Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Estimação da Posição do Teclado em Dispositivos Móveis a partir de Vídeos Capturados por Câmeras de Vigilância

Marcelo A. Winkler

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. Dr. Alexandre Zaghetto

Brasília
2016

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Estimação da Posição do Teclado em Dispositivos Móveis a partir de Vídeos Capturados por Câmeras de Vigilância

Marcelo A. Winkler

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof. Dr. Alexandre Zaghetto (Orientador)
CIC/UnB

Prof. Dr. Luiz Henrique M. Aguiar
CIC/UnB

Prof. Dr. Marcus Vinícius Chaffim Costa
CIC/UnB

Prof. Dr. Ricardo Jacobi
Coordenador do Curso de Engenharia da Computação

Brasília, 15 de novembro de 2016

# Dedicatória

À minha família: meu pai, minha mãe, minha irmã, Glorinha e Louella. Sem vocês nada disso seria possível.

# Agradecimentos

À minha mãe, *Maria de Lujan Caputo Winkler*, e ao meu pai, *José Calmon Winkler*, por terem sempre proporcionado todas as oportunidades que me foram dadas, sempre com muito amor e carinho. Pela confiança em minhas habilidades e por todo o apoio dado, os quais sempre me motivaram a sonhar mais alto e não perder o foco em face dos desafios apresentados. Agradeço também por todo esforço e trabalho que realizam diariamente para tornar a nossa família a melhor possível.

À minha irmã, *Stephanie Winkler*, que sempre está ao meu lado independente da situação. Especificamente, agradeço toda a amizade e ajuda nesse último ano. Pelo companheirismo, até nos momentos difíceis, que apesar de tudo, sempre me mostra que é possível dar risadas.

À minha namorada, *Louella Trindade Silva*, que desde quando a conheci só tem me trazido felicidade. Pelo amor, pelo carinho e por todo o tempo que temos passados juntos. Pela sua paciência e compreensão nos momentos em que nos faltava tempo para ficarmos juntos.

À minha segunda mãe, *Gloria Guevara Teran Nogueira*, que tem me auxiliado em manter à ordem dentro de casa e em certificar que eu sempre estivesse bem alimentado. Por todo seu esforço e trabalho duro que realiza diaramente dentro de casa.

Ao meu orientador, *Alexandre Zaghetto*, por ser um dos motivos de ter escolhido mudar de curso e seguir a área profissional que sempre desejei. Por ser um dos melhores professores que eu já tive, pela sabedoria que passa e pela compreensão durante todo esse processo.

# Resumo

O presente trabalho tem como objetivo a utilização de técnicas de processamento de imagens e vídeos para demonstrar a vulnerabilidade existente em sistemas de segurança baseados no uso de códigos PINs (Personal Identification Number) em ambientes vigiados por câmeras. Para exemplificar essa fragilidade do sistema de autenticação, um experimento foi desenvolvido em que várias pessoas foram filmadas ao inserirem alguns PINS em teclados diferentes. Um algoritmo foi criado para detectar a localização aproximada do teclado, aplicando técnicas de estimação de movimento e operações morfológicas, a fim de demonstrar a viabilidade desse modelo de ataque.

**Palavras-chave:** processamento de imagens, processamento de vídeos, segurança, pin

# Abstract

The main goal of the following paper is to demonstrate the existing vulnerability in PIN code based security systems present in locations under surveillance, utilizing techniques in image and video processing. To exemplify the fragility of this authentication system, an experiment was conducted in which several people were filmed as they inserted a couple of PIN codes into different keypads. An algorithm was created to detect the approximate location of the keypad, by applying motion estimation techniques and morphological operations, to demonstrate the viability of this attack model.

**Keywords:** image processing, video processing, security, pin

# Sumário

# Lista de Figuras

# Capítulo 1

# Introdução

## 1.1 Contextualização

O sistema de segurança do código PIN atualmente é implementado em vários dispositvos como uma forma de proteger o acesso aos dados ou aos estabelecimentos particulares de invasores. Máquinas de cartão de crédito, celulares, *tablets*, portas de segurança, entre outros, comumente implementam este tipo de sistema. Mesmo presente já há algumas décadas e amplamente difundido, está longe de ser totalmente seguro. Diversos tipos de ataques foram desenvolvidos para roubar o código ou até mesmo burlar o mecanismo de autenticação realizado com o PIN.

Locais que utilizam formas de autenticação baseados no uso do código PIN, frequentemente adotam câmeras de vigilância para reforçar a segurança dos estabelecimentos. Postos de gasolina, lojas de departamento, empresas privadas, e até edifícios residências são alguns exemplos que possuem ambos sistemas. Em 2012, sistemas de circuitos de TV (baseados no monitoramento por câmeras de vigilância) representavam 43% das principais tecnologias aplicadas em segurança no Brasil, segundo dados da ABESE (Associação Brasileira de Empresas de Sistemas Eletrônicos de Segurança). Atualmente, é possível encontrar mais de um milhão de câmeras instaladas só na cidade de São Paulo[14].

## 1.2 Apresentação do Problema e Justificativa

O propósito das câmeras de segurança é monitorar um local e registrar os acontecimentos de forma a prevenir a ocorrência de atos ilícitos, ou ajudar na perícia após o crime. Por esse motivo, muitas pessoas possuem uma confiança inerente ao frequentar estabelecimentos com câmeras de vigilância. Ao mesmo tempo que as câmeras servem de certa forma para proteger as pessoas, também se revelam como uma ameaça a sua privacidade. Estes dipositivos não distinguem o que estão registrando e é justamente por isso que são capa-

zes de registrar dados sensíveis como o código PIN de uma pessoa. Este trabalho explora esse cenário específico, em que ocorre o registro de código PINs por câmeras, e verifica as eventuais e possíveis vulnerabilidades presentes neste sistema. Uma das vulnerabilidades identificadas deve-se ao descaso do usuário do sistema ao ser gravado pelas câmeras de vigilância e será apresentada por meio dos resultados de uma pesquisa realizada com um questionário. Alguns métodos foram desenvolvidos para expor parcialmente as vulnerabilidades do próprio sistema por meio de um experimento realizado. O problema, que se tenta resolver nesse trabalho, consiste em detectar automaticamente a localização do teclado utilizado por pessoas para inserirem seus PINs. Com esta informação, será possível extrair mais facilmente os PINs inseridos, tornando o modelo de ataque descrito mais efetivo.

## 1.3 Organização do Trabalho

O segundo capítulo desta monografia apresenta uma revisão bibliográfica dos conceitos fundamentais para a compreensão deste trabalho, assim como uma comparação com trabalhos correlatos. O terceiro capítulo descreve a metodologia empregada para o desenvolvimento do projeto e a explicação da solução proposta. O quarto capítulo discute os resultados experimentais obtidos. O quinto capítulo apresenta as conclusões e as possibilidades de trabalhos futuros.

# Capítulo 2

# Fundamentação Teórica

Este capítulo descreve os conceitos téoricos que servem como base para o entendimento do trabalho realizado.

## 2.1 Personal Identification Number

O código PIN é um número de identificação pessoal utilizado em sistemas de segurança de diversos dispositivos, que permite realizar a autenticação do usuário ao inserir o código correto. Caso qualquer outra sequência de números for inserida, o acesso ao dispositivo é negado. Esse sistema surge como consequência e necessidade da invenção da caixa eletrônica, ou *ATM* (*Automatic Teler Machine*). Em 1960 e nos anos que seguiram, várias pessoas não conseguiam sacar dinheiro ou realizar outras transações bancárias, pois trabalhavam nas horas em que os bancos estavam abertos. Diversos bancos buscaram soluções para este problema, visando o desenvolvimento de uma máquina que poderia disponibilizar dinheiro para os seus clientes[9]. Há controversias sobre quem foi o primeiro inventor da caixa eletrônica e do código PIN, mas há dois que se destacam, John Sheperd-Barron e James Goodfellow.

John Sheperd-Barron concebeu a ideia de uma máquina que receberia cheques (não havia cartões de plástico naquela época) e dispensaria uma quantia de dinheiro. Os cheques eram impregnados com uma substância levemente radioativa, o carbono-14, e inseridos na máquina que identificava o cliente com o seu código PIN de quatro números. A instalação da primeira caixa ocorreu em 1967 numa agência do banco inglês Barclays[1].

James Goodfellow trabalhava como engenheiro na empresa *Smith Industries*, quando foi designado com a tarefa de construir o mesmo dispositivo. Ao invés de cheques, a sua invenção aceitava cartões perfurados de plástico codificados que possuiam uma relação aritmética com o PIN inserido pelo cliente. Em 1967, Goodfellow patenteou a tecnologia PIN conforme as patentes US3905461 e GB1197183[7][8].

A utilização da tecnologia PIN tem evoluído para o desbloqueio de telas de dispositivos móveis, para a autenticação em pagamentos realizados com máquinas de cartão de crédito e outras formas de segurança. Muitos destes utilizam apenas quatro números para realizar a validação, o que limita um ataque de força bruta a somente dez mil tentativas.

## 2.2 Imagens

Uma imagem é denotada como uma função bidimensional de forma $f(x, y)$, onde o valor, nas coordenadas $(x, y)$, é proporcional a energia irradiada de uma fonte. A energia correspondente pertence ao espectro da luz visível, cuja faixa de frequências engloba todas as cores. Já a luminosidade desprovida de cor é dita acromática e apenas o seu atributo quantitativo é medido. A escala de cinza é utilizada para descrever este atributo em termos de intensidade luminosa e varia de preto para tons de cinza até o branco.

Imagens são modeladas como contínuas, significando que sua quantidade de energia, ou amplitude, e suas coordenadas podem assumir valores reais arbitrários, os quais podem estar dentro de um intrevalo definido. Imagens digitais são formadas a partir da amostragem dos valores das coordenadas e da quantização da amplitude. A amostragem consiste em dividir os valores das coordenadas em intervalos de espaço iguais para obter um conjunto discreto de localizações. Já a quantização consiste em dividir a faixa de valores da amplitude em intervalos iguais obtendo um conjunto discreto de níveis.

A amostragem e a quantização resultam em um conjunto de valores reais que podem ser representados por uma matriz. O número de linhas e colunas da matriz é determinado pela quantidade de amostras das coordenadas x e y, respectivamente. O valor de cada elemento da matriz é o valor quantizado da amplitude para aquela coordenada. Assim, cada elemento de uma matriz é chamado de *picture element*, ou *pixel* [6].

## 2.3 Vídeo

Vídeos são cenas compostas por sequências de imagens transcorridas ao longo do tempo. Além das dimensões espaciais discutidas anteriormente das imagens, vídeos possuem uma nova dimensão, o tempo. O período de captura é continuo, portanto, uma amostragem é necessária também. Para isto, uma imagem é gravada em intervalos regulares de tempo.

Cada imagem digital de um vídeo é chamado de quadro, ou *frame*. A frequência em que os *frames* são capturados, ou seja a amostragem temporal do vídeo, é chamado de *frame rate*. O *frame rate* é dado em segundos (*frames per second*, ou fps) e os padrões atuais de vídeos digitais geralmente possuem desde 25 *frames* até 120 *frames* por segundo. A

exibição dos quadros em sequência dá a impressão de movimento e quanto mais amostras são capturadas mais suave aparenta ser a movimentação.

A amostragem de um vídeo pode, ao invés de utilizar a captura de *frames* completos, ser realizada por meio de uma sequência de linhas intercaladas. Cada imagem, neste caso, é capturada em duas passagens, ambas de cima para baixo, onde a primeira realiza a leitura de linhas horizontais intercaladas e na segunda passagem as demais linhas são lidas. Cada leitura é realizada a cada amostra de tempo e uma sequência de linhas intercaladas é chamada de *field*. A vantagem de utilizar este tipo de amostragem é que é possível enviar duas vezes mais *fields* que *frames* no mesmo intervalo de tempo dado uma taxa fixa de dados.

## 2.4 Estimação de Movimento

Vídeos e imagens capturados são quantizados e codificados em bits, onde os valores de cada *pixel* possui uma representação binária. Os bits de informação são organizados de acordo com o formato de arquivo e muitas vezes o tamanho destes arquivos são muito grandes. Uma imagem sem compressão de tamanho 640x480 pixels que utiliza 24 bits para representar o espectro de cores, por exemplo, ocupa quase um mega byte de espaço:

$$640 \times 480 \times 24 = 7.372.800 \text{ bits e}$$

$$7.372.800 \div 8 = 921.600 \text{ bytes.}$$

Para reduzir o tamanho dos arquivos é necessário utilizar técnicas de compressão. Vídeos utilizam um sistema de compressão que envolve converter os dados originais para um formato que ocupa um tamanho reduzido de bits para ser armazenado. Esse sistema é chamado de *encoder* e realiza a compressão removendo informação redundante. *Frames* consecutivos frequentemente possuem muita informação similar, considerado redundância temporal, e pixels próximos uns dos outros, em um mesmo frame, geralmente possuem alta taxa de correlação, considerado redundância espacial. Posteriormente, outro sistema, conhecido como *decoder*, realiza a decompressão dos dados transformando-os de volta ao seu formato original. A remoção da redundância pode acarretar a perda de informação, geralmente em uma taxa aceitável que permite reduzir ainda mais o tamanho ocupado sem afetar muito a qualidade visual.

O *encoder* utiliza um modelo de previsão que visa reduzir redundâncias ao construir uma previsão do *frame* atual, a partir de informações de *frames* vizinhos, e subtraindo esta previsão do *frame* atual. O resultado deste processo é um *frame* residual com menos dados, mas com informação suficiente para que possa ser utilizada para reconstrução do

*frame* original na decodificação. Uma forma eficiente de formar a previsão é utilizando uma técnica de compensação de diferenças entre *frames* vizinhos.

Uma das causas das diferenças entre quadros é o movimento, podendo este ser de objetos ou da própria câmera. Para compensar este movimento, são utilizados métodos de estimação de movimento que determinam vetores de movimento usados para identificar os deslocamentos entre os quadros. A seguir, o trabalho discute alguns algoritmos do método de estimação por *block-matching*.

### 2.4.1 Block-Matching

Os métodos de *block-matching* são uns dos mais utilizados, pois estão presentes em todos os padrões de codificação de vídeos atuais[16]. Segundo os autores Béatrice, Marco e Frédéric (2014)[12], a técnica consiste em encontrar blocos, ou submatrizes, em dois *frames* que possuem a menor diferença entre os valores de seus pixels para determinar os vetores de movimento. Um bloco $B_{p,q}$ é definido como um conjunto de indíces de um *frame* começando de $(p, q)$ e possui tamanho $P \times Q$:

$$B_{p,q} = \{p, p+1, ...p+P-1\} \times \{q, q+1, q+Q-1\}$$

Apenas um vetor de movimento é determinado para todos os pixels contidos no bloco. Na execução dos algoritmos de *block-matching*, um bloco do atual quadro é comparado com outro bloco de um segundo quadro, chamado de referência. O bloco no quadro de referência é deslocado em relação à posição inicial do bloco no quadro atual e seu deslocamento é representado por um vetor rotulado como *candidate motion vector*. A predição que utiliza um quadro anterior ao atual como referência é chamada de *forward prediction* e quando o *frame* de referência é um quadro futuro é conhecida como *backward prediction*.

Existem diferentes estratégias de busca pelo bloco, no *frame* de referência, que mais se assemelha ao bloco *frame* atual sendo comparado. A estratégia *full search* envolve procurar por todo o quadro de referência, onde todos os possíveis blocos são analizados, pelo o bloco com a menor diferença. De acordo com os autores, a procura por todo o *frame* não é necessária, podendo ser restringida a uma área retangular centrada na posição $(p, q)$. Essa área conhecida como *search window*, ou janela de busca, constitui o conjunto de *candidate motion vectors* adequados para realizar estimação do movimento. O tamanho apropriado da janela é diretamente proporcial à amplitude do movimento, sendo que movimentos maiores requerem uma área de busca maior [3]. A vantagem do *full search* é que ele sempre determina o melhor bloco correspondente, porém é o algoritmo mais computacionalmente custoso.

Uma outra estratégia de busca bastante conhecida é a *three step search*. Esse método visa reduzir ainda mais o número de comparações realizadas dentro da janela de busca. O algoritmo *three step search*, ou TSS, consiste em tomar o centro do bloco do *frame* atual e colocá-lo no centro da janela de busca no *frame* de referência. Ao invés de comparar com todas as localizações possíveis na janela de busca, inicialmente realiza-se a comparação no centro e nas outras oito posições vizinhas, a uma distância de quarto pixels, conforme a Figura 2.1. Das nove localizações aquela com a menor diferença é escolhida como centro para a próxima iteração. Novamente outras oito posições vizinhas são selecionadas, mas a distância é reduzida à metade, conforme ilustrado na figura Figura 2.2. Novamente a localização com melhor resultado de comparação é selecionado como o centro da última iteração. Mais oito vizinhos são selecionados, reduzindo novamente a distância pela metade, e são executadas mais nove comparações. O vetor de movimento é determinado como sendo a distância do centro da janela de busca até posição do melhor resultado da última iteração Figura 2.3. O custo computacional do *three step search* é significamente menor quando comparado com o *full search* e mesmo que não garanta encontrar o melhor vetor de movimento é um método eficiente para a estimação de movimento.

