escrita na linguagem de programação Java.

As linguagens de programação que podem ser utilizadas no Arduino IDE são a linguagem C, e a linguagem C++ [10].

O Arduino IDE foi utilizado para desenvolver o *firmware* do microcontrolador empregado na solução.

### 3) Eclipse

A IDE Eclipse é um software livre, mantida pela organização Eclipse Foundation, que possibilita a seus usuários criarem aplicações Java, C, C++, entre outras.

Ela possui um amplo suporte de *plug-ins* disponíveis que auxiliam os programadores em diferentes necessidades [11].

A versão da IDE utilizada foi o Eclipse Oxygen e o objetivo foi o pré-desenvolvimento de módulos do projeto, como o tratamento da imagem capturada pela câmera do *smartphone*.

### 4) Android Studio

O Android Studio é uma IDE gratuita criada pela Google, para o desenvolvimento de aplicativos para a plataforma Android. Foi anunciado na Google I/O de 2013, porém ganhou a versão 1.0 somente um ano após o anúncio, e tornou-se a principal ferramenta para desenvolvimento de aplicativos Android [12][13].

O Android Studio é utilizado no projeto para o desenvolvimento do aplicativo *mobile* e a versão da ferramenta utilizada é a 2.3.3.

## B. Hardware

### 1) Arduino

O Arduino foi criado em 2005 no *Interaction Design Institute*, em Ivrea na Itália, sendo uma plataforma eletrônica de código aberto que se destaca pela sua facilidade e simplicidade de aprendizado, atingindo desde desenvolvedores iniciantes até os mais experientes.

As placas Arduino consistem basicamente em entradas e saídas controladas por um microcontrolador. Os

tratamentos de entradas e as informações necessárias para alcançar as saídas desejadas são codificadas pelo Arduino IDE, citado no tópico A.2) [14].

Existem no mercado muitos tipos de placas Arduino. Porém neste projeto foi usado o modelo Arduino Uno R3. Suas principais características são: tensão de operação de 5V, 14 portas de I/O, microcontrolador ATmega328P que opera em 16Mhz com 32 KB de memória flash, 2 KB de SRAM e 1 KB de EEPROM [15].

O objetivo da utilização do Arduino é gerar a frequência de oscilação do LED.

A Figura 4 apresenta a placa Arduino Uno R3



Figura 4 – Placa Arduino Uno R3.

### 2) Transistor

O transistor é um componente eletrônico que foi o principal responsável pela revolução da eletrônica na década de 1960. São utilizados principalmente como amplificadores e chaveadores de sinais elétricos [16].

No projeto é utilizado o transistor TIP 120 no modo chaveador. Este transistor suporta até 60V e 5A em seu coletor.

A função do TIP 120 é de chavear a tensão da saída do Arduino fazendo o LED acender e apagar.

A Figura 5 apresenta o transistor TIP 120.



Figura 5 – Transistor TIP 120.

### 3) LED

O LED é um diodo semiconductor emissor de luz capaz de emitir luzes em diversas cores [17].

No projeto é utilizado um LED branco de alto brilho de 10W de potência com tensão de alimentação de 10V, e consumo de corrente de 1A.

A função do LED neste projeto é transmitir a informação de frequência modulada pela saída do Arduino.

A Figura 6 ilustra um LED branco de alto brilho.



Figura 6 – LED branco de potência 10W.

#### 4) Câmera

A câmera do *smartphone* é o dispositivo de *hardware* utilizado no projeto para captar a informação transmitida pelo LED através da luz.

