

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE ELETRÔNICA
CURSO DE TECNOLOGIA EM AUTOMAÇÃO DE PROCESSOS INDUSTRIAIS

ALAN LEANDRO VILARINO
MARCELO GONÇALVES PETRALHA
WILSON GASPAR VEIGA

SARO – SISTEMA AUTOMATIZADO DE RECONHECIMENTO ÓTICO

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2012

ALAN LEANDRO VILARINO
MARCELO GONÇALVES PETRALHA
WILSON GASPAR VEIGA

SARO – SISTEMA AUTOMATIZADO DE RECONHECIMENTO ÓTICO

Trabalho de Diplomação de graduação, apresentado à disciplina de Trabalho de Diplomação, dos Cursos de Tecnologia em Automação de Processos Industriais e Tecnologia em Automação Industrial da Coordenação de Eletrônica – COELE – da Universidade Tecnológica Federal do Paraná – UTFPR, Campus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo em Automação de Processos Industriais.

Orientador: Prof. Fábio Brignol de Moraes

PATO BRANCO

2012

TERMO DE APROVAÇÃO

O trabalho de diplomação intitulado **SARO – SISTEMA AUTOMATIZADO DE RECONHECIMENTO ÓTICO**, dos alunos **ALAN LEANDRO VILARINO, MARCELO GONÇALVES PETRALHA E WILSON GASPAR VEIGA**, foi considerado **APROVADO** de acordo com a ata da banca examinadora N° **143** de 2013.

Fizeram parte da banca os professores:

André Macário Barros

Fábio Brignol de Moraes

Marcos Junior Marini

DEDICATÓRIA

Dedicamos esse trabalho de Diplomação aos nossos familiares, aos nossos professores, que nos apoiaram nesta longa jornada que até aqui chegamos. Com a satisfação de nos tornamos pessoas melhores e mais qualificadas para a sociedade.

AGRADECIMENTOS

Muitas vezes acreditamos que só sonhar não basta, temos que realizar esse sonho, pois sonhar nós deixa fortes, esperançosos, sonhamos em fazer uma faculdade, sonho esse realizado com muito suor, dedicação, com dificuldades. Não foi fácil chegarmos até aqui, onde muitos dos nossos amigos não conseguiram chegar até o objetivo final, mas lutaram bravamente.

Nossos sinceros agradecimentos a aqueles que na corrida pelo sucesso não conseguiram realizar o tão esperado sonho, de um dia se tornarem Tecnólogos, espero que eles nunca desistam de seus sonhos. Agradecemos também a aqueles que conseguiram concluir seus objetivos e se tornarem Tecnólogos.

Agradecemos o professor Fábio Brignol de Moraes, pela dedicação, pelo empenho e total compreensão, disponibilizado assim o tempo necessário para nós orientar. Fazendo como possível a realização de mais uma etapa de nossas vidas.

Nossos sinceros agradecimentos aos nossos familiares, que sempre nos apoiaram, acreditaram que íamos conseguir chegar até o final, mesmo sabendo que o caminho percorrido era longo e complicado. Muito obrigado a todos que acreditaram que conseguiríamos atingir nossos sonhos.

Agradecemos a todos os professores que soube transmitir seus valores em conhecimentos, fazendo que se tornasse possível o entendimento das matérias aplicadas. Formando assim pessoas habitas para o mercado de trabalho e para o convívio com a sociedade.

EPÍGRAFE

Talento é 1% inspiração e 99% transpiração. (Thomas Edson)

Não me desencorajo, porque cada tentativa errada descartada é outro passo à frente. (Thomas Edison)

RESUMO

VILARINO, Alan L.,PETRALHA Marcelo G., VEIGA Wilson G.SARO – Sistema Automatizado de Reconhecimento Ótico. 2012. 45 f. Trabalho de Diplomação – Curso de Tecnologia em Automação de Processos Industriais, Universidade Tecnológica Federal do Paraná. Pato Branco, 2012.

Este trabalho apresenta um estudo sobre biometria e ferramentas que possam fazer o reconhecimento de uma imagem e compara-la com outra armazenada em um banco de dados, através de técnicas de visão computacional e processamento de imagens digitais. Foi desenvolvido um sistema de captura de imagens gratuito, utilizando linguagem C++, para a captura das imagens dos olhos de uma pessoa, onde a íris vai ser utilizada como senha de acesso. Caso os dados da íris sejam compatíveis com algum armazenado no banco de dados, o sistema permitirá o acesso desta pessoa, um software desenvolvido enviará um pulso elétrico através de um periférico do computador (USB, serial, paralela), para o acionamento de uma fechadura eletromagnética permitindo a abertura ou não de um ambiente reservado.