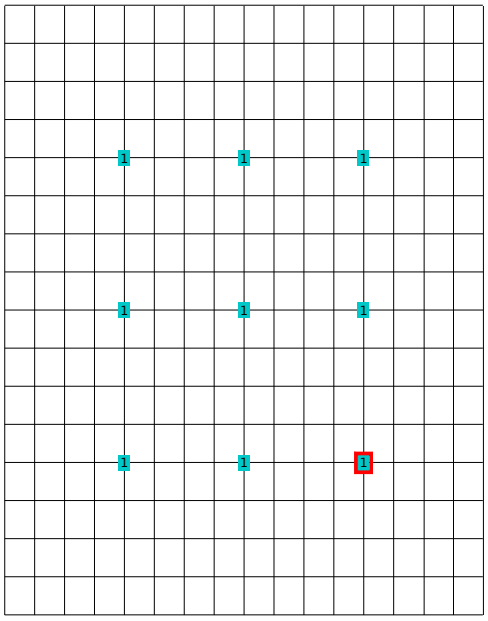

Figura 2.1: Passo 1 do TSS. A posição marcada em vermelho foi a escolhida.

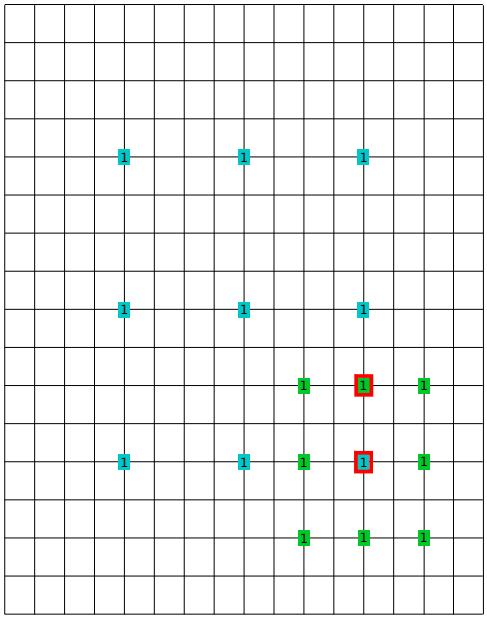

Figura 2.2: Passo 2 do TSS. A posição marcada em vermelho foi a escolhida.

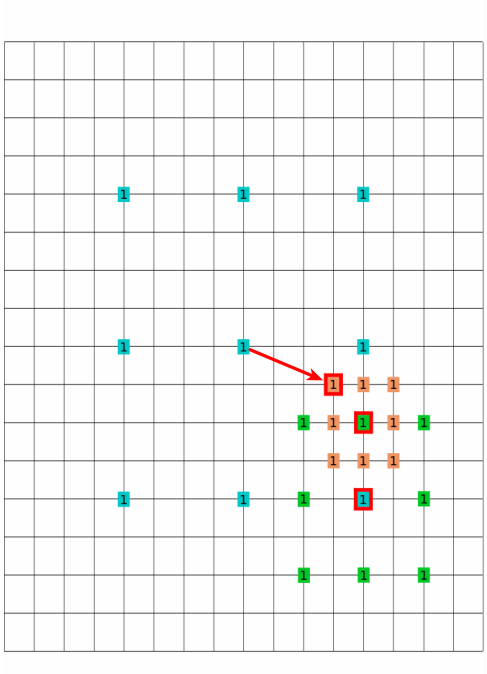

Figura 2.3: Passo 3 do TSS. A posição marcada em vermelho foi a escolhida.

## 2.5 Sistema de cores

O olho humano possui milhões de células fotoreceptoras que absorvem e convertem a luz em sinais elétricos, enviando-os para o nervo óptico e posteriormente ao cérebro para serem processados. Cones são as células fotoreceptoras que são responsáveis pela visão das cores. As células cones podem ser divididas em três tipos, cada uma possuindo um fotopigmento sensível a um comprimento de onda diferente [15]. Estes comprimentos de onda são das cores vermelha, verde e azul e por este motivo surgiu o padrão RGB (*Red*, *Green*, *Blue*).

O sistema RGB forma suas variadas cores por meio da superposição das ondas de suas cores primárias, adicionando cada um de seus comprimentos de ondas para formar a mistura resultante [13]. As cores secundárias surgem ao adicionar duas cores primárias, como ilustra a Figura 2.4. Adicionando todas as três cores primárias, a luz branca é gerada. Imagens que utilizam o sistema RGB são formadas por matrizes tridimensonais, onde cada dimensão representa uma cor primária. Cada pixel em cada dimensão denota a quantidade luminosa da cor, sendo que um valor zero indica a ausência de cor (preto) e quanto maior o seu valor, mais clara é a cor. A Figura 2.5, a Figura 2.6 e a Figura 2.7 mostram os componentes vermelho, verde e azul da Figura 2.8.

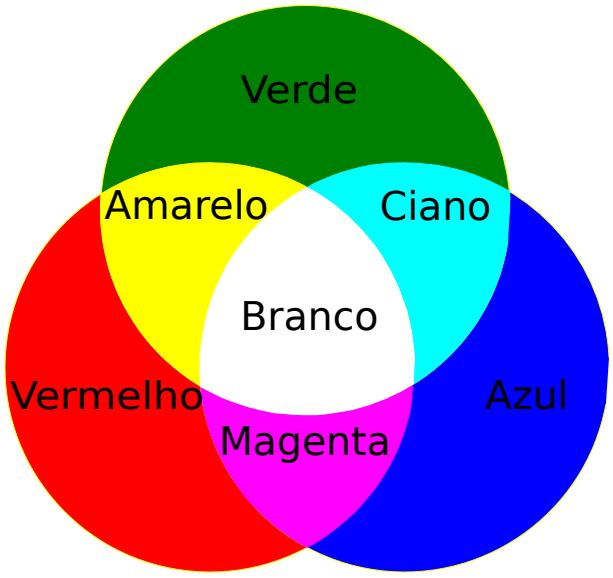


Figura 2.4: Propriedade aditiva das cores primárias forma as cores secundárias e a luz branca[6].

Outro sistema de cor bastante utilizado no processamento de imagens é o HSV (*Hue*, *Saturation and Value*). *Hue*, ou matiz, é a tonalidade da cor percebida de forma mais evidente, caracterizada pelo comprimento de onda dominante. Existem quatro matizes unitários básicos: o vermelho, o amarelo, o verde e o azul. *Saturation*, ou saturação, é a

Figura 2.5: Componente vermelho da imagem Lena.

Figura 2.6: Componente azul da imagem Lena.

quantidade de matiz relativa a quantidade de luz branca de uma dada cor. Cores mais claras como rosa, possuem maior componente acromático do que cromático. Neste caso, a matiz vermelha possue menor saturação. Já *value* refere-se a precepção da quantidade luminosa provinda de uma cor quando esta é a única fonte de luz. Uma dada cor deste modelo é representada por três números que indicam o valor de cada componente. A matiz é dada por um número de graus variando de 0 a 360, em que cada 60 graus indica

Figura 2.7: Componente verde da imagem Lena.

Figura 2.8: Imagem da Lena composta pelos três componentes RGB.

a saturação máxima de uma cor primária ou secundária. Por exemplo, 0 graus representa a cor vermelha, 60 graus, a cor amarela, 120 graus, a cor verde e assim por diante. A saturação é um número entre 0 e 1, onde 0 representa a luz branca e 1 a saturação máxima. O valor, assim como a saturação, é um número entre 0 e 1, onde 0 representa nenhuma luminosidade e 1 a intensidade máxima luminosa [2]. A representação em 3-D do modelo HSV é ilustrada na figura Figura 2.9.

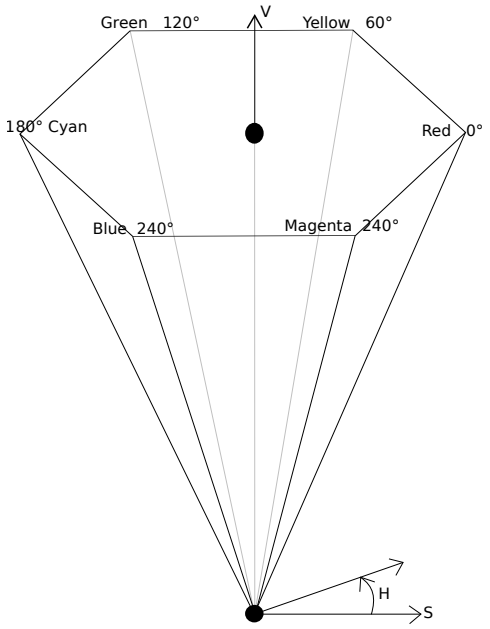


Figura 2.9: Modelo “Hexcone” do sistema HSV.

## 2.6 Segmentação

Segmentação é a divisão de uma imagem digital em um conjunto de regiões diferentes (segmentos) que possuem características de interesse. Os objetivos da segmentação são extrair as regiões de interesse, o que facilita a análise, processar adicionalmente as mesmas, e modificar essas regiões de forma que representem imagens com maior significado. Segmentação é comumente utilizada para identificar segmentos de linhas de borda de figuras, formas variadas e objetos específicos [17].

Uma técnica de segmentação envolve extrair regiões da imagem que pertencem a uma determinada faixa de cores. Uma forma de realizar este tipo de segmentação é determinando um conjunto de cores como referência, do sistema RGB, e estabelecer um limite, ou *threshold*, máximo de variação aceitável desse conjunto de cores. Para cada pixel da imagem, determina-se a distância euclidiana de cada componente RGB para cada cor de referência e caso a distância for menor que o *threshold*, o *pixel* é mapeado para o valor 1. Se a distância for maior que o limiar, o pixel é mapeado para 0. O resultado é uma imagem binária onde a região com as cores desejadas são ilustradas em branco e todo o restante da imagem é preta. *Thresholding* também pode ser utilizado em imagens de escala de cinza para segmentar regiões de interesse.

Uma outra forma de *thresholding* é, ao invés de determinar uma distância limite, simplesmente definir um valor limite, onde qualquer *pixel* com valor igual ou abaixo desse

limiar é mapeado para 0 e qualquer pixel com valor maior que o limite é mapeado para o valor máximo. Esta forma é comumente utilizada por ser um método simples e eficiente de limiarização.

## 2.7 Morfologia

Na área da matemática, morfologia é uma teoria que começou a ser desenvolvida na França na década de 1960 e que começou a ser estudada no Brasil na década de 1980. A morfologia matemática utiliza ferramentas matemáticas para a análise de estruturas geométricas em imagens. A morfologia é baseada na teoria de conjuntos, em que um conjunto de elementos bem definidos de uma imagem, chamado de elemento estruturante, é utilizado para realizar comparações com o restante da imagem a fim de extrair informações relativas à geometria dos elementos desconhecidos. O elemento estruturante é composto por um conjunto de pixels que podem ou não interagir com a imagem. O elemento estruturante é composto por pixels que interagem com a imagem, denotados por "•", já os que não interagem são representados por "·". Assim, o sistema a seguir é um exemplo de um elemento estruturante [4]: $\left\{\begin{matrix}\cdot\bullet\cdot\\ \bullet\bullet\bullet\\ \cdot\bullet\cdot\end{matrix}\right\}$. O resultado da interação do elemento com a imagem é geralmente inserido no centro do sistema, simbolizado por "()", como se vê abaixo: $\left\{\begin{matrix}\cdot\bullet\cdot\\ \bullet(\bullet)\bullet\\ \cdot\bullet\cdot\end{matrix}\right\}$.

### 2.7.1 Noções básicas da teoria de conjuntos

Algumas definições da teoria de conjuntos são fundamentais para a compreensão dos demais conceitos sobre morfologia a serem apresentados. Uma operação importante é a de interseção de conjuntos denotada como

$$C = A \cap B,$$

em que $A$, $B$ e $C$ são conjuntos de forma que $C$ possui todos os elementos que pertencem tanto a A quanto a B. Caso a interseção dos conjuntos for vazia, ou seja, não há elementos que pertençam a ambos, os conjuntos são ditos como disjuntos. Um conjunto $C$ que possui todos os elementos distintos de outros dois conjuntos $A$ e $B$ é dito que a união destes e é denotada como

$$C = A \cup B.$$

Um conjunto A é dito contido em outro conjunto B quando todos os elementos de A também são elementos de B, representado como

$$A \subset B.$$

O conjunto A, neste caso, é dito um subconjunto de B, se e somente se, o número de elementos da interseção de ambos os conjuntos for igual ao número de elementos do próprio conjunto A, denotado formalmente por $A \subset B \Leftrightarrow |A \cap B| = |A|$ . O complemento de um conjunto A refere-se a todos os elementos que não estão contidos em A e é expresso da seguinte forma

$$A^C = \{x \mid x \notin A\}$$

A diferença entre dois conjuntos $A$ e $B$ é o conjunto constituído por todos os elementos que pertencem a $A$, mas não pertencem a $B$, formalmente definido como

$$A - B = \{x \mid x \in A,\, x \notin B\}\,.$$

A reflexão de um conjunto $A$, denotado $\hat{A}$, é a reflexão de todos os seus elementos a partir de um determinado ponto de origem, definido como

$$\hat{A} = \{x \mid x = -y, \text{ para } y \in A\}\,.$$

A operação de translação de um conjunto $A$ por um ponto $x = (x_1, x_2)$ é o deslocamento das duas coordenadas de todos os elementos de $A$ por x, ou seja, a translação de $A$ é

$$(A)_x = \{c \mid c = a + x, \text{ para todo } a \in A\}$$

onde $c = (c_1, c_2) = (a_1 + x_1, a_2 + x_2) = a + x$. A seguir serão apresentadas quatro operações bastante utilizadas no processamento morfológico de imagens: a erosão, a dilatação, a abertura e o fechamento.

### 2.7.2 Erosão

A erosão de um conjunto $A$ por um elemento estruturante $B$ é a operação morfológica que fornece todos os pontos de um conjunto $x$, resultantes da translação de $B$ por $x$, de forma que $B$ está contido em $A$. A erosão é denotada $A \ominus B$ e pode ser formalmente definida como

$$A \ominus B = \{x \mid (B)_x \subset A\}\,.$$

A erosão é útil para separar objetos que estão se tocando, remover ruído, reduzir extrusões de formas e detectar bordas ao subtrair a imagem resultante da erosão da imagem original. A erosão também resulta na redução do tamanho dos objetos.

### 2.7.3 Dilatação

A dilatação de um conjunto $A$ por um elemento estruturante $B$ consiste em transladar a reflexão de $B$ por todo os conjunto de pontos de $x$ de forma que o resultado é o conjunto não vazio de pontos que pertencem a interseção da reflexão de $B$ e $A$. A dilatação de $A$ por $B$ é denotada como $A \oplus B$ e pode ser formalmente definida como

$$A \oplus B = \left\{ x \mid \left(\hat{B}\right)_x \cap A \neq \varnothing \right\}.$$

A dilatação é comumente aplicada para preencher quebras existentes em objetos, reparar intrusões e remover ruído. A dilatação aumenta o tamanho dos objetos permitindo a conexão de objetos próximos.

### 2.7.4 Abertura

A abertura de um conjunto $A$ por um elemento estruturante $B$ é a combinação das operações de erosão de $A$ por $B$ seguida pela dilatação do resultado da erosão por $B$. A abertura de $A$ por $B$ é denotado como $A \circ B$, e é definida como

$$A \circ B = (A \ominus B) \oplus B.$$

A abertura também pode ser descrita como sendo a união de todas as translações de $B$, por um conjunto de pontos x, contidas em $A$, definida como

$$A \circ B = \cup \left\{ (B)_x \mid (B)_x \subset A \right\}.$$

Dessa forma, a abertura realiza um nivelamento dos contornos de objeto por seu interior. A vantagem deste tipo de operação é que ela reduz os impactos da modificação do tamanho dos objetos, resultantes da execução independente das operações de erosão e dilatação, sem afetar a suas demais aplicações.

### 2.7.5 Fechamento

O fechamento de um conjunto $A$ por um elemento estruturante $B$ é a combinação das operações de dilatação de $A$ por $B$ seguida pela erosão do resultado da dilatação por $B$.

O fechamento de $A$ por B é denotado como $A \bullet B$, e definida como

$$A \bullet B = (A \oplus B) \ominus B.$$

Diferentemente da abertura, o fechamento é o conjunto não vazio de pontos da interseção de $B$, transladado por um conjunto $x$, por $A$, definido como

$$A \bullet B = \{x \mid (B)_x \cap A \neq \varnothing\}.$$

Assim, o fechamento nivela os contornos de objetos por seu exterior. O fechamento possui a mesma vantagem da abertura de minimizar os impactos da modificação do tamanho dos objetos e ambas operações geram imagens menos ricas em detalhes que suas originais.