### IV. DESENVOLVIMENTO

A proposta deste trabalho é oferecer uma solução para a identificação de dispositivos transmissores que utilizam a tecnologia VLC com foco na aplicação em sistemas de localização *indoor*. Para tal, é necessário que o mesmo reconheça pontos de referência em uma imagem, processe e identifique a informação correspondente a esse dispositivo. Neste processo modula-se a luz emitida por um LED em uma frequência fixa. A câmera de um *smartphone*, após ter alguns de seus parâmetros fotográficos alterados, captura a imagem da fonte de luz que é então tratada por uma série de algoritmos com o objetivo de decodificar a frequência transmitida e associá-la a informação de localização correspondente. Os detalhes desse processo estão descritos nos tópicos a seguir:

#### A. Transmissor

A luz do LED é modulada com um *duty-cycle* de 50% e sua frequência é definida em uma função que está sendo executada no Arduino, onde o intervalo de tempo de mudança de estado é definido em milissegundos.

A Figura 7 mostra a função geradora de frequência implementada no Arduino.

```
const long interval = 2;
unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    if (ledState == LOW) {
        ledState = HIGH;
    } else {
        ledState = LOW;
    }

    digitalWrite(ledPin, ledState);
}
```

Figura 7– Função geradora de frequência.

Para um melhor resultado foi utilizado um LED de 10W com uma luminosidade bem alta, que varia entre 800 e 900 lúmens. Lúmen é unidade de medida de fluxo luminoso [18]. Com essa alta luminosidade do LED é possível observar sua frequência de oscilação até pelo reflexo em paredes, mesas e etc.

O TIP 120 está sendo utilizado para fazer o chaveamento da tensão de alimentação do LED fazendo ele acender e apagar de acordo com a frequência enviada pelo Arduino.

O LED é alimentado por uma fonte de 12V, seu polo positivo está em série com um resistor de 2,2 ohms e 3W e o polo negativo está conectado ao coletor do TIP 120 como ilustrado pela Figura 8.

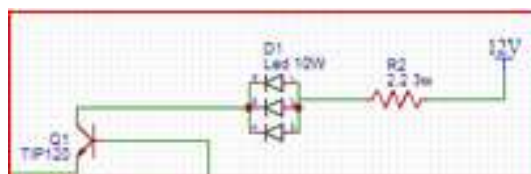


Figura 8 – Circuito de acionamento do LED.

O emissor do TIP está conectado ao terra do Arduino e ao terra da fonte. A base do TIP 120 está em série com um resistor de 1000 ohms e com a saída 4 do Arduino. A Figura 9 ilustra o circuito completo de acionamento do TIP 120.

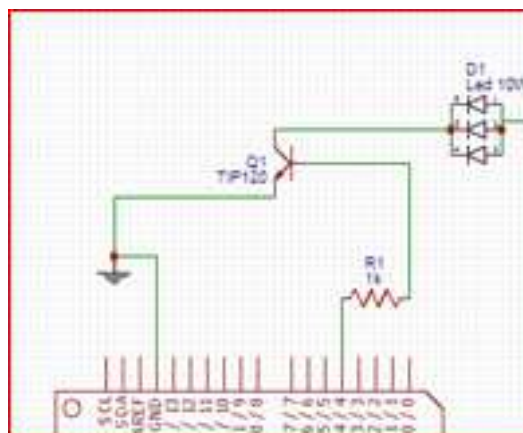


Figura 9 – Circuito do TIP 120.



O esquema de ligação do LED, TIP 120 e Arduino é ilustrado pela Figura 10 a seguir:

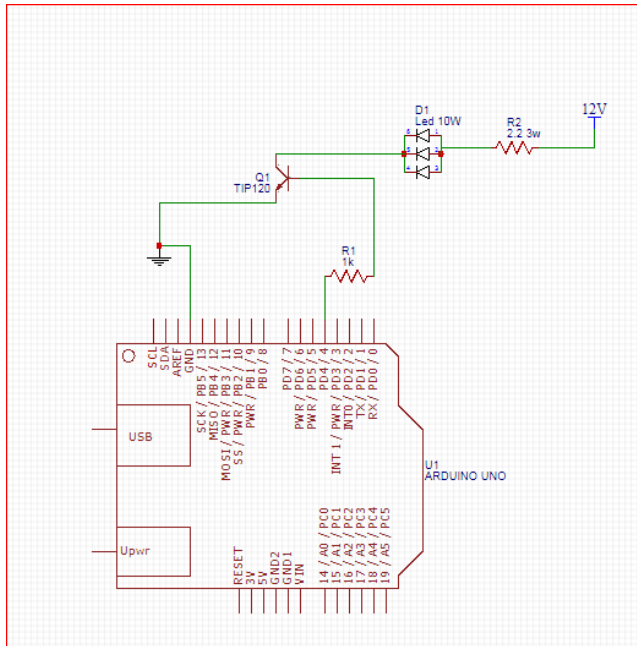


Figura 10– Circuito de acionamento do LED

Para a oscilação do LED está sendo usada a faixa de frequência de 500hz à 1khz. Não foram utilizadas frequências abaixo de 200hz pois essas causam efeito de cintilação ao olho humano [19].

#### B. Câmera

No momento que uma foto é tirada através da câmera de um *smartphone*, um sensor digital CMOS (*complementary metal-oxide semiconductor*) captura uma ou mais colunas de *pixels* (*pixel* é o menor elemento de uma imagem [20]), mas lê somente uma coluna por vez, percorrendo a imagem a uma taxa de varredura fixa para compor o resultado final. Quando então a luz que foi modulada em alta frequência e emitida pelo LED é capturada pela câmera, o resultado é um efeito de bandas na imagem, no qual algumas colunas de *pixels* capturam o LED aceso e outras o LED apagado. Este efeito não é visível a olho nu nem em uma imagem estática capturada com um tempo de exposição na configuração automática. Entretanto, o efeito de bandas é perfeitamente visível quando a imagem é capturada com um tempo de exposição curto, como pode ser visto nas Figuras 11a e 11b.

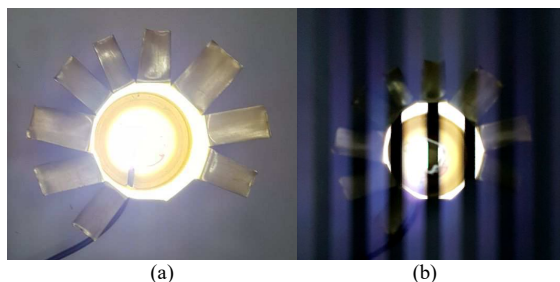


Figura 11 – Foto tirada com tempo de exposição automático (a) e com um tempo de exposição de 1/2000 segundos (b).

As câmeras exportam muitas propriedades que afetam a forma como elas capturam imagens. As duas principais e mais importantes propriedades para a execução deste projeto são o tempo de exposição (*shutter time*) e o ISO (*film speed*).

##### 1) Tempo de Exposição

O tempo de exposição, ou *shutter time*, representa o período de tempo que o obturador da câmera fica aberto expondo o sensor à luz e é um dos três pilares da fotografia, juntamente com o ISO e a abertura do obturador [21]. O objetivo da manipulação do tempo de exposição é obter a maior diferenciação possível entre as faixas capturadas com o LED aceso e as faixas capturadas com o LED apagado. Após a realização de alguns testes foi observado que independente do ISO, o melhor desempenho é encontrado utilizando um tempo de exposição bastante curto. Isso porque o raio de luz vindo do transmissor é intenso e requer um pequeno espaço de tempo para que o *pixel* seja saturado, ou seja, atinja seu valor máximo de luminância.

##### 2) ISO

O ISO, ou *film speed*, determina a sensibilidade do sensor da câmera. Ele regula quantos *fótons* são necessários para saturar um *pixel*. Um ISO alto aumenta a sensibilidade do sensor e faz com que os *pixels* da imagem saturam com menos *fótons*. Isso pode ser vantajoso caso o transmissor esteja distante ou possuir uma baixa potência de transmissão, já que melhorará o contraste da imagem. Porém, um ISO alto também pode amplificar reflexos indesejáveis e atrapalhar o processamento da imagem. A Figura 12 ilustra a diferença entre fotos obtidas com valores de ISO distintos. O projeto utiliza um ISO baixo na captura das imagens devido à proximidade e intensidade das fontes de luz presentes no ambiente de teste.