Palavras-chave: Reconhecimento Ótico, Biometria, Controle de acesso, Íris, Sistemas de Segurança, comunicação serial, Linguagem C++, OpenCV.

ABSTRACT

VILARINO, Alan L., PETRALHA Marcelo G., VEIGA Wilson G. ASOA – Automated System of Optical Acknowledge. 2012. 45 f. Trabalho de Diplomação – Curso de Tecnologia em Automação de Processos Industriais, Universidade Tecnológica Federal do Paraná. Pato Branco, 2012.

This work presents a study on biometrics and tools that can make the recognition of an image and compare with other stored in a database, using techniques of computer vision and image processing. A free system was developed to capture images, using C++ language, to capture images of a person's eyes, where the iris will be used as a password. If the iris data is compatible with the stored in a database, the system will allow this person access. A software developed will send an electrical pulse through a peripheral computer (USB, serial, parallel), for the actuation of a lock electromagnetic allowing the opening or not a reserved environment.

Keywords: Optical Recognition, Biometrics, Access Control, Iris, Security Systems, serial communication, language C++, OpenCV.

|

LISTA DE FIGURAS

Figura 1 – Fluxograma da execução do software que será desenvolvido	13
Figura 2 – Elementos de um sistema de processamento de imagens	16
Figura 3 – Relação entre a linguagem "C" e a linguagem "C++"	19
Figura 4 – Olho humano em corte	22
Figura 5 – Diferenças entre nove íris	23
Figura 6 – Código fonte: Captura da Imagem	24
Figura 7 – Centro da Pupila e da Íris	24
Figura 8 – Níveis de abstração e técnicas de segmentação de imagens	26
Figura 9 – Parâmetros da circunferência	27
Figura 10 – Princípio da Transformada de Hough para circunferências.	27
Figura 11 – Algoritmo da Transformada de Hough para círculos	28
Figura 12 - Código Fonte: Conceito de HOUGH	29
Figura 13 – Fecho eletromagnético THEVEAR	33
Figura 14– <i>IrisAccess</i> da LG	34
Figura 15 – BM-ET200 da Panasonic	35
Figura 16 – Diagrama de blocos do sistema SARO	36
Figura 17 – Webcams utilizadas no sistema SARO	37
Figura 18 - Imagem da íris capturada pela câmera.....	37
Figura 19 – Computador Utilizado no SARO.....	38
Figura 20 - Circuito SARO.....	39
Figura 21 – Cadastro do Sistema SARO.....	40
Figura 22 – Acesso Permitido	41
Figura 23 – Acesso negado.....	41

LISTA DE QUADROS

Quadro 1: Pinagem porta Serial, conexão DB – 9.....	30
Quadro 2: Pinagem porta Paralela, conexão 25 pinos.....	32

LISTA DE ABREVIATURAS

ASCII	American Standard Code for Information Interchange
EIA	Electronics Industry Association
GND	Terra(do inglês <i>Ground</i>)
GPU	Unidade de Processamento Gráfico
IBM	International Business Machines
OPENCV	Open Source Computer Vision
OPENGL	Open Graphics Library
RX	Receptor
SARO	Sistema Automatizado de Reconhecimento Ótico
SQL	Linguagem de Consulta Estruturada
SVC	Sistema de Visão Computacional
TX	Transmissor
VCC	Tensão de Corrente Continua