## 2.8 Trabalhos correlatos

O trabalho intitulado "*Heat of the Moment: Characterizing the Efficacy of Thermal Camera-Based Attacks*" [10] aborda o uso de câmeras termais para detectar a temperatura de teclas de dispositivos afim de detectar o código PIN inserido. Os autores afirmam que, ao pressionar cada tecla, há transferência de calor do corpo e esta transferência deixa traços termais residuais, que podem ser capturados pela câmera mesmo após um período significativo de tempo. Os experimentos realizados utilizaram dois tipos de teclados, um de metal polido e outro de borracha, que foram filmados antes, durante e depois da inserção do código. As gravações foram primeiramente revisadas por uma pessoa para verificar se era possível determinar o código inserido e assim determinar uma base de performance. Depois, foram passadas para o algoritmo para serem processadas de forma automática e os resultados dos dois foram comparados. O algoritmo analisava, *frame* a *frame*, 10 áreas fixas diferentes, as teclas. Ele, por sua vez, também comparava as regiões de interesse dentro destas àreas em um *frame* de referência, tomado antes da inserção do código, com as áreas dos *frames* após a inserção. A região de interesse era definida por uma de três formas possíveis: pela temperatura máxima de cada área, pela média artimética da área, ou pela análise caso houvesse aumento de temperatura. A sequência do código era definida por meio da subtração das regiões determinadas nos *frames* após a inserção com as regiões do *frame* de referência, onde a ordem era representada pela ordem crescente da temperatura dos resultados. Os resultados dos experimentos demonstram altas taxas de sucesso em identificar os números digitados, mas não muito efetivo em determinar a ordem em que foram inseridos. O método utilizado apresenta algumas vantagens sobre os métodos tradicionais de ataque, como conseguir identificar o código inserido, mesmo que a visão da câmera do teclado for bloqueada durante a inserção. Uma outra vantagem

do método é o fato de não ser necessário fixar a posição da câmera para realizar o ataque. O método também apresenta limitações, como dificuldade em identificar as teclas pressionadas em teclados metálicos, devido a sua alta condutibilidade térmica, da mesma forma quando a pessoa possui o toque mais leve, ou quando esta possuir uma temperatura corporal mais baixa.

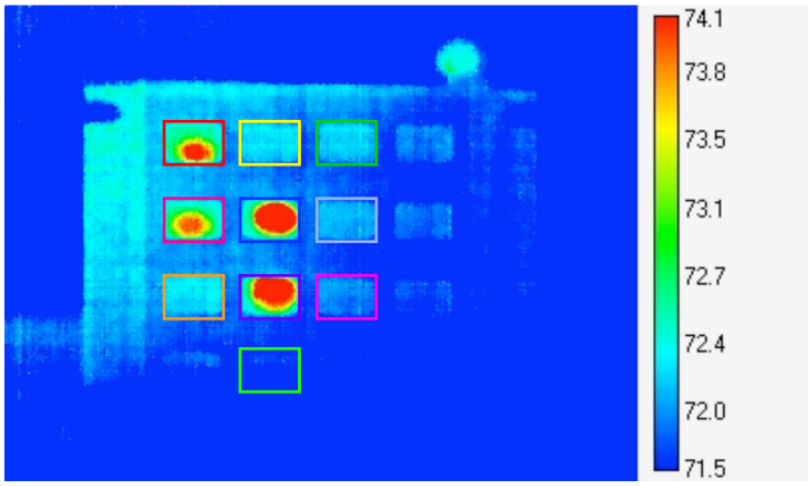


Figura 2.10: Captura realizada com a câmera termal, após a inserção de um código PIN, no momento em que a mão não estava mais presente no quadro. As dez áreas que representam as teclas são indicadas por caixas coloridas e as temperaturas são apresentadas na escala à direita. Observa-se que os dígitos 1, 4, 5 e 8 foram pressionados, sendo que os dígitos 5 e 8 provavelmente foram pressionados por último, pois apresentam maior temperatura.

Outro trabalho intitulado "*Security Impact of High Resolution Smartphone Camera*" [5] aborda o uso de câmeras de celulares para a detecção de entrada de dados como código PINs. Os autores utilizam as câmeras frontais, comumente encontrados em vários modelos de *smartphones* atuais, para capturar imagens da reflexão da tela do celular encontrada nos olhos dos sujeitos inserindo o código. O trabalho determina o tamanho do teclado nas imagens gravadas levando em conta principalemente a qualidade da resolução da câmera e a distância entre o teclado e o olho do sujeito. Ao localizar o teclado na imagem, uma imagem de referência do teclado é sobreposto para facilitar a identificação da entrada de dados. Os autores realizam experimentos pedindo para um sujeito tentar identificar os números do código PIN inseridos, na sequência correta, a partir da exibição das imagens gravadas. Os resultados mostram que dos quatro códigos testados, dois foram corretamente identificados na primeira tentativa pelos sujeitos.

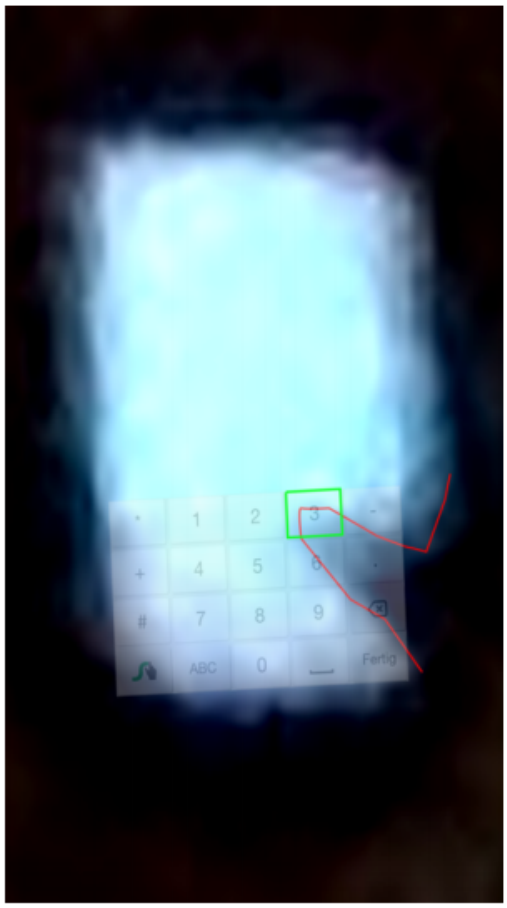


Figura 2.11: Captura do reflexo do olho com sobreposição da imagem de referência do teclado. A captura foi realizada com a câmera do celular OPPO N1 de 13 *megapixels*.

# Capítulo 3

# Solução Proposta

## 3.1 Design do experimento

O modelo de experimento adotado para este trabalho foi estruturado da seguinte forma: primeiramente uma câmera foi montada em um tripé para gravar os voluntários inserindo seus códigos PINs em dispositivos móveis. Após a aquisição das imagens, os voluntários assistiam a uma gravação, diferente da sua, e tentavam identificar os código PINs inseridos. Por fim, um algoritmo foi desenvolvido para processar os dados adquiridos com o objetivo de tentar localizar o teclado utilizado. O experimento realizado é composto por quatro etapas: aquisição de dados, verificação manual dos dados, verificação automática do teclado e análise dos dados. As primeiras três etapas serão discutidas neste capítulo e a última será interpretada no seguinte capítulo.

## 3.2 Aquisição de dados

A aquisição das imagens foi realizada utilizando uma *webcam* de modelo *Microsoft LifeCam Studio* montada em um tripé. O tripé foi colocado sobre uma mesa e a *webcam* foi posicionada perpendicularmente a uma altura de aproximadamente um metro de distância da mesa. Os teclados utilizados para este experimento foram os *touch-screens* de dois dispositivos móveis: o primeiro foi um tablet *iPad 2* da *Apple* com tela de 9,7 polegadas e resolução de 768 x 1024 pixels e o segundo foi um *smartphone iPhone 5* também da *Apple* com tela de 4 polegadas e resolução de 640 x 1136 pixels. Os vídeos gravados possuem resolução de 640 x 480 pixels com *frame rate* de 30 quadros por segundo e padrão MPEG-4 [11].

Os sujeitos foram filmados sentados em uma cadeira com o dispositivo à sua frente. Cada voluntário foi delegado a inserir cinco códigos PINs de quatro dígitos gerados aleatoriamente. A forma e a velocidade em que digitavam os códigos era de critério dos próprios

sujeitos, segundo seus costumes. Devido a natureza da autenticação dos dispositivos, que bloqueia o dispositivo após quatro tentativas inválidas, algumas medidas adicionais foram tomadas. No caso do *iPad*, o número de dígitos necessários para o desbloqueio do aparelho foi estendido para seis, requerendo o sujeito teclar o botão "apagar" após a inserção de cada código para evitar o bloqueio. Já no caso do *iPhone*, cada voluntário teve que inserir um PIN adicional, o código "verdadeiro" que desbloqueava o *smartphone*, após a inserção das primeiras três sequências de números.

Ao encerrar a filmagem dos códigos, cada voluntário realizou uma verificação manual de uma das gravações de um outro sujeito com o objetivo de identificar os número digitados e sua ordem. Devido ao posicionamento da câmera em relação aos aparelhos, foi necessário rotacionar o vídeo em 180° antes de sua exibição para facilitar a visualização dos PINs inseridos.

Além das gravações, foi elaborado um questionário com o intuito de determinar:

- qual a porcentagem de pessoas que de fato utilizam o sistema de autenticação por código PIN em algum dispositivo;
- quantas pessoas tem o costume de notar a presença de câmeras de vigilância em estabelecimentos;
- se as pessoas possuem o hábito de tomar alguma providência para dificultar o registro da inserção de seus PINs; e
- se as pessoas acreditam que o sistema PIN é confiável.

O questionário foi respondido por uma amostra de 39 pessoas por meio do sistema de formulários *online* do *Google Forms*.

## 3.3 Processamento dos dados

O primeiro tipo de processamento realizado sobre os vídeos foi a estimação de movimento. A estimação de movimento permite reduzir a quantidade de informação presente em cada quadro e identificar as regiões com maior ou menor movimento. Com este fim, um programa em linguagem C++ utilizando as ferramentas presentes da biblioteca OpenCV foi desenvolvido. O algoritmo recebe como entrada o nome completo do vídeo com a extensão que deve ser processado. O algoritmo compara dois *frames* sequênciais realizando *backward prediction*, assim, blocos de pixels de um quadro, denominado quadro "atual", são comparados com os blocos dentro de uma janela de busca no *frame* anterior. O tamanho do bloco utilizado foi de 8 x 8 pixels por ser um divisor relativamente pequeno da resolução dos quadros. A largura da janela de busca no quadro anterior, denotado

como *swl* (*search window length*), foi determinado a partir da largura do bloco, *bl* (*block length*) como sendo:

$$swl = 2 \times bl \times m + bl,$$

em que $m$ é uma constante inteira. Para este trabalho, $m$ possui valor igual a três. Devido ao uso da janela de busca, que realiza a procura em uma área retangular em volta do bloco, houve a necessidade de adicionar bordas em volta do quadro anterior, para não exceder os limites de tamanho da matriz.

As imagens gravadas são coloridas utilizando o sistema de cores RGB composta por matrizes tridimensionais, onde cada componente de cor é representado por oito bits. Para facilitar a comparação entre *frames*, realizou-se a conversão das imagens coloridas para a escala de cinza, também representada por oito bits. Assim, cada pixel é representado por um único número inteiro entre 0 e 255.

Após a comparação de todos os blocos na janela de busca e a deteminação do vetor de movimento, o bloco é inserido em uma matriz na mesma posição do bloco no quadro atual. Ao finalizar todas as comparações entre os dois *frames*, a matriz gerada constitiu o *prediction frame*. Este quadro de predição é subtraído do quadro atual gerando o *frame* residual o qual passará por um outro tipo de processamento.

O processamento das imagens residuais da estimação de movimento é realizado por outro programa, também na linguagem C++ e com o uso das ferramentas da biblioteca OpenCV. Este algoritmo realiza uma segmentação da imagem por meio da técnica de *thresholding*, que irá separar as regiões com maior intensidade de movimento. Estas regiões de interesse são identificadas como sendo os pixels de maior valor, as mais próximas da cor branca, como ilustra a figura Figura 3.1.

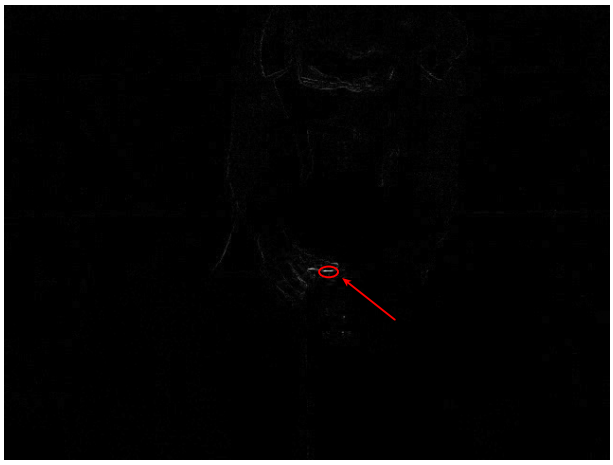

Figura 3.1: Exemplo de imagem residual identificando os pixels a serem considerados por meio do *thresholding* binário.

O valor do limiar escolhido foi 190, assim qualquer pixel com este valor ou menor, é mapeado para zero e qualquer valor acima é mapeado para o valor máximo, 255. O resultado da segmentação das imagens residuais são imagens binárias que foram utilizadas para formar uma única imagem resultante por meio da operação *bitwise* lógica "*or*". A Figura 3.2 mostra o resultado do *thresholding* e a Figura 3.3 mostra um exemplo do resultado da operação lógica para todos os *frames* segmentados de uma das gravações.

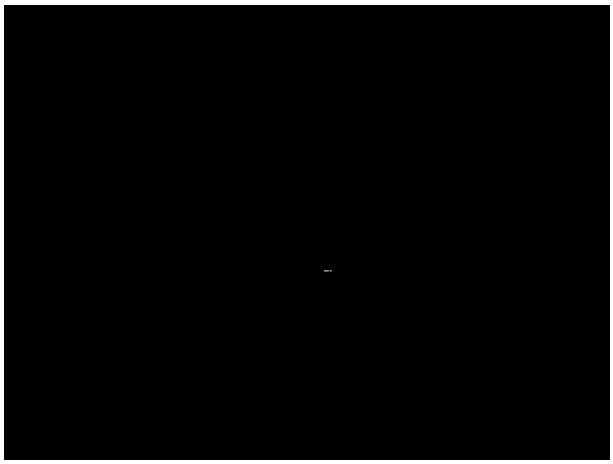

Figura 3.2: Imagem binária resultante do *thresholding*.

Figura 3.3: Imagem binária resultante da operação lógica "*or*" entre todos os *frames* resultantes da segmentação.

A imagem resultante da operação lógica é utilizada como entrada em um terceio programa com as mesmas características dos primeiros dois. Neste programa, a imagem de entrada passa por um processamento morfológico para preencher as cavidades presentes sem modificar o tamanho dos conjuntos. A operação morfológica utilizada foi, portanto,

a de fechamento com um elemento estruturante elíptico preenchido de tamanho 5 x 5 pixels com centro no ponto $(3, 3)$. A Figura 3.4 apresenta o resultado do fechamento da Figura 3.3.

Figura 3.4: Imagem binária resultante da operação morfológica de fechamento sobre a imagem da Figura 3.3.

Observa-se que a imagem resultante do fechamento apresenta muito ruído que deve ser eliminado. Assim, uma última operação morfológica de erosão é realizada com o objetivo de reduzir o ruído remanescente e tentar separar melhor os conjuntos maiores restantes. A Figura 3.5 ilustra a repercusão desta operação sobre a Figura 3.4.

Figura 3.5: Imagem binária resultante da operação morfológica de erosão sobre a imagem da Figura 3.4.

O algoritmo continua calculando o somatório dos pixels em cada linha e em cada coluna da última imagem. O programa utiliza esta informação para tentar estimar a

localização aproximada do centro do teclado. A estimação da localização é realizada de forma iterativa e é calculada por meio de uma média ponderada, definida como

$$\frac{\sum_i^n P_i \times N_i}{N},$$

em que $i$ e $n$ são os índices iniciais e finais respectivamente, $P_i$ representa a i-ésima linha ou coluna, $N_i$ representa o peso associado, ou seja, a soma dos valores dos pixels nesta posição, e $N$ a soma total dos pesos. Inicialmente, toda a extensão do quadro é considerada e a posição é estimada. A cada nova iteração o tamanho da região de consideração é reduzida a uma àrea em volta da posição estimada por último. Esta área é determinada, tomando uma porcentagem da região considerada por último e subtraindo-a da posição estimada, para obter o novo índice inicial, e adicionando-a a posição, para obter o novo índice final. A localização é determinada quando a diferença entre as últimas duas posições estimadas for menor do que 1.

Uma vez determinada a localização central do teclado, o algoritmo estima seus limites da localização. Para isto, o programa verifica a soma de cada posição antes e depois da localização central. Caso a soma seja igual a um quinto da soma de pixels da posição central, o limite é encontrado. Caso a soma seja menor que um quinto, a posição anterior é tomada como o limite. Por fim, o algoritmo esboça os limites da localização calculados do teclado sobre uma imagem da gravação original que possui o teclado, como ilustra a imagem da Figura 3.6.