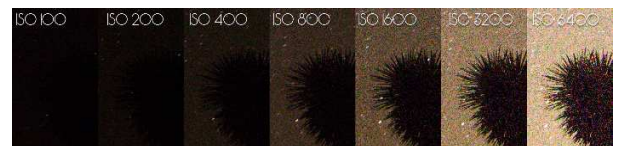


Figura 12 – Diferença entre fotos obtidas com diferentes valores de ISO. Os valores de ISO com que as imagens foram capturadas crescem da direita para a esquerda. [22]

#### C. Algoritmos

A imagem capturada pela câmera é manipulada por uma série de algoritmos até chegar ao objetivo final do processo, que é a obtenção da frequência de oscilação do LED. Os tópicos a seguir descrevem cada um destes algoritmos bem como sua finalidade no processo.

##### 1) Gray Scale

Antes de todo o processo, a imagem é convertida para escalas de cinza, pois a informação de cor na imagem não se faz necessária no processo. O algoritmo manipula cada *pixel* da imagem de modo que a informação de cor, chamada de crominância, seja removida restando apenas a

informação de luz, luminância, de cada *pixel*. A Figura 13 ilustra a conversão da imagem RGB para escala de cinza.



Figura 13 – Conversão de uma Imagem RGB para escalas de Cinza.

## 2) Gaussian Blur

O *Gaussian Blur* é um filtro utilizado para criar uma versão de uma imagem suavemente borrada da versão original [23]. Neste trabalho o algoritmo tem como objetivo desfocar a imagem de modo que fique explícito a região em que se concentra a fonte de luz. Os próximos algoritmos tratam a imagem a partir desta região de modo que todo o ruído contido em demais posições da imagem não interfira no resultado final do processo. A Figura 14 ilustra o resultado após o filtro *Gaussian Blur* ser aplicado à imagem.



Figura 14 – Imagem em escala de cinza após o filtro *Gaussian Blur*.

## 3) OTSU

OTSU é um algoritmo de limiarização proposto por Nobouyuki Otsu [24]. Este algoritmo é utilizado para detectar automaticamente um limiar de separação dos *pixels* da imagem, reduzindo a imagem a uma imagem binária, portanto, uma imagem representada por apenas dois valores de *pixels* distintos. Após o cálculo do limiar ótimo que separa as duas classes de *pixel* definidas como *pixel* de primeiro plano e *pixel* de fundo, a imagem é convertida de modo que a fonte de luz na imagem fique claramente visível e representado como *pixel* de primeiro plano. A Figura 15 ilustra o procedimento OTSU aplicado a imagem.

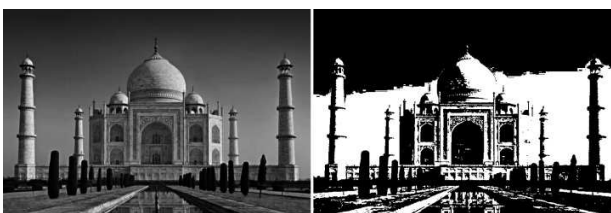


Figura 15 – Imagem em escalas de cinza convertida para uma imagem binária através do algoritmo OTSU.

## 4) Canny Edge

O algoritmo de detecção de bordas *Canny Edge*, tem como função detectar bordas em uma imagem. Neste trabalho o *Canny Edge* é utilizado para que seja definida a região onde se concentra a fonte de luz, e também para deixar explícita na imagem as faixas de *pixels* capturados com o LED aceso e as faixas de *pixels* capturados com o LED apagado. A Figura 16 mostra o resultado obtido no processamento da imagem utilizando o algoritmo *Canny Edge*.

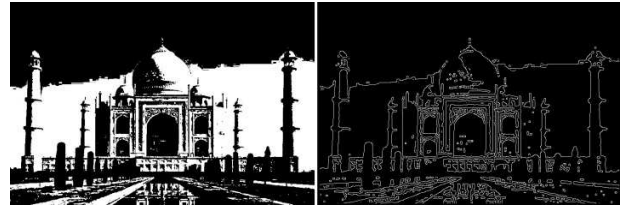


Figura 16 – Contornos de uma imagem binária obtidos pelo *Canny Edge*.

## 5) Distância Horizontal entre Pixels Brancos vizinhos

A última etapa do processamento da imagem é calcular a distância em *pixels* entre uma faixa e outra da imagem, e isso é feito através do algoritmo que calcula a distância horizontal entre *pixels* brancos vizinhos. O algoritmo consiste em traçar linhas horizontais em cada linha da imagem, onde cada *pixel* branco interceptado corresponde a uma faixa capturada pela câmera do smartphone devido a oscilação do LED, e cada *pixel* preto encontrado entre dois *pixels* brancos corresponde à distância em *pixel* entre uma faixa e outra. Isto é feito repetidamente em várias linhas da imagem, caso haja alguma distância diferente, é realizado o cálculo aritmético da moda das distâncias encontradas. A distância que mais ocorrer entre os *pixels* brancos é considerada a distância final entre uma faixa e outra. A Figura 17 ilustra o resultado obtido com esse algoritmo para uma imagem qualquer.

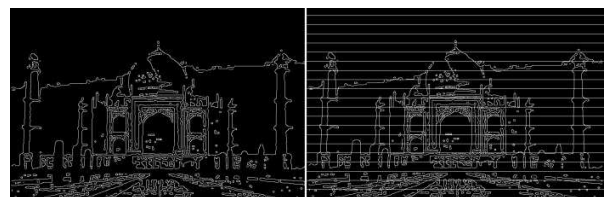


Figura 17 – Linhas horizontais traçadas para interceptar os contornos obtidos pelo *Canny Edge*.

## D. Processamento e Decodificação

O processamento da imagem capturada é realizado em duas etapas: detecção da região de interesse e cálculo da distância entre faixas. A primeira etapa tem por finalidade detectar a região de interesse, ou seja, a região da imagem na qual se concentra a fonte de luz. Já a segunda etapa tem como objetivo extrair a distância em *pixels* entre as faixas capturadas com o LED aceso e as faixas capturadas com o LED apagado.

Após o termino do processamento da imagem e com a distância em *pixels* entre as faixas já conhecida, é aplicada a

fórmula de decodificação e então obtida a frequência na qual o LED foi modulado pelo transmissor.

As duas etapas do processamento da imagem e a decodificação da frequência estão detalhadas nos tópicos que se seguem.

### 1) Detecção da Região de Interesse

O processo de detecção da região de interesse inicia-se pela conversão da imagem capturada para escalas de cinza, utilizando o algoritmo *Gray Scale* apresentado anteriormente. Após esta conversão é aplicado o Filtro Gaussiano (*Gaussian Blur*) para que os ruídos sejam removidos e facilite a detecção das bordas. Uma vez aplicado o Filtro Gaussiano, a imagem passa por uma conversão binária através do algoritmo OTSU. É nessa etapa onde acontece a separação da região de interesse de outros elementos não relevantes na análise da imagem. Na Figura 18 pode se observar a região de interesse sendo definida pelos pixels brancos. Logo após a conversão binária é utilizado o algoritmo *Canny Edge* para que sejam detectadas as bordas da região da imagem que contém nossa região de interesse e consequentemente é onde se concentra a fonte de luz.