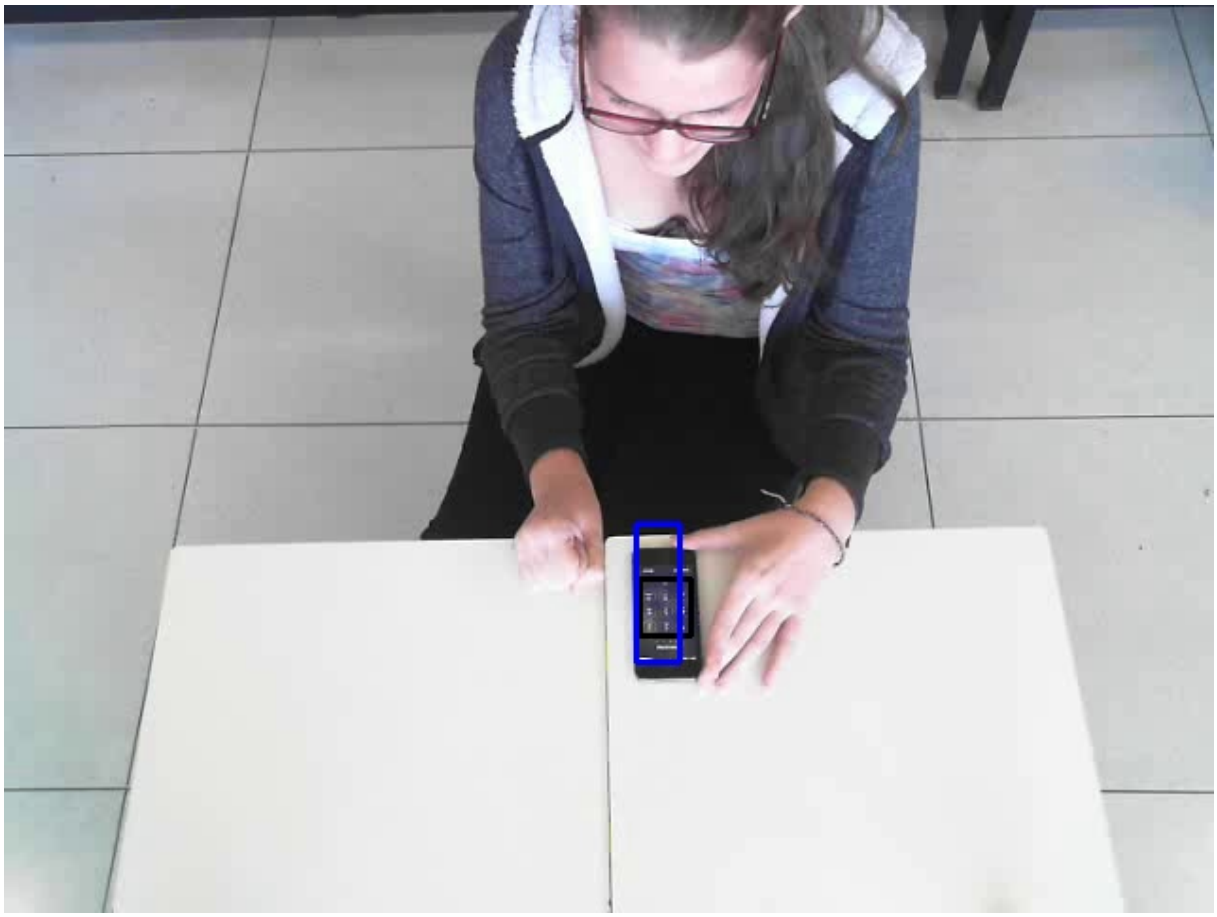

Figura 3.6: *Frame* da gravação original com o teclado delimitado em preto, desenhado manualmente, e a estimação da localização do teclado, realizado pelo algoritmo, em azul.

# Capítulo 4

# Resultados Experimentais

## 4.1 Resultados do questionário

O questionário realizado é composto por dez questões, em que cada pergunta requeria selecionar apenas uma das respostas dadas. As respostas variaram entre responder "Sim" ou "Não" e escolher entre cinco alternativas, aquela que melhor representava a realidade da pessoa. As primeiras três questões tinham como objetivo determinar em quais dispositivos as pessoas utilizam com frequência o sistema PIN e verificar a relevância deste sistema de autenticação atualmente. Os resultados mostram que mesmo que a maioria das pessoas entrevistadas utilizem o código PIN para seus dispositivos móveis, cerca de 60%, e na autorização de transações bancárias com cartões, mais do que 80%, como ilustram as Figura 4.1 e Figura 4.2.

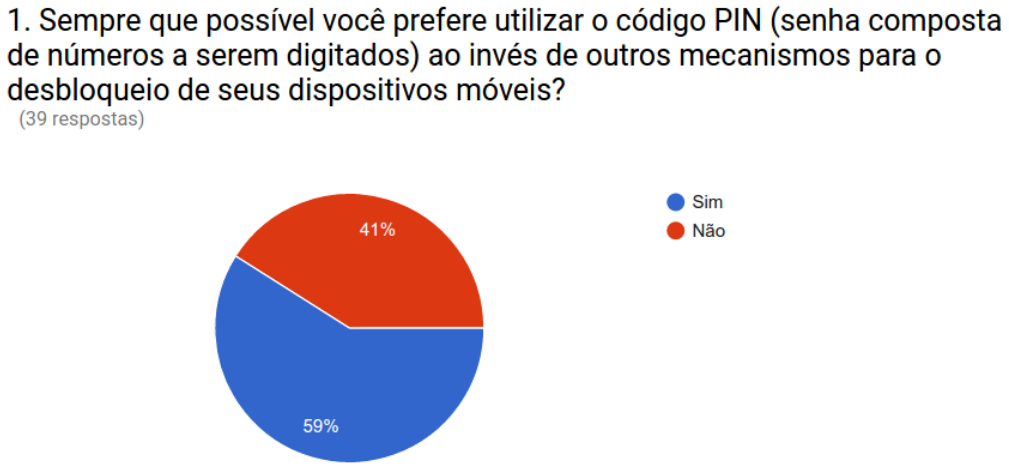


Figura 4.1: Resultado da primeira pergunta do questionário realizado.

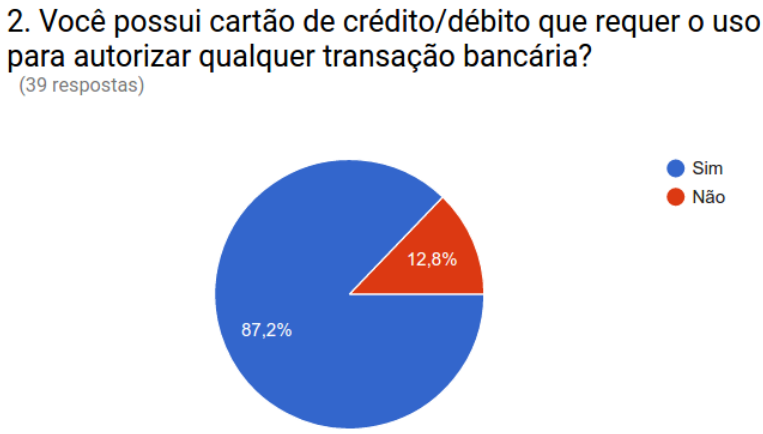


Figura 4.2: Resultado da segunda pergunta do questionário realizado.

Além destes dispositivos mais comuns, mais de um terço das pessoas utilizam este tipo de sistema em outros meios, conforme pode ser visto pela Figura 4.3. Estes resultados demonstram que o sistema é frequentemente utilizado atualmente e é o sistema de autenticação de preferência da maioria dos entrevistados.


Figura 4.3: Resultado da terceira pergunta do questionário realizado.

A quarta e quinta questão do questionário visavam determinar o comportamento das pessoas em estabelecimentos vigiados por câmeras de segurança. Esta verificação de comportamento permite determinar se existe uma preocupação, por parte das pessoas, em

serem gravadas, e, se a presença das câmeras as deixa com maior sentimento de segurança. Esta informação é útil para indicar a probabilidade das pessoas de realizarem algum esforço ativo para impedir ou dificultar o registro de seus PINs nestes ambientes. Os resultados evidenciam que cerca de 70% das pessoas nunca ou raramente tomam consciência das câmeras e se sentem relativamente seguras nestes estabelecimentos, conforme é evidenciado nas Figura 4.4 e Figura 4.5.

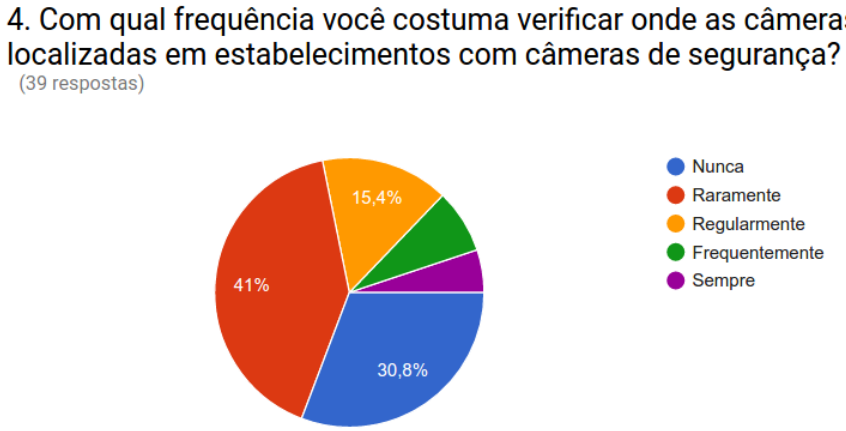


Figura 4.4: Resultado da quarta pergunta do questionário realizado.

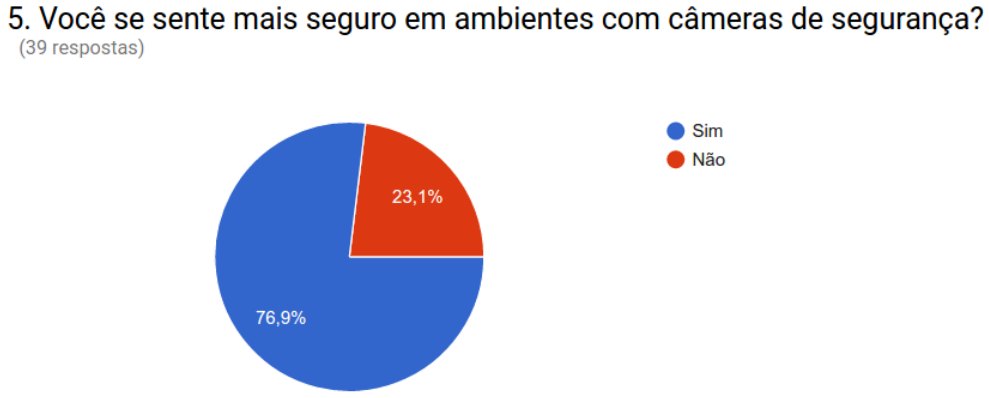


Figura 4.5: Resultado da quinta pergunta do questionário realizado.

Já as questões 5 e 6 foram elaboradas para avaliar se as pessoas de fato realizam algum esforço para impedir o registro de seus PINs. Os resultados mostram que menos de 5% das pessoas frequentemente tomam providências para evitar de serem gravados ao inserirem seus PINs. Estes resultados deixam claro que um modelo de ataque envolvendo o uso de câmeras de vigilância tem alta probabilidade de conseguir filmar os PINs dado o posicionamento favorável das mesmas. Os resultados são exibidos nas Figura 4.6 e Figura 4.7.

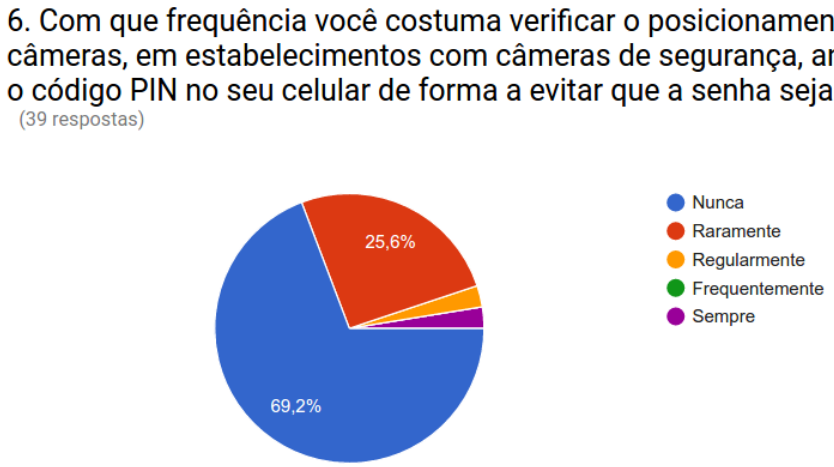


Figura 4.6: Resultado da sexta pergunta do questionário realizado.

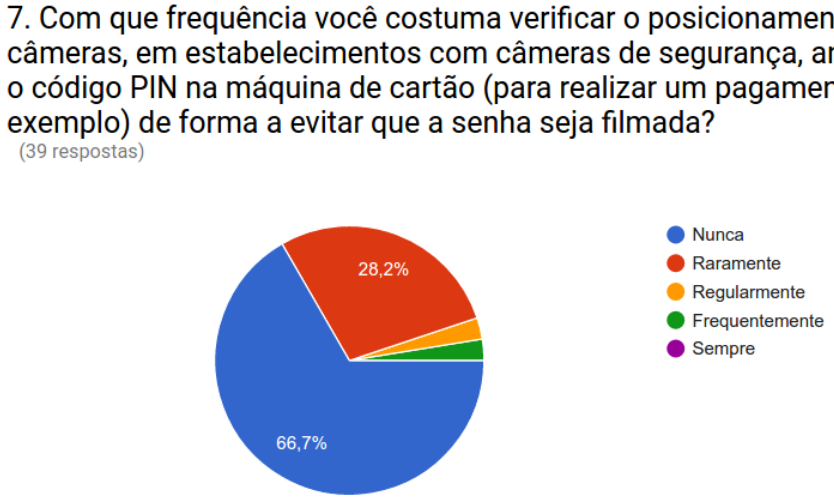


Figura 4.7: Resultado da sétima pergunta do questionário realizado.

As últimas três questões avaliam o grau de confiança das pessoas no sistema PIN, certificando os cenários em que os PINs de fato foram gravados. Os resultados mostram que a grande maioria não confia no sistema PIN, mas cerca de 25% das pessoas ainda afirmam que não correram riscos e confiam que os estabelecimentos não farão mau uso das informações. De forma geral, as pessoas avaliaram positivamente o grau de seguraça do sistema. Os resultados são apresentados nas Figura 4.8, Figura 4.9 e Figura 4.10.

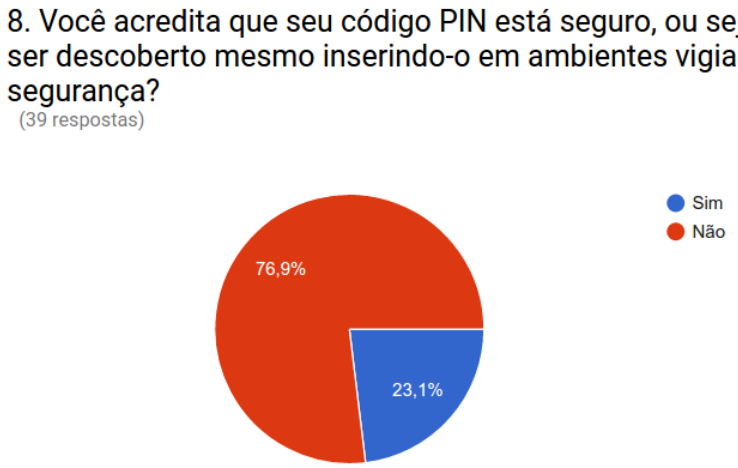


Figura 4.8: Resultado da oitava pergunta do questionário realizado.

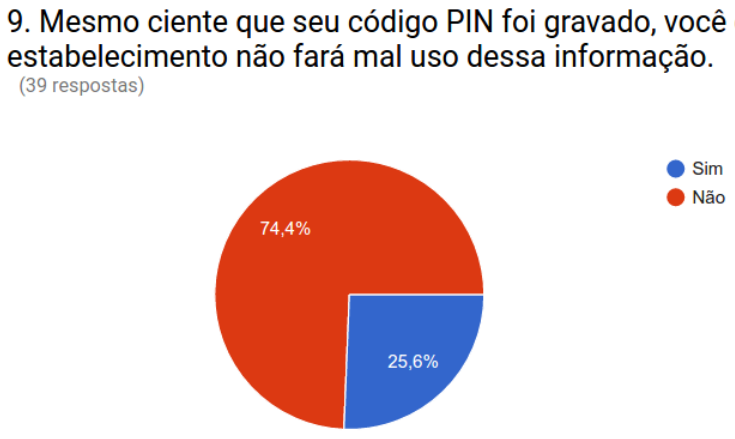


Figura 4.9: Resultado da nona pergunta do questionário realizado.