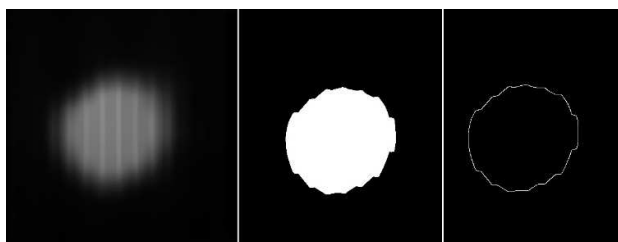


Figura 18 – Região da fonte de luz, onde é aplicado o *Gaussian Blur*, OTSU e *Canny Edge*.

### 2) Cálculo da Distância entre Faixas

Depois de realizada a primeira etapa uma nova imagem é criada apenas com a região de interesse definida anteriormente. Inicia-se então o processo de detecção das faixas definidas pela frequência transmitida, onde a imagem é novamente convertida pelo algoritmo OTSU e passa a conter apenas *pixels* binários. Em seguida, é utilizado o algoritmo *Canny Edge* para a detecção de bordas de cada faixa, onde as bordas são definidas por *pixels* brancos. Uma vez que as bordas das faixas estão claramente definidas torna-se possível utilizar o algoritmo que calcula a distância entre *pixels* brancos vizinhos na horizontal. Como estamos modulando a frequência utilizando um *duty cycle* de 50% a distância entre esses *pixels* definirá a frequência contida na imagem. Vale ressaltar que independente da distância entre a fonte de luz e a câmera a distância entre as faixas não se altera. O processo de cálculo da distância entre faixas pode ser melhor visualizado pela Figura 19 a seguir:

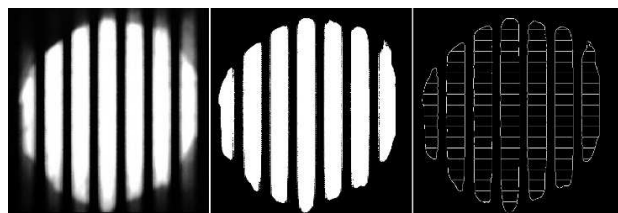


Figura 19 - Imagem analisada na região de fonte de luz, onde respectivamente é utilizado o algoritmo OTSU, *Canny Edge* e algoritmo de distância entre pixels brancos na horizontal.

### 3) Decodificação

A partir da distância em *pixels* obtida pelo processamento da imagem e conhecendo a taxa de varredura (*scan rate*) da câmera do *smartphone*, a frequência do transmissor pode ser obtida através da equação a seguir:

$$frequência = \frac{1}{\frac{1}{scan\ rate} \times distância\ em\ pixels}} \times 2$$

## V. CONCLUSÃO

A precisão no posicionamento *indoor* vem sendo um grande desafio para a computação. Este trabalho contribui com uma pequena, mas fundamental parte neste desafio que é a identificação da frequência transmitida por dispositivos que utilizam VLC.

O desempenho do protótipo desenvolvido apresentou-se de forma satisfatória, conseguindo identificar frequências transmitidas pelo LED em ambientes pouco ou muito iluminados e até mesmo através do reflexo em objetos como paredes e mesas.

Durante o desenvolvimento observou-se que apesar do sistema operacional Android disponibilizar o acesso a métodos capazes de ajustar parâmetros da câmera depende do fabricante tornar acessível estes ajustes do hardware.

Tendo demonstrado a viabilidade da abordagem básica, um trabalho futuro poderia explorar melhor a modulação da frequência transmitida pelo LED, possibilitando um maior canal de comunicação e poderiam ser implementadas melhorias no algoritmo de processamento de imagem para se adaptar aos parâmetros de diferentes câmeras.

O código fonte do trabalho descrito neste artigo está disponível em:

<https://github.com/gustavohsantana/identificacao-de-frequencia-transmissores-vlc>

## REFERÊNCIAS

- [1] Philips Systems, Bringing the power of GPS indoors. Disponível em <<http://www.lighting.philips.com/main/systems/themes/led-based-indoor-positioning>>. Acessado em 28 de agosto de 2017.
- [2] PEREIRA, Ana Rita. *Multilateração por localização Indoor*. Julho de 2014. 39 f. Dissertação (Mestrado) – Curso de Matemática, Universidade de Coimbra, Coimbra, Portugal, 2014.
- [3] OLIVEIRA, Felipe. Smartphones estão nas mãos de 62% dos brasileiros. *Folha de São Paulo*, São Paulo, 27 de fevereiro de 2017.



- Disponível em: <<http://www1.folha.uol.com.br/tec/2017/02/1862362-smartphones-estao-nas-maos-de-62-dos-brasileiros-diz-google.shtml>>. Acessado em 28 de Agosto de 2017.
- [4] YANG, Zhice; WANG, Zeyu; ZHANG, Jiansong; HUANG, Chenyu; ZANG, Quian. Wearables Can Afford: Light-weight Indoor Positioning with Visible Light. Florença, Itália, 2015. Disponível em: <<https://www.microsoft.com/en-us/research/wp-content/uploads/2015/05/pixel.pdf>>. Acessado em 12 de Setembro de 2017.
  - [5] HUYNH, Phat; YOO, Myungsik. VLC-Based Positioning System for an Indoor Environment Using an Image Sensor and an Accelerometer Sensor. Brasília, Suíça, 2016. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4934209/#sec1-sensors-16-00783title>>. Acessado em 12 de Setembro de 2017.
  - [6] EKØDESIGN&INTERACTIVE; (São Paulo). **Indoor Maps**: Sistema de Posicionamento Indoor. 2013. Disponível em: <<http://indoormap.com.br/>>. Acesso em: 12 set. 2017.
  - [7] QUALCOMM (Califórnia). **Lumicast**. 2016. Disponível em: <<https://www.qualcomm.com/lumicast>>. Acesso em: 12 set. 2017.
  - [8] STATCOUNTER. Android overtakes Windows for first time. <<http://gs.statcounter.com/press/android-overtakes-windows-for-first-time>>
  - [9] ANDROID DEVELOPERS. Android, the world's most popular mobile platform <<https://developer.android.com/about/index.html>>
  - [10] ARDUINO. Arduino Software IDE. Disponível em: <<https://www.arduino.cc/en/Guide/Environment>>. Acessado em: 27 de setembro de 2017.
  - [11] IBM. Introdução à plataforma Eclipse. Disponível em: <<https://www.ibm.com/developerworks/br/library/os-eclipse-platform/>>. Acessado em: 27 de setembro de 2017.
  - [12] ROCHA, Leonardo. **Android Studio**: ferramenta de criação de apps da Google ganha versão 1.0. 2014. Disponível em: <<https://www.tecmundo.com.br/android/69111-android-studio-ferramenta-criacao-apps-google-ganha-versao-1-0.htm>>. Acesso em: 29 set. 2017.
  - [13] ANDROID. Android Studio: O IDE oficial do Android. Disponível em: <<https://developer.android.com/studio/index.html?hl=pt-br>>. Acessado em: 29 de setembro de 2017.
  - [14] ARDUINO. Introdução ao Arduino. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acessado em: 29 de setembro de 2017.
  - [15] ARDUINO. Arduino Uno Rev3. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acessado em: 29 de setembro de 2017.
  - [16] SANTOS, Marco Aurélio da Silva. Transistor. Disponível em: <<http://mundoeducacao.bol.uol.com.br/fisica/transistor.htm>>. Acessado em: 12 de novembro de 2017.
  - [17] JORDÃO, Fábio. LED: a iluminação do futuro já disponível no presente. 2009. Disponível em: <<https://www.tecmundo.com.br/qd-led/2654-led-a-iluminacao-do-futuro-ja-disponivel-no-presente.htm>>. Acessado em: 29 de setembro de 2017.
  - [18] Lúmens. O que são watts e lúmens? Disponível em: <<https://www.philips.com.br/c-m-li/advice-on-lighting-ideas/low-watt-high-lumen-bulbs>>. Acessado em: 16 de novembro de 2017.
  - [19] "A Review of the Literature on Light Flicker: Ergonomics, Biological Attributes, Potential Health Effects, and Methods in Which Some LED Lighting May Introduce Flicker," IEEE Standard P1789, February 2010.
  - [20] Cambridge, Dictionary. Pixel <<https://dictionary.cambridge.org/pt/dicionario/ingles/pixel>> acessado em 15/11/2017
  - [21] MANSUROV, Nasim. Understanding Shutter Speed – A Beginner's Guide. Disponível em: <<https://photographylife.com/what-is-shutter-speed-in-photography>>. Acessado em: 30 de outubro de 2017.
  - [22] Norman, Ian. How to Find the Best ISO for Astrophotography: Dynamic Range and Noise. <<https://www.lonelyspeck.com/how-to-find-the-best-iso-for-astrophotography-dynamic-range-and-noise>>. Acessado em: 12 de novembro de 2017.
  - [23] RASTERGRID, Efficient Gaussian blur with linear sampling. Disponível em: <<http://rastergrid.com/blog/2010/09/efficient-gaussian-blur-with-linear-sampling>>. Acessado em 13 de Novembro de 2017
  - [24] Torok, Leonardo. Método de Otsu. <<http://www2.ic.uff.br/~aconci/OtsuTexto.pdf>>. Acessado em: 12 de novembro de 2017.