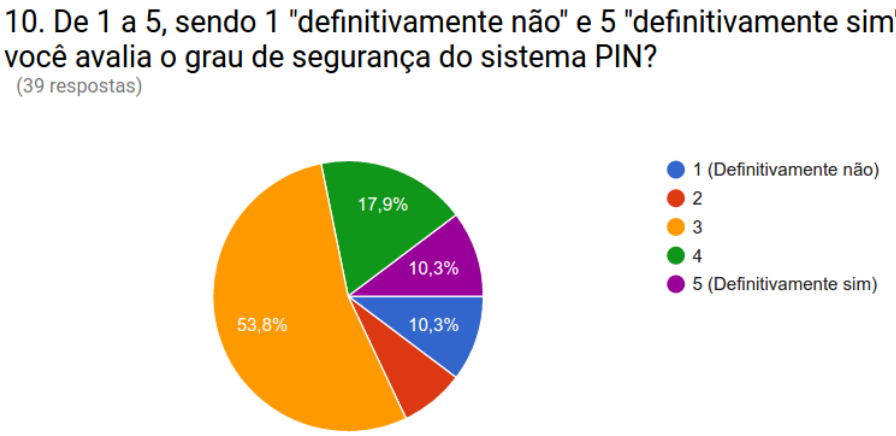


Figura 4.10: Resultado da décima pergunta do questionário realizado.

## 4.2 Resultados da solução proposta

### 4.2.1 Validação do algoritmo de *block-matching*

Um caso de teste foi elaborado para verificar o funcionamento correto do algoritmo de estimação de movimento. O teste consistiu em utilizar duas imagens testes iguais e inserir um bloco de ruído branco em ambas, mas em posições distintas relativamente próximas (dentro da janela de busca). O bloco possui tamanho de 8 x 8 e o valor de seus pixels é gerado aleatoriamente. O teste utiliza imagens idênticas e o ruído aditivo gaussiano de média média nula e variância, pois ao comparar qualquer bloco da imagem com o ruído branco, a diferença seria máxima por ser um padrão totalmente aleatório e não uniforme. Da mesma forma, ao comparar o bloco de ruído branco de uma imagem com o bloco de ruído branco na outra, a diferença seria igual a zero e em qualquer outro bloco o resultado seria muito maior. O padrão gerado é apresentado na Figura 4.11 e as duas imagens com os padrões estão presentes nas Figura 4.12 e Figura 4.13. O algoritmo apresenta como saída a posição em que encontrou o bloco de rúido branco na imagem anterior de referência e a distância entre o primeiro pixel do bloco nesta imagem e a posição do primeiro pixel do bloco na imagem atual. Foi verificado que o bloco foi localizado na posição $(24, 24)$ e a distância euclidiana, definida como sendo

$$distância = \sqrt{(x_1 + x_2)^2 + (y_1 + y_2)^2}$$

em que $x_1$ e $y_1$ são as coordenadas do primeiro pixel do bloco na primeira imagem (atual) e $x_2$ e $y_2$ o primeiro pixel do bloco encontrado na segunda imagem (referência), foi de

22, 6274.

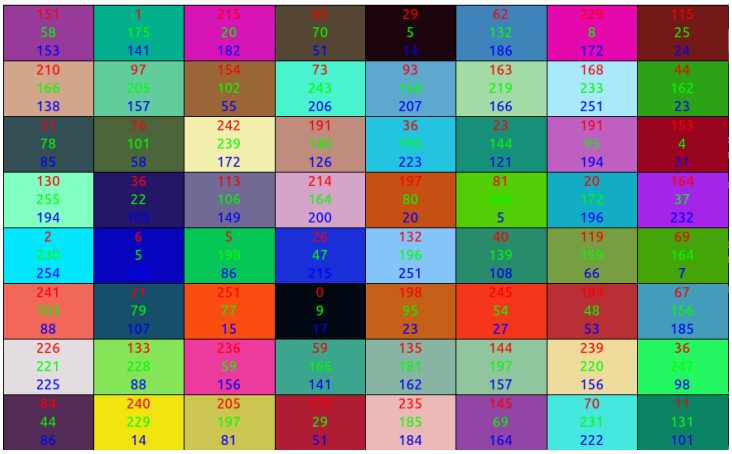

Figura 4.11: Padrão do ruído branco mostrando os valores de cada componente RGB.

Figura 4.12: Primeira imagem de teste utilizada como *frame* atual na comparação. O padrão aleatório aparece na posição $(2, 2)$ da imagem.

O algoritmo de estimação de movimento, inicialmente, foi desenvolvido de forma que as imagens resultantes seriam compostas a partir dos menores valores de módulo das distâncias entre os blocos de menor diferença dos dois quadros sendo comparados. Observou-se que este método era inviável, pois as imagens possuiam muitos locais onde a diferença

Figura 4.13: Segunda imagem de teste utilizada como *frame* anterior de referêcnia na comparação. O padrão aleatório aparece na posição $(24, 24)$ da imagem.

entre regiões próximas era muito pequena, assim muita energia residual persistia. A Figura 4.14 demonstra o resultado deste algoritmo.

Figura 4.14: Imagem composta pelo módulo das distâncias entre blocos. Observe-se a alta taxa de energia residual evidenciado pelos pixels mais próximos do branco.

### 4.2.2 Resultados da verificação manual dos voluntários

As capturas realizadas foram exibidas aos voluntários de forma que cada um analizou apenas uma captura de um outro voluntário. Ao finalizar a análise da gravação, o voluntário informou quais foram os cinco códigos PINs inseridos. O voluntário pode assistir ao mesmo vídeo mais de uma vez para certificar que havia identificado corretamente a sequência de dígitos. Dois critérios foram utilizados para validar a verificação: caso o voluntário tenha acertado todos os dígitos inseridos e caso o voluntário tenha acertado a ordem dos dígitos. Em todos os vídeos os voluntários tiveram êxito em identificar os digítos inseridos, sendo que em apenas três deles os voluntários erraram a ordem dos dígitos. Em cada um destes três vídeos, apenas um dos cincos códigos inseridos não foi identificado corretamente. Assim, dos 50 códigos registrados, apenas 3 foram incorretamente identificados acarretando uma taxa de acerto de 94% para a verificação manual. Esse resultado é importante, pois indica a viabilidade da utilização de um algoritmo para realizar a estimação da localização do teclado. Caso o resultado não apresentasse uma alta taxa de acerto, a automatização do processo seria improvável.

### 4.2.3 Análise comparativa

Esta subseção do capítulo realiza uma análise comparativa dos dados coletados. Serão analisados os resultados de duas gravações realizadas com o *smartphone* e os resultados de duas gravações realizadas com o *tablet*. São realizadas análises individuais das peculiaridades de cada gravação, comparações das similaridades e diferenças observadas entre as gravações com o mesmo dispositivo e entre os dispositivos diferentes.

O *thresholding* binário realizado permitiu identificar e isolar as regiões com maior movimento de cada frame. A razão de realizar este tipo de segmentação deve-se ao fato de que o agente que realiza maior movimento durante o vídeo é a mão, justamente durante a inserção do PIN. Este movimento ocorre predominantemente sobre o teclado, ou em uma região próxima, permitindo localiza-o. Ao sobrepor cada um destes resultados da segmentação por meio da operação lógica "or" o formato do teclado do *smartphone* torna-se evidente como pode-se observar nas imagens Figura 4.15, Figura 4.16.

Contudo, a segmentação não foi suficiente para isolar a localização do teclado em uma única região. As figuras anteriores mostram que ainda há muito ruído e algumas regiões a serem retiradas. A Figura 4.17 e a Figura 4.18 mostram as concentrações de pixels em cada linha e cada coluna da Figura 4.15. Já a Figura 4.19 e a Figura 4.20 mostram as mesmas concentrações da Figura 4.16.

O gráfico da Figura 4.17 apresenta dois picos distintos de concentração de pixels. O primeiro pico representa o conjunto de pixels brancos mais próximos ao topo da imagem

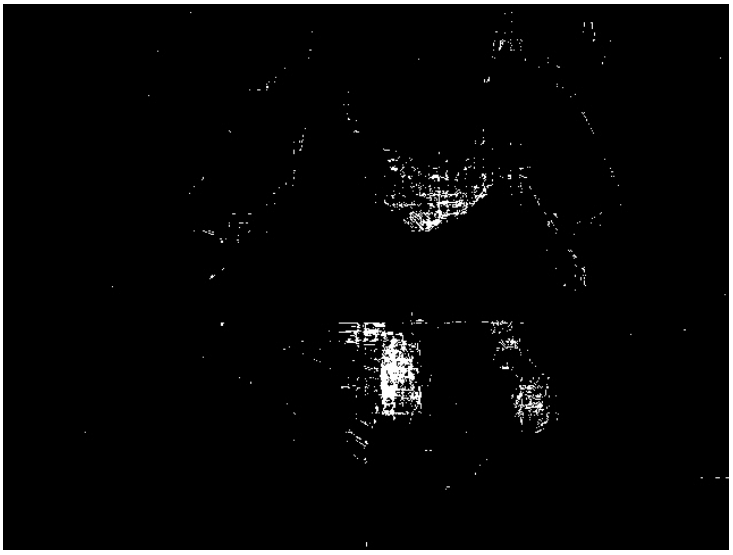

Figura 4.15: Imagem composta da sobreposição dos resultados do *thresholding*. O formato retangular do teclado do *smartphone* é evidenciado pelo conjunto de pixels brancos.

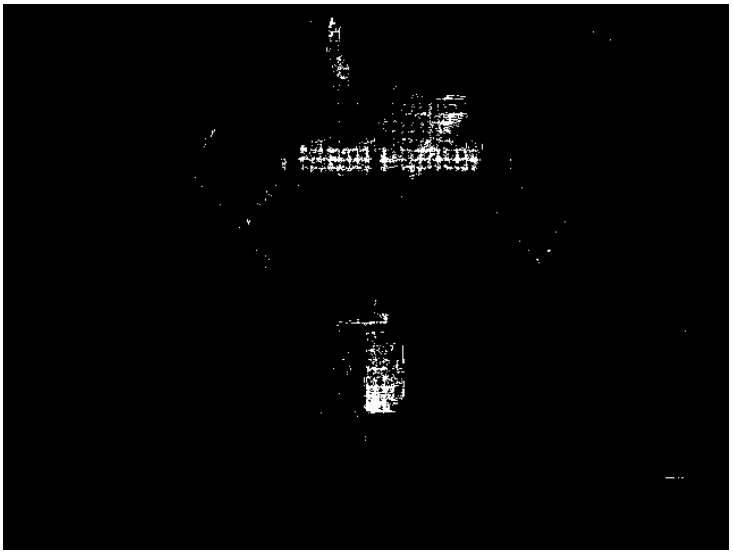

Figura 4.16: Outro exemplo do resultado da sobreposição. Novamente é possível observar o formato retangular do dispositivo móvel pelo conjunto de pixels brancos na região inferior da imagem.

e descrevem principalmente o movimento da cabeça do sujeito quando desvia o seu olhar do teclado para o papel com os códigos PINs a serem inseridos (presente fora do *frame* do vídeo). Já o segundo pico possui alguns pontos maiores e é mais largo que o primeiro, representando uma concentração maior de pixels. Este segundo pico deve-se principalmente ao movimento das mãos sobre o teclado.

O gráfico da Figura 4.18 apresenta novamente dois picos de concentração de pixels.

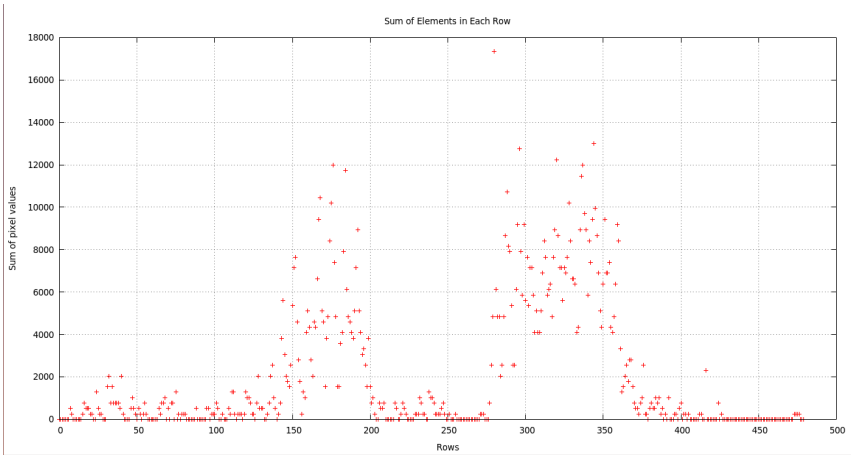


Figura 4.17: Gráfico que apresenta a soma dos valores dos pixels de cada linha da Figura 4.15. A origem do eixo das abscissas se refere à primeira linha do topo da imagem e os demais valores do eixo se referem às linhas subsequentes.

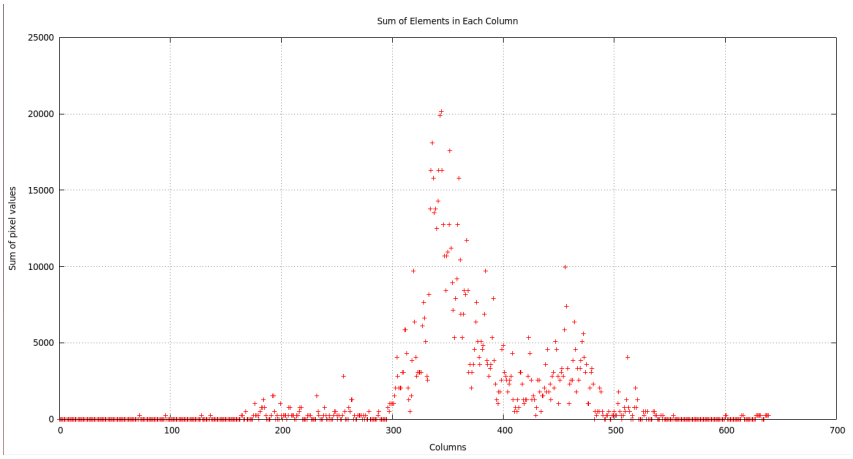


Figura 4.18: Gráfico que apresenta a soma dos valores dos pixels de cada coluna da Figura 4.15. A origem do eixo das abscissas se refere à primeira coluna do lado direito da imagem e os demais valores do eixo se referem às colunas subsequentes.

O primeiro pico é significativamente maior que o segundo e localizado aproximadamente no centro. Este pico particularmente é formado pela concentração de pixels presentes na região do teclado e em parte pela região de movimento da cabeça. O segundo pico, por sua vez, possui concentração mais baixa de pixels, formado por movimentos na região da cabeça e pelo conjunto de pixels brancos ao lado direito da região do teclado. Este último conjunto de pixels indicam os movimentos realizados pelo braço e pela mão esquerdos do sujeito nos instantes em que o voluntário bloqueia e desbloqueia o dispositivo.

O gráfico apresentado na Figura 4.19 exibe duas regiões distintas também. A primeira região é caracterizada por poucos pontos de valores bastantes elevados de pixels. Esta

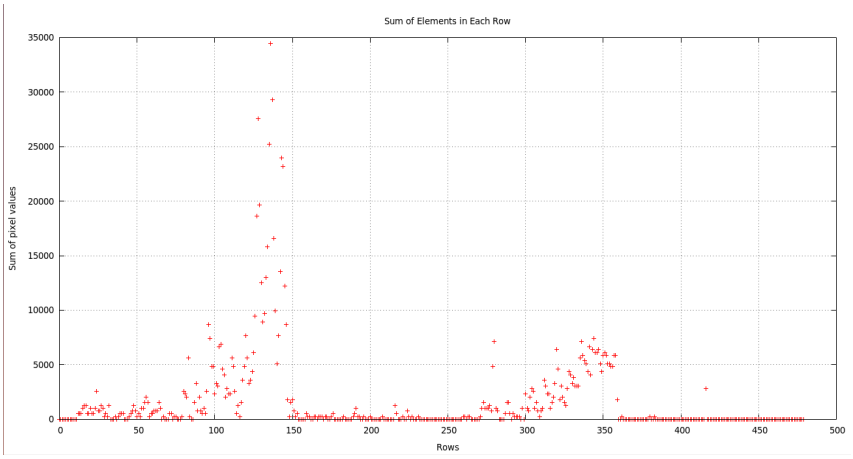


Figura 4.19: Gráfico que apresenta a soma dos valores dos pixels de cada linha da Figura 4.16. A origem do eixo das abscissas se refere à primeira linha do topo da imagem e os demais valores do eixo se referem às linhas subsequentes.

região, referente ao conjunto de pixels da parte superior da imagem, representa o movimento realizado pela cabeça e pelo deslocamento para frente (em direção ao *smartphone*) e para trás (na direção oposta do dispositivo móvel) realizado para obter uma visualização mais nítida dos códigos PINs a serem inseridos. A segunda região apresenta uma concentração maior de pixels do que a primeira, mas de valores significativamente menores. Este resultado deve-se ao fato do sujeito segurar o dispositivo com as duas mãos e os movimentos serem realizados principalmente por alguns dedos da mão direita. Assim, os valores dos pixels nesta região são menores devido a variação pequena deste movimentos e por serem bastantes concentrados em uma só área.

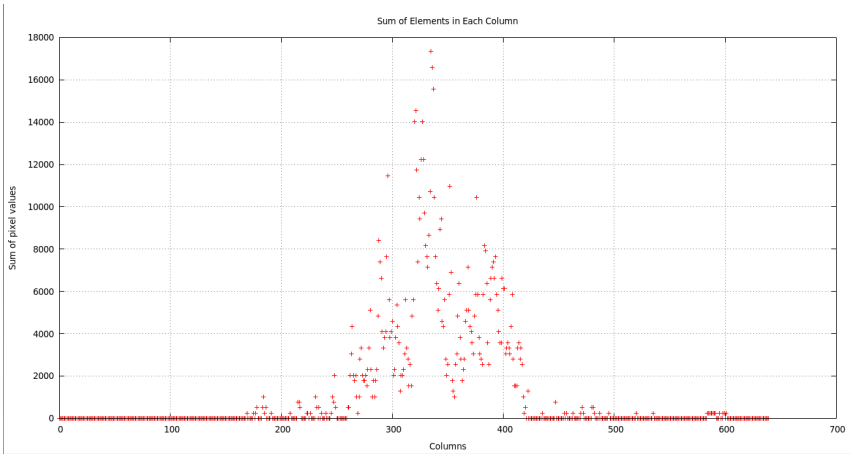


Figura 4.20: Gráfico que apresenta a soma dos valores dos pixels de cada coluna da Figura 4.16. A origem do eixo das abscissas se refere à primeira coluna do lado direito da imagem e os demais valores do eixo se referem às colunas subsequentes.

O gráfico presente na Figura 4.20 demonstra que o movimento ocorre predominantemente na região central do quadro. Os maiores picos ocorrem nos locais em que há movimento tanto dos dedos sobre o teclado quanto da parte superior do sujeito.

Observando as duas imagens e os seus gráficos, nota-se que há uma maior movimentação generalizada na primeira imagem do que na segunda. O teclado torna-se mais evidente na primeira do que na segunda devido aos movimentos realizados com toda a mão direita ao invés dos movimentos limitados dos dedos, como foi realizado na segunda. Por este motivo também, a primeira imagem se apresenta bem mais ruído do que a segunda. Em qualquer caso, não é possível ainda encontrar a localização exata do teclado em cada imagem com apenas este processamento.

A Figura 4.21 e a Figura 4.22 mostram as imagens geradas a partir da mesma operação de sobreposição dos resultados do *thresholding* binário para o tablet *iPad*. Nota-se nestas imagens também, a alta presença de ruído e o formato retangular do dispositivo.

Figura 4.21: Imagem composta da sobreposição dos resultados do *thresholding*. O formato retangular do teclado do *tablet* é evidenciado pelo conjunto de pixels brancos na parte inferior central da imagem.

A mesma análise da concentração de pixels em cada linha e coluna, realizada com o *iPhone*, foi realizada com o *iPad*. Os gráficos da Figura 4.23 e da Figura 4.24 mostram a concentração dos pixels em cada linha e coluna, respectivamente, da Figura 4.21.

O gráfico da Figura 4.23 demonstra duas regiões principais de concentração de pixels na imagem Figura 4.21. A primeira região está localizada na parte superior da imagem e, ao analisar a gravação original, percebe-se que esta região deve-se principalmente pela movimentação da cabeça e da mão direita enconstada na face. Frequentemente a mão é afastada de perto da cabeça e logo em seguida retorna para encostar no rosto da própria

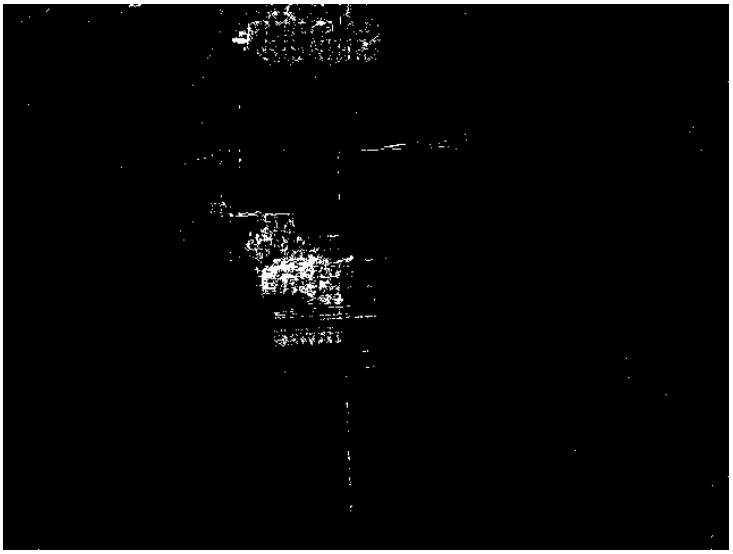

Figura 4.22: Imagem composta da sobreposição dos resultados do *thresholding*. O formato retangular do teclado do *tablet* é evidenciado pelo conjunto de pixels brancos na parte inferior central da imagem.

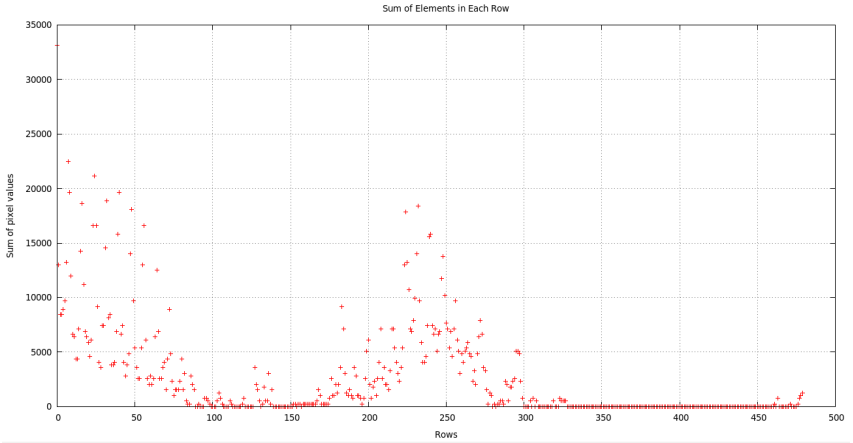


Figura 4.23: Gráfico que apresenta a soma dos valores dos pixels de cada linha da Figura 4.21. A origem do eixo das abscissas se refere à primeira linha do topo da imagem e aos demais valores do eixo referem às linhas subsequentes.

pessoa. A segunda região, desta vez localiza próximo ao centro da imagem, representa o movimento da mão sobre o teclado ao inserir os códigos PINs.

O gráfico da Figura 4.24 apresenta uma concentração maior de pixels na região central com pequenas variações ao redor. Ao lado esquerdo da região central existe uma menor, porém significativa, concentração de pixels provavelmente indicativa da movimentação realizada pela mão.

Os gráfico da Figura 4.25 e da Figura 4.26 mostram a concentração dos pixels em

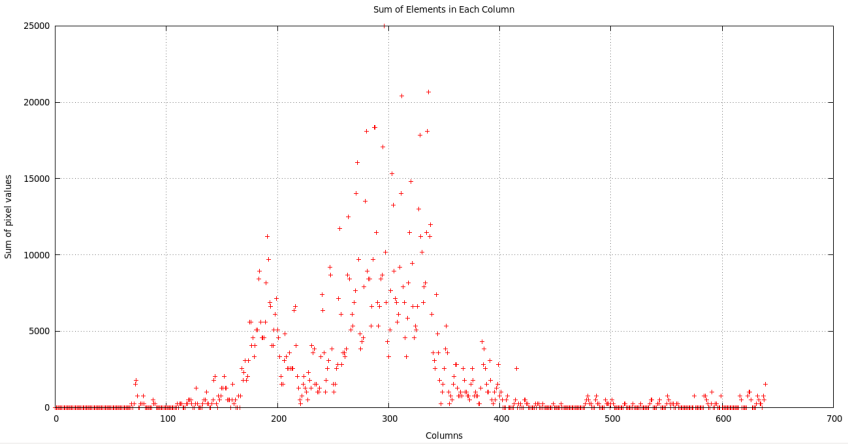


Figura 4.24: Gráfico que apresenta a soma dos valores dos pixels de cada coluna da Figura 4.21. A origem do eixo das abscissas se refere à primeira coluna do lado direito da imagem e os demais valores do eixo se referem às colunas subsequentes.

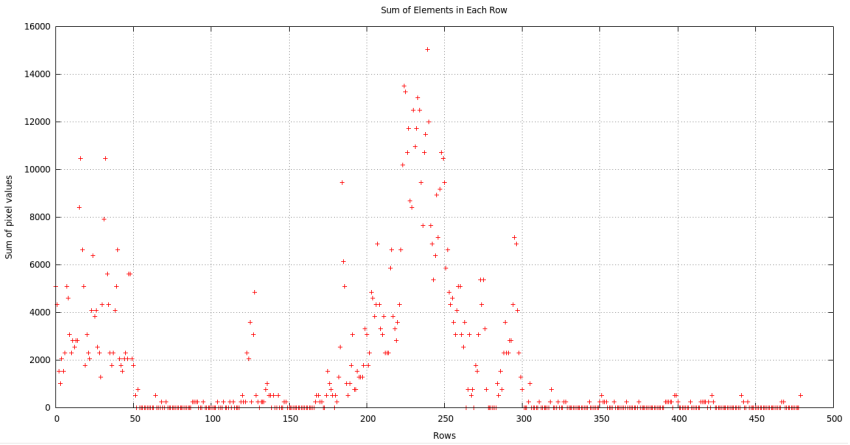


Figura 4.25: Gráfico que apresenta a soma dos valores dos pixels de cada linha da Figura 4.22. A origem do eixo das abscissas se refere à primeira linha do topo da imagem e os demais valores do eixo se referem às linhas subsequentes.

cada linha e coluna, respectivamente, da Figura 4.22. Uma concentração maior de pixels é localizada na região central do gráfico da Figura 4.25 oriunda da movimentação da mão direita sobre o teclado do dispositivo. Assim como em diversas outras imagens, encontra-se uma região na parte superior da imagem com quantidades significativas de pixels, região próxima a origem do gráfico, indicando o movimento realizado predominantemente pela cabeça do sujeito.

O gráfico presente na Figura 4.26 demonstra que todos os movimentos realizados estão concentrados próximos à região central da imagem, indicando que há muito pouco ou quase nenhum movimento realizado nas laterais do *frame*.

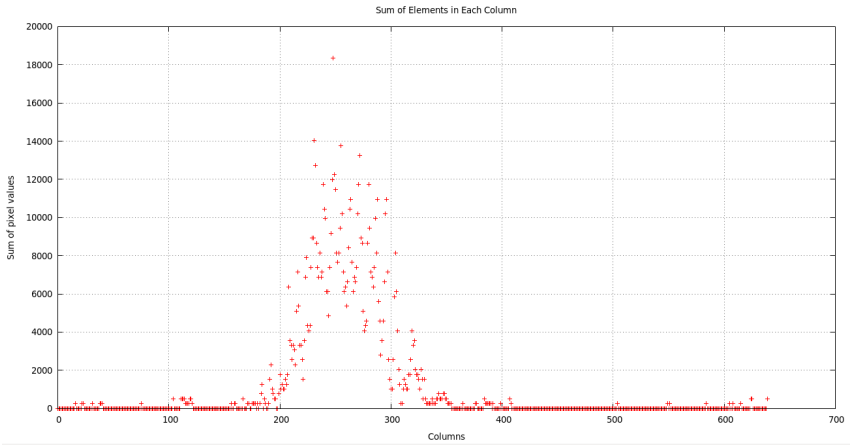


Figura 4.26: Gráfico que apresenta a soma dos valores dos pixels de cada coluna da Figura 4.22. A origem do eixo das abscissas se refere à primeira coluna do lado direito da imagem e os demais valores do eixo se referem às colunas subsequentes.

A próxima etapa de processamento realizada teve como objetivo preencher as cavidades dos conjuntos presentes nas imagens anteriores e tentar determinar o local aproximado dos teclados de cada dispositivo. Por este motivo foi utilizado a operação morfológica de fechamento. As imagens presentes na Figura 4.27 e Figura 4.28 mostram os resultados desta operação para o *iPhone*. Já as imagens da Figura 4.29 e da Figura 4.30 mostram os resultados da mesma operação para o *iPad*.

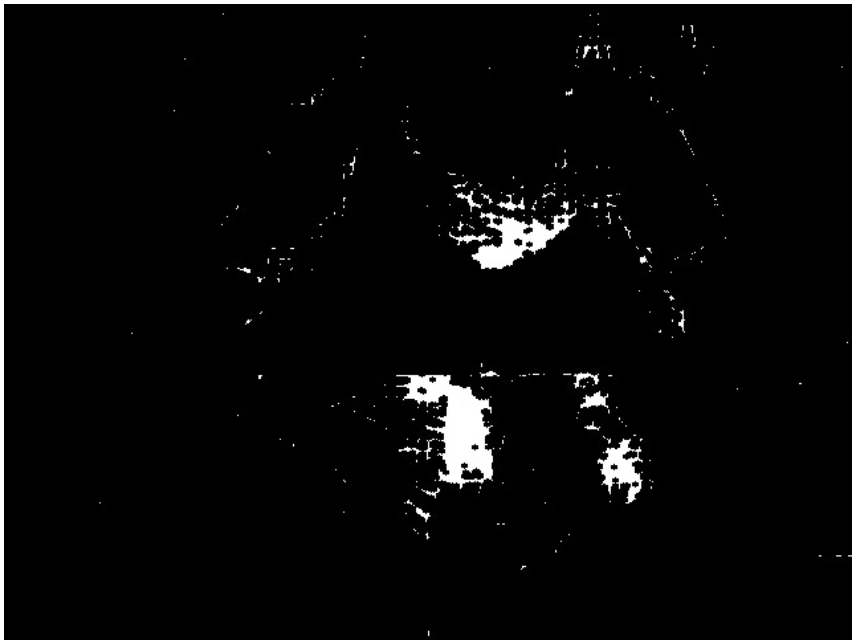

Figura 4.27: Imagem resultante da operação de fechamento sobre a Figura 4.15.

Figura 4.28: Imagem resultante da operação de fechamento sobre a Figura 4.16.

Figura 4.29: Imagem resultante da operação de fechamento sobre a Figura 4.16.

Em seguida a erosão foi realizada para eliminar o ruído remanescente. Ao final de todo estes processamentos, foi realizada a estimação da localização do teclado. Novamente foi realizada uma análise da concentração de pixels em cada linha e cada coluna das imagens erodidas. Utilizou-se a média ponderada para realizar a estimação, pois a região com

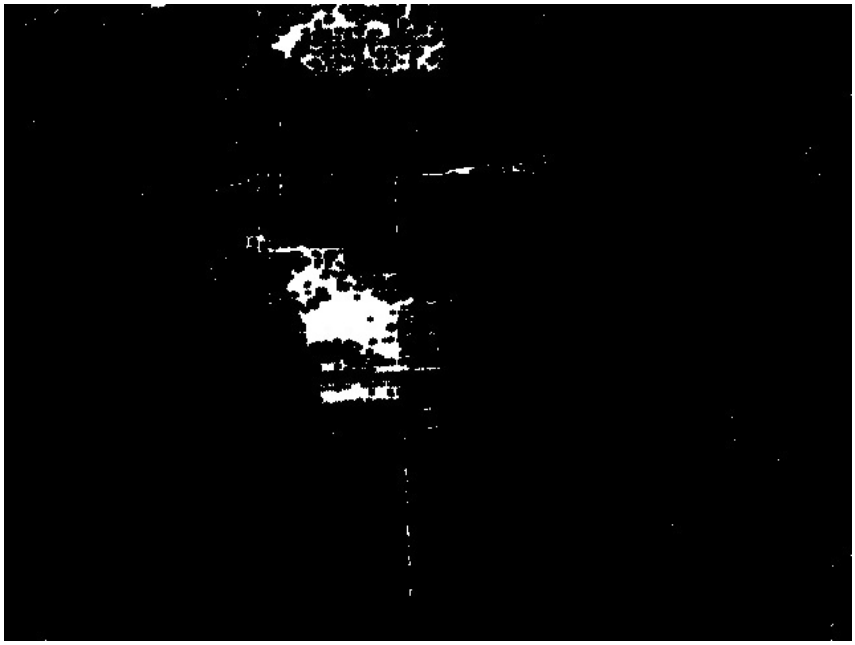

Figura 4.30: Imagem resultante da operação de fechamento sobre a Figura 4.21.

maior concentração de pixels, supostamente a região do teclado, predominaria sobre as demais localidades e, assim, seria a escolhida. A posição encontrada pela estimação foi definida como sendo o centro do teclado. As imagens da Figura 4.31 e da Figura 4.32 mostram os resultados da erosão sobre as gravações com o *smartphone*. Já as imagens da Figura 4.33 e da Figura 4.34 mostra os resultados da erosão sobre as gravações com o *tablet*.