**Daniele de Sá Freitas** nasceu em Volta Redonda, RJ, em outubro de 1994. Graduanda do 10º período de Engenharia de Computação pelo Inatel – Instituto Nacional de Telecomunicações. Participou do trabalho voluntário Casa Viva da instituição durante os anos de 2013 a 2016. Ingressou na empresa júnior do INATEL, CP2eJr, em 2016, no núcleo de engenharia e desenvolveu projetos de software para a mesma. Concluiu o estágio obrigatório no primeiro semestre de 2017, na empresa Ciena, com trabalho em área de gerenciamento de projetos. Tem interesse em áreas de TI, gerenciamento de projetos e empreendedorismo.

**Gustavo Henrique Santana de Souza** nasceu em Pouso Alegre, MG, em maio de 1995. Graduando do décimo período de Engenharia de Computação pelo Inatel – Instituto Nacional de Telecomunicações. Publicou artigos de Iniciação Científica no Incitel – Congresso de Iniciação Científica do Inatel com os temas de Software de apoio acadêmico para o estudo de Algoritmos em 2016 e Aplicação de Software para sistemas de localização Indoor em 2017. Desde fevereiro de 2017 é estagiário da área de *supply chain* na empresa Embracer S.A. Tem interesse na área de empreendedorismo e Business Intelligence.

**Ney Correa da Silva Neto** nasceu em Itajubá, MG, em julho de 1992. Graduando do 10º período de Engenharia da Computação pelo Instituto Nacional de Telecomunicações. Desde fevereiro de 2017 é estagiário na empresa Teclable Produtos Eletrônicos Ltda. onde atua na área de manutenção e desenvolvimento de sistemas. Tem interesse nas áreas de *machine learning* e desenvolvimento de sistemas inteligentes.

**Pedro Henrique Bernardes Almeida** nasceu em Santa Rita do Sapucaí, MG, em junho de 1995. Possui o título de Técnico em Eletrônica (ETE “FMC”, 2013) e atualmente é graduando do 10º período de Engenharia da Computação pelo Instituto Nacional de Telecomunicações. Desde 2014 atua no desenvolvimento de aplicativos móveis para as plataformas Android e iOS. Já passou por empresas como Inatel Competence Center, Leucotron Telecom e atualmente é desenvolvedor *mobile* na WatchGuard Technologies. Tem interesse nas áreas de desenvolvimento de aplicativos móveis e empreendedorismo.