Os limites, que representam onde cada teclado deve ser localizado, foram definidos a partir da análise da concentração de pixels ao redor do centro escolhido. Foi determinado empiricamente que os melhores resultados para os limites eram as posições onde a concentração dos pixels era próxima a um quinto da concentração encontrada na posição central. Os gráficos da Figura 4.35 e Figura 4.36 demonstram os limites encontrados a partir da análise das linhas para as gravações com o *iPhone*. Os gráficos da Figura 4.37 e da Figura 4.38 mostram os limites encontrados para as gravações com o *iPad*, a partir da mesma análise. Já análise a partir das colunas são evidenciadas pelos gráficos da Figura 4.39 e Figura 4.40 para as gravações com o *smartphone* e pelos gráficos da Figura 4.41 e Figura 4.42 para as gravações com o *tablet*.

Por fim, as imagens da Figura 4.43 e da Figura 4.44 mostram os resultados do algoritmo para as gravações com o *iPhone* e as imagens da Figura 4.45 e da Figura 4.44 mostram os resultados do algoritmo para as gravações realizadas com o *iPad*. A taxa de acerto da estimação do teclado foi definida como sendo a razão entre a região de interseção, da área do teclado definida manualmente e da área estimada pelo algoritmo, e a própria área do

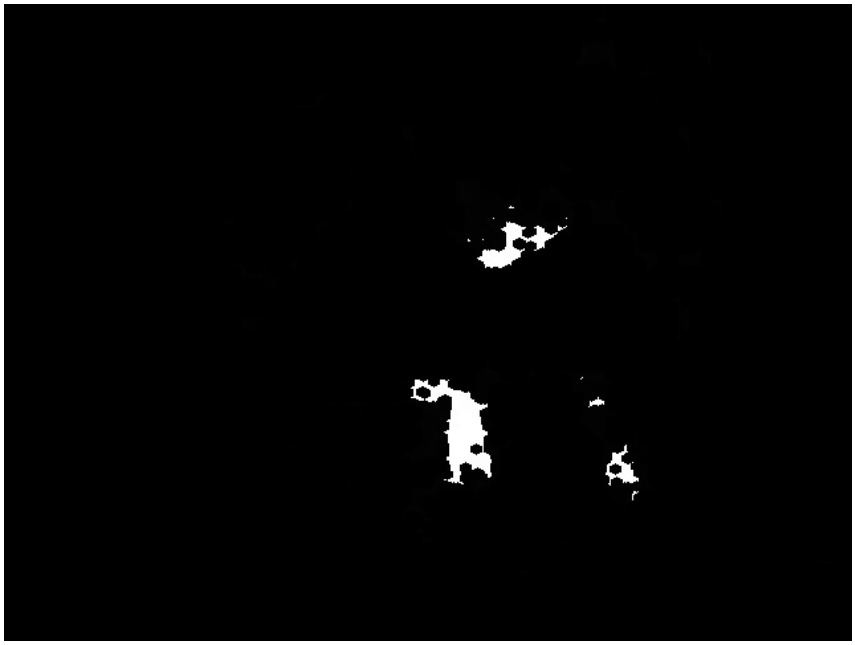

Figura 4.31: Imagem resultante da operação de erosão sobre a Figura 4.27.

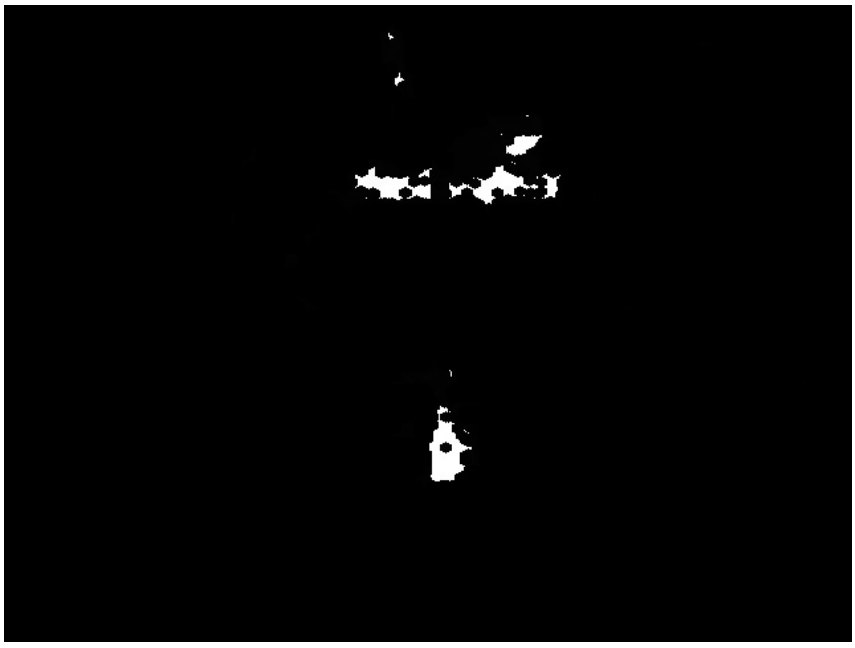

Figura 4.32: Imagem resultante da operação de erosão sobre a Figura 4.28.

teclado definida manualmente.

Figura 4.33: Imagem resultante da operação de erosão sobre a Figura 4.29.

Figura 4.34: Imagem resultante da operação de erosão sobre a Figura 4.30.

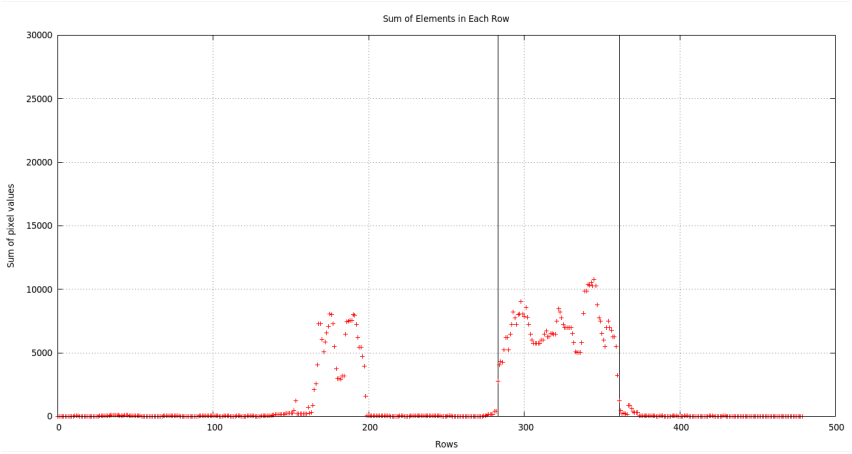


Figura 4.35: Gráfico que apresenta a soma dos valores dos pixels de cada linha da Figura 4.31. A origem do eixo das abscissas se refere à primeira linha do topo da imagem e os demais valores do eixo se referem às linhas subsequentes. As setas indicam os limites inferiores e superiores da localização do teclado encontrados pelo algoritmo.

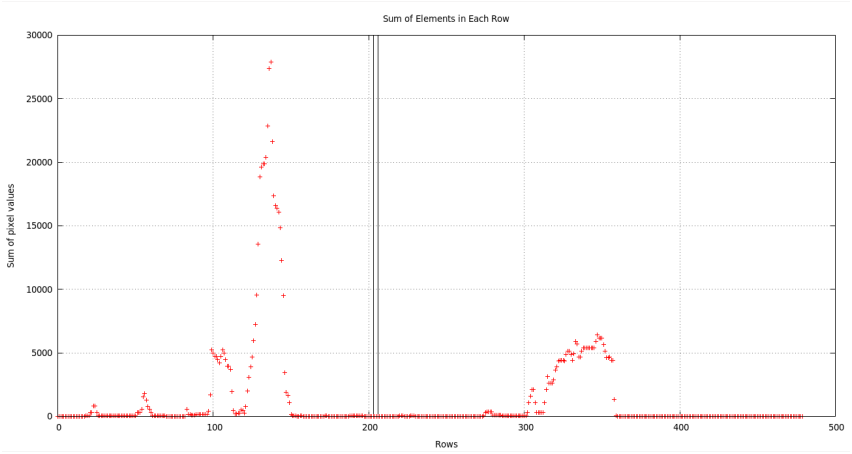


Figura 4.36: Gráfico que apresenta a soma dos valores dos pixels de cada linha da Figura 4.32. A origem do eixo das abscissas se refere à primeira linha do topo da imagem e os demais valores do eixo referem às linhas subsequentes. As setas indicam os limites inferiores e superiores da localização do teclado encontrados pelo algoritmo.

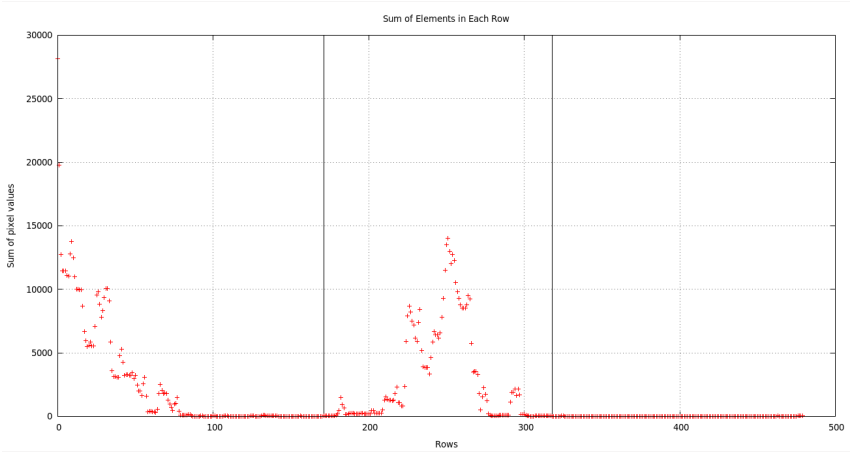


Figura 4.37: Gráfico que apresenta a soma dos valores dos pixels de cada linha da Figura 4.33. A origem do eixo das abscissas se refere à primeira linha do topo da imagem e os demais valores do eixo referem às linhas subsequentes. As setas indicam os limites inferiores e superiores da localização do teclado encontrados pelo algoritmo.

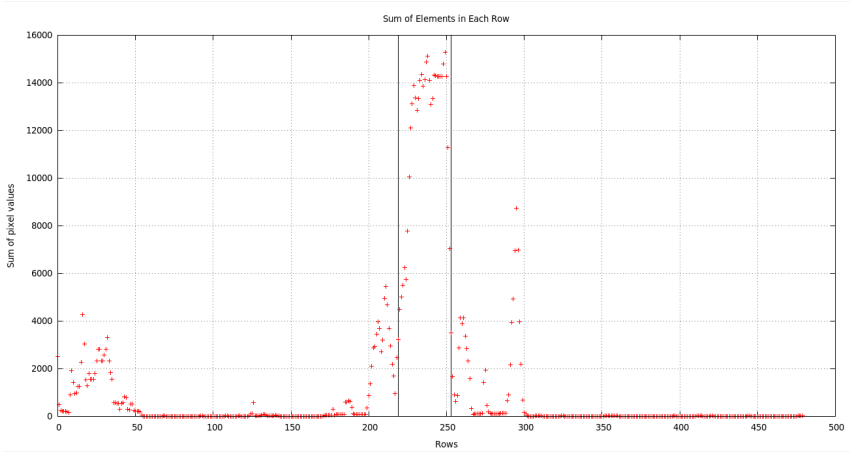


Figura 4.38: Gráfico que apresenta a soma dos valores dos pixels de cada linha da Figura 4.34. A origem do eixo das abscissas se refere à primeira linha do topo da imagem e os demais valores do eixo referem às linhas subsequentes. As setas indicam os limites inferiores e superiores da localização do teclado encontrados pelo algoritmo.

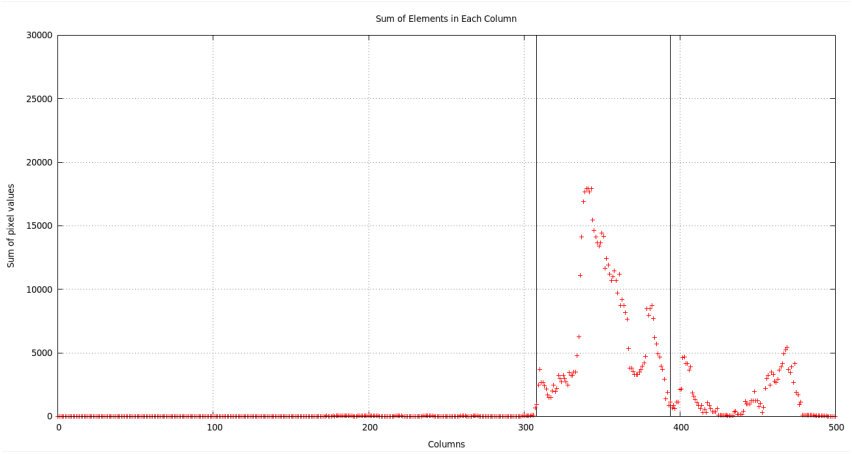


Figura 4.39: Gráfico que apresenta a soma dos valores dos pixels de cada coluna da Figura 4.31. A origem do eixo das abscissas se refere à primeira coluna do topo da imagem e os demais valores do eixo referem às colunas subsequentes. As setas indicam os limites inferiores e superiores da localização do teclado encontrados pelo algoritmo.

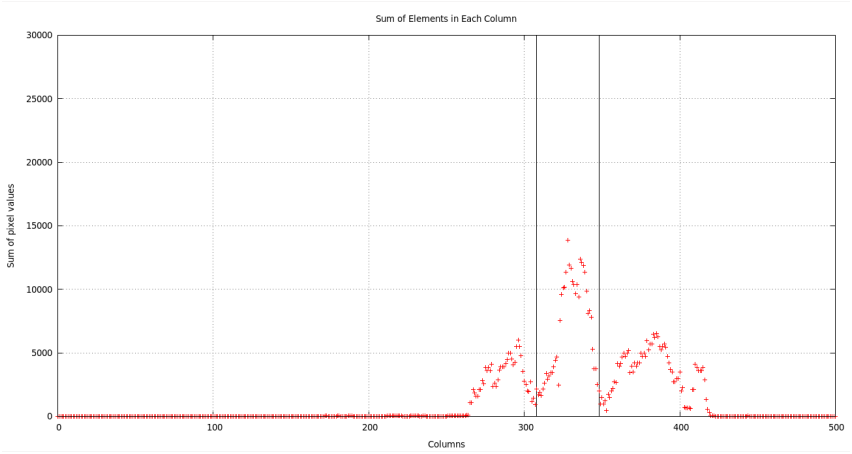


Figura 4.40: Gráfico que apresenta a soma dos valores dos pixels de cada coluna da Figura 4.32. A origem do eixo das abscissas se refere à primeira coluna do topo da imagem e os demais valores do eixo referem às colunas subsequentes. As setas indicam os limites inferiores e superiores da localização do teclado encontrados pelo algoritmo.

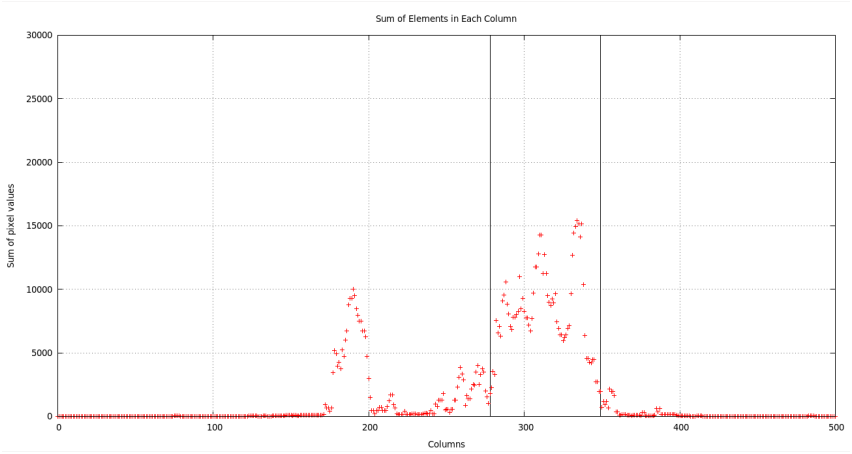


Figura 4.41: Gráfico que apresenta a soma dos valores dos pixels de cada coluna da Figura 4.33. A origem do eixo das abscissas se refere à primeira coluna do topo da imagem e os demais valores do eixo referem às colunas subsequentes. As setas indicam os limites inferiores e superiores da localização do teclado encontrados pelo algoritmo.

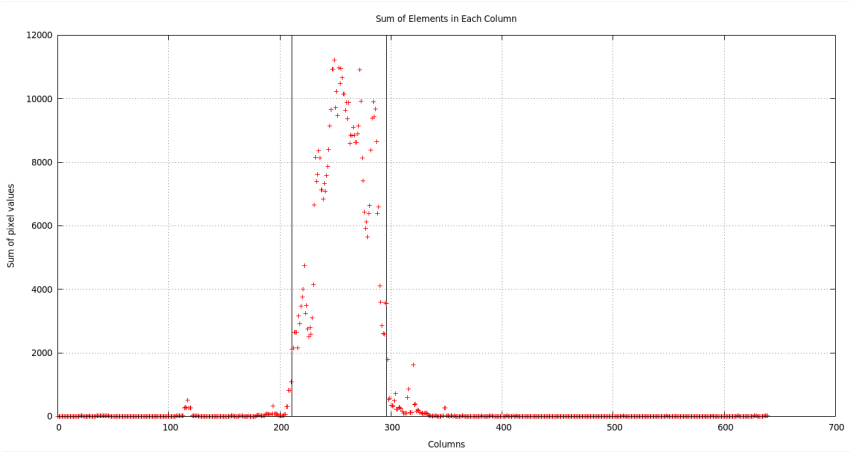


Figura 4.42: Gráfico que apresenta a soma dos valores dos pixels de cada coluna da Figura 4.34. A origem do eixo das abscissas se refere à primeira coluna do topo da imagem e os demais valores do eixo referem às colunas subsequentes. As setas indicam os limites inferiores e superiores da localização do teclado encontrados pelo algoritmo.

Figura 4.43: Imagem ilustra a estimação da localização do teclado realizado pelo algoritmo, em azul, e a localização real do teclado, em preto. Observa-se que neste caso a localização real do teclado está totalmente contida na região estimada pelo algoritmo.

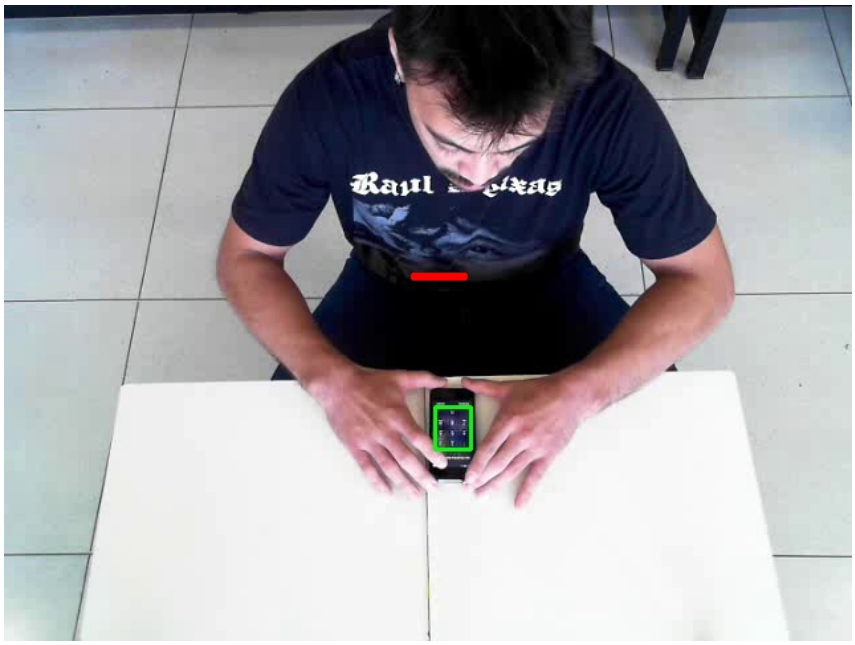

Figura 4.44: Imagem ilustra a estimação da localização do teclado realizado pelo algoritmo, em vermelho, e a localização real do teclado, em verde. Observa-se que neste caso a estimação não foi realizada com sucesso, devido a grande quantidade de movimento realizado pela parte superior do corpo do sujeito e a restrição dos movimentos realizados apenas pelos dedos sobre o teclado.

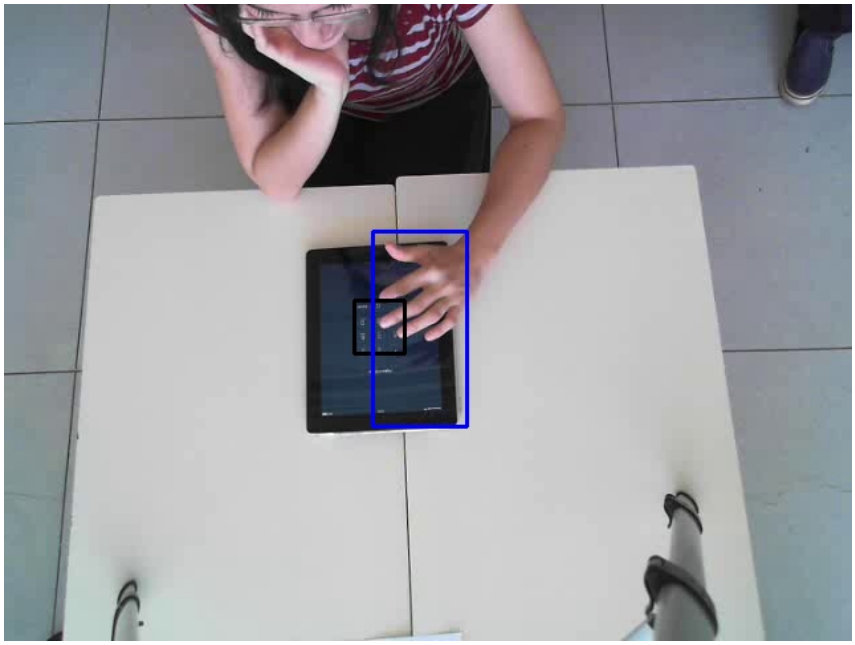

Figura 4.45: Imagem ilustra a estimação da localização do teclado realizado pelo algoritmo, em azul, e a localização real do teclado, em preto. Observa-se que neste caso que houve uma região de interseção entre a região estimada e a localização real do teclado. Esta região de interseção representa cerca de 63,2% da região real total do teclado.

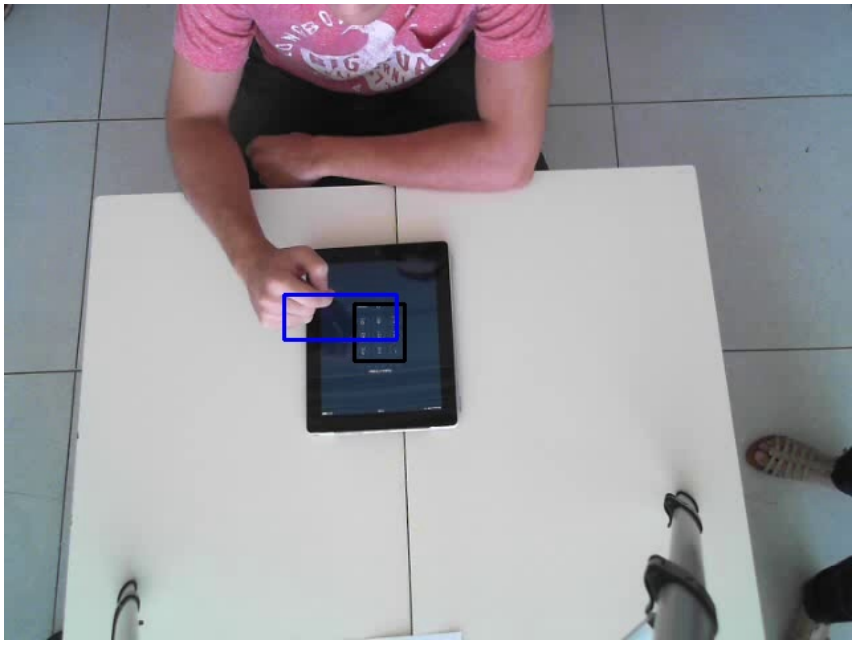

Figura 4.46: Imagem ilustra a estimação da localização do teclado realizado pelo algoritmo, em azul, e a localização real do teclado, em preto. Observa-se neste caso que houve novamente uma região de interseção entre a região estimada e a localização real do teclado. Esta região de interseção representa cerca de 52,9% da região real total do teclado.

# Capítulo 5

# Conclusões e Trabalhos Futuros

O presente trabalho explorou os conceitos de processamento de vídeo e imagens digitais a fim de desenvolver um algoritmo capaz de extrair a localização de teclados de dispositivos móveis a partir dos registros realizados por uma câmera. O experimento realizado teve como objetivo simular uma eventual situação de um estabelecimento com câmeras de vigilância e demonstrar a vulnerabilidade do sistema de autenticação por código PIN. O algoritmo desenvolvido pode ser utilizado como passo inicial para a extração de códigos PINs de um modelo de ataque baseado no uso de câmeras.

Os resultados coletados mostraram que a abordagem adotada não conseguiu localizar o teclado em todas as gravações analisadas, mas conseguiu determinar regiões em que há a presença de mais de 50% do teclado. Esta informação é bastante valiosa e facilita a futura extração dos códigos. O algoritmo de estimação de movimento é de alto custo computacional, podendo demorar várias horas mesmo para vídeos de curta duração, mas o uso do *full-search* justifica-se por sempre apresentar os melhores resultados. Um ataque baseado neste modelo poderá optar por substituir o uso desta técnica por uma estratégia mais rápida, como o *three-step search*, em troca de uma pequena perda na acurácia da localização, de forma a torná-lo mais escalonável. Os resultados também mostraram que a forma em que os sujeitos inserem o código, por exemplo, movimentando apenas os seus dedos, e a presença de outros elementos em movimento, influenciaram na precisão da localização do teclado. Os resultados do questionário realizado demonstraram que a maioria das pessoas sentem-se mais seguras em ambientes vigiados por câmeras e, portanto, não realizam grandes esforços para impedir que registrem seus PINs, mesmo preocupados com as possíveis consequências desta ocorrência. Esse comportamento deixa evidente a vulnerabilidade do sistema devido ao descaso por parte do usuário. Esta informação torna o modelo de ataque mais viável e apresenta maiores chances de sucesso.

Em trabalhos futuros, a detecção dos números do teclado poderá ser o próximo passo a ser explorado. Além disto, pode-se mapear o movimento realizado pelas mãos a fim de

extrair o código PIN inserido. A segmentação por cor de pele sobre a região do teclado pode ser um tipo de processamento útil para a detecção automática do código. Realizar testes posicionando a câmeras em diferentes localizações e a distâncias diferentes podem prover maiores informações para tornar a detecção mais robusta. No caso da estimação da localização do teclado que não foi bem sucedida, uma melhoria a ser realizada envolveria analizar a quantidade de pixels presentes na localização central estimada. Caso essa concentração fosse muito baixa, indicando que provalvemente a estimação não foi bem sucedida, o algoritmo deveria buscar ao seu redor regiões com maiores concentrações. Um método, a ser implementado em trabalhos futuros, poderia determinar que a região mais próxima da primeira estimação não efetiva com maior concentração de pixels seria a região mais apropriada para a localização do teclado.

# Referências


[1] *John shepherd-barron*, maio 2010. http://www.telegraph.co.uk/news/obituaries/culture-obituaries/7746651/John-Shepherd-Barron.html. 3

[2] Agoston, M. K.: *Computer Graphics and Geometric Modeling Implementations and Algorithms*. Springer, 2005. 11

[3] Brajatya, Aroh: *Block matching alogrithm for motion estimation*. Em *DIP 66620 Spring 2004 Final Project Paper*, 2004. 6

[4] Facon, J.: *A morfologia matemática e suas aplicações em processamento de imagens*. Em *VII Workshop de Visao Computacional – WVC 2011*, 2011. 13

[5] Fiebig, Tobias, Jan Krissler e Ronny Hansch: *Security impact of high resolution smartphone cameras*. Em *WOOT 2014*, 2014. https://www.usenix.org/system/files/conference/woot14/woot14-fiebig.pdf. 17

[6] Gonzalez, R. C. e R. E. Woods: *Digital Image Processing*. Prentice Hall, 2ª edição, 2001. ix, 4, 9

[7] Goodfellow, J. e A.I.O. Davies: *Access-control equipment*, maio 1967. https://goo.gl/YyH9jK. 3

[8] Jeffries, Stuart: *Pins and needle*, janeiro 2005. https://www.theguardian.com/money/2005/jan/20/accounts.saving. 3

[9] Jones, Rupert: *Who invented the cash machine? i did – and all i earned was £10*, abril 2016. https://www.theguardian.com/money/2016/apr/29/who-invented-cash-machine-james-goodfellow-first-atm-pin. 3

[10] K. Mowery, S. Meiklejohn e S. Savage.: *Heat of the Moment: Characterizing the efficacy of thermal camera-based attacks*. Em *Proceedings of WOOT 2011*, agosto 2011. 16

[11] Microsoft: *Lifecam studio*. Relatório Técnico, Microsoft, 2016. http://download.microsoft.com/download/0/9/5/0952776D-7A26-40E1-80C4-76D73FC729DF/TDS_LifeCamStudio.pdf. 19

[12] Pesquet-Popescu, Béatrice, Marco Cagnazzo e Frédéric Dufaux: *Academic Press Library in Signal Processing*, volume 5. Elsevier, 2014. https://cagnazzo.wp.mines-telecom.fr/files/2013/05/poly_me.pdf. 6

[13] Poynton, Charles: *Digital Video and HD Algorithms and Interfaces*. Elsevier, 2012. 9

[14] Progianti, Carlos: *Mercado avança com a evolução das câmeras inteligentes*, abril 2012. `http://www.abese.org.br/blog/?p=46`. 1

[15] PURVES, D, G. J. AUGUSTINE, D. FITZPATRICK, W. C. Hall, A. S. LAMANTIA, J. O. McNamara e S. M. Williams: *Neuoscience*. Sinauer Associates, 3ª edição, 2004. 9

[16] Richardson, Iain E.: *The H.264 Advanced Video Compression Standard*. John Wiley & Sons Ltd, 2010. 6

[17] Shapiro, L. G. e G. C Stockmang: *Computer Vision*. Prentice Hall, 2001. 12

# Apêndice A

# Questionário

# PIN e Câmeras de Vigilância

Este questionário faz parte do trabalho de conclusão de curso intitulado "Detecção Automática de PIN em Ambientes Vigiados por Câmeras" por Marcelo André Winkler do curso de Engenharia da Computação. O objetivo desse questionário é determinar o comportamento das pessoas em locais vigiados por câmeras (de segurança) em determinadas situações a fim de estabelecer possíveis vulnerabilidades existentes em sistemas de autenticação por código PIN (Personal Identification Number). Este sistema está presente em diferentes dispositivos como senhas de cartão de crédito, de desbloqueio de tela de celulares e de acesso de portas de segurança. Agradeço a sua participação!

*Obrigatório

1. Sempre que possível você prefere utilizar o código PIN (senha composta de números a serem digitados) ao invés de outros mecanismos para o desbloqueio de seus dispositivos móveis? *

- ◯ Sim
- ◯ Não

2. Você possui cartão de crédito/débito que requer o uso de um código PIN para autorizar qualquer transação bancária? *

- ◯ Sim
- ◯ Não

3. Além dos dispositivos citados acima, existem outros dispositivos que você utiliza que possuem autenticação por meio de código PIN (por exemplo, para abrir uma porta)? *

- ◯ Sim
- ◯ Não

4. Com qual frequência você costuma verificar onde as câmeras estão localizadas em estabelecimentos com câmeras de segurança? *

- ◯ Nunca
- ◯ Raramente
- ◯ Regularmente
- ◯ Frequentemente
- ◯ Sempre

5. Você se sente mais seguro em ambientes com câmeras de segurança? *

- ◯ Sim
- ◯ Não

6. Com que frequência você costuma verificar o posicionamento das câmeras, em estabelecimentos com câmeras de segurança, antes de digitar o código PIN no seu celular de forma a evitar que a senha seja filmada? *

- ◯ Nunca
- ◯ Raramente
- ◯ Regularmente
- ◯ Frequentemente
- ◯ Sempre

7. Com que frequência você costuma verificar o posicionamento das câmeras, em estabelecimentos com câmeras de segurança, antes de digitar o código PIN na máquina de cartão (para realizar um pagamento, por exemplo) de forma a evitar que a senha seja filmada? *

- ◯ Nunca
- ◯ Raramente
- ◯ Regularmente
- ◯ Frequentemente
- ◯ Sempre

8. Você acredita que seu código PIN está seguro, ou seja, não corre o risco de ser descoberto mesmo inserindo-o em ambientes vigiados por câmeras de segurança? *

- ◯ Sim
- ◯ Não

9. Mesmo ciente que seu código PIN foi gravado, você confia que o estabelecimento não fará mal uso dessa informação. *

- ◯ Sim
- ◯ Não

10. De 1 a 5, sendo 1 "definitivamente não" e 5 "definitivamente sim", como você avalia o grau de segurança do sistema PIN? *

- ◯ 1 (Definitivamente não)
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 (Definitivamente sim)

ENVIAR

---