|

SUMÁRIO

1.	INTRODUÇÃO.....	11
1.1	APRESENTAÇÃO DO TRABALHO	12
1.1.1	OBJETIVO GERAL	13
1.1.2	OBJETIVOS ESPECÍFICOS	13
1.2	ORGANIZAÇÃO DO TRABALHO	14
2.	FUNDAMENTAÇÃO TEORICA	15
2.1	AQUISIÇÃO E PROCESSAMENTO DE IMAGENS.....	15
2.1.1	VISÃO COMPUTACIONAL	16
2.1.2	LINGUAGEM DE PROGRAMAÇÃO C++	18
2.1.3	OPENCV (OPEN SOURCE COMPUTER VISION).....	20
2.1.4	BIOMETRIA E RECONHECIMENTO DE ÍRIS.....	21
2.2	COMUNICAÇÃO SERIAL E PARALELA	29
2.2.1	FECHO ELETROMAGNÉTICO.....	33
2.2.2	ALGUNS PRODUTOS JÁ EXISTENTES.....	34
3	O SISTEMA SARO	36
3.1	CÂMERA.....	36
3.1.1	MICROCOMPUTADOR	37
3.1.2	ACIONAMENTO DO FECHO.....	38
4	RESULTADOS E DISCUSSÃO.....	40
4.1	RESULTADOS OBTIDOS.....	40
4.2	SISTEMA OPERACIONAL WINDOWS OU LINUX.....	42
4.3	COMPILADORES X BIBLIOTECA.....	43
5	CONCLUSÕES.....	44
5.1	SUGESTÕES E FUTURAS MELHORIAS.....	44
	REFERÊNCIAS.....	46
	APÊNDICE A – CAPTURA DE IMAGEM	49
	APÊNDICE B – COMPARAÇÃO DE IMAGEM	50

1. INTRODUÇÃO

A área de segurança, em geral, está em ampla expansão, propiciando uma maior utilização da automação. Atualmente, os sensores de presença, acesso por cartão magnético já são de uso frequente no controle de acesso. Outra forma que esta sendo utilizada para o controle de acesso é através de sistemas biométricos. Dentre as formas biométricas de reconhecimento, utilizar-se-á a íris do olho. Este método de reconhecimento, assim como impressões digitais, é para reconhecimento único de pessoas.

De acordo com Duarte e Valadão (2008)

A probabilidade de uma íris ser idêntica a outra é aproximadamente 1 em 10^{72} . Essa probabilidade, praticamente, garante que não haverá nenhuma íris idêntica à outra no mundo. A utilização de lentes de contato e óculos, também não afetam o reconhecimento da íris. O único caso que pode afetar o reconhecimento é um dano sério na íris, que modificará sua forma. O reconhecimento da íris é eficaz até mesmo para gêmeos idênticos e cegos. Portanto, como não segue padrões genéticos, até gêmeos idênticos podem possuir suas íris totalmente diferentes. O reconhecimento de íris foi considerado, por meio de testes, o método biométrico mais eficaz para identificar um indivíduo. Um teste feito com 2 milhões de pessoas obteve 100% de acertos.

Esse trabalho de diplomação tem o objetivo de desenvolver e implementar um sistema, baseado em computador, que efetue o controle automático de acesso por meio do reconhecimento da íris. Tratamento de imagem do olho, a fim de utilizar apenas a íris para o restante do processo, criação de um banco de dados para o acesso das características das íris das pessoas autorizadas, baixo custo de montagem, sistema com interface interativa de fácil uso, controle de acesso confiável.

O sistema a ser desenvolvido tem como finalidade controlar acesso de pessoas a determinadas áreas, suprir a necessidade de controle em tempo real, atender empresas do setor privado e público, com controle de acesso as suas áreas restritas. O trabalho foi proposto também com o intuito de ser desenvolvido de forma modular, permitindo sua utilização em futuros trabalhos também envolvendo identificação biométrica, por exemplo, impressão digital, voz entre outros.

1.1 APRESENTAÇÃO DO TRABALHO

Para a realização do sistema SARO foi utilizado o sistema de captura de imagens *free OpenCV (Open Source Computer Vision)* biblioteca de visão computacional de código aberto desenvolvido pela Intel, utilizando linguagem C++ (OpenCVWiki, 2008), para a captura das imagens dos olhos do usuário, onde o código obtido através de uma imagem da íris vai ser utilizada como senha de acesso.

Com a foto do olho de uma pessoa, é localizada a íris, através da utilização de algumas rotinas do *OpenCV*, a imagem da íris é então convertida em dados ou informações. Essas informações serão armazenadas em um banco de dados contendo as características das íris das pessoas cadastradas, o qual foi desenvolvido para utilização em banco de dados *MySQL* que é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL (Linguagem de Consulta Estruturada) .

No momento de execução do sistema, será novamente fotografada a íris do olho do usuário, e esta será comparada com as características das íris armazenadas no banco de dados. Caso a íris seja compatível com alguma armazenada no banco de dados, o sistema permitirá o acesso desta pessoa, o software enviará um pulso elétrico através da porta paralela para o acionamento de uma fechadura eletromagnética permitindo a abertura ou não de uma porta. Caso o acesso não seja aceito, o *software* gerará no monitor a informação de acesso inválida e a fechadura não é acionada, conforme a figura 1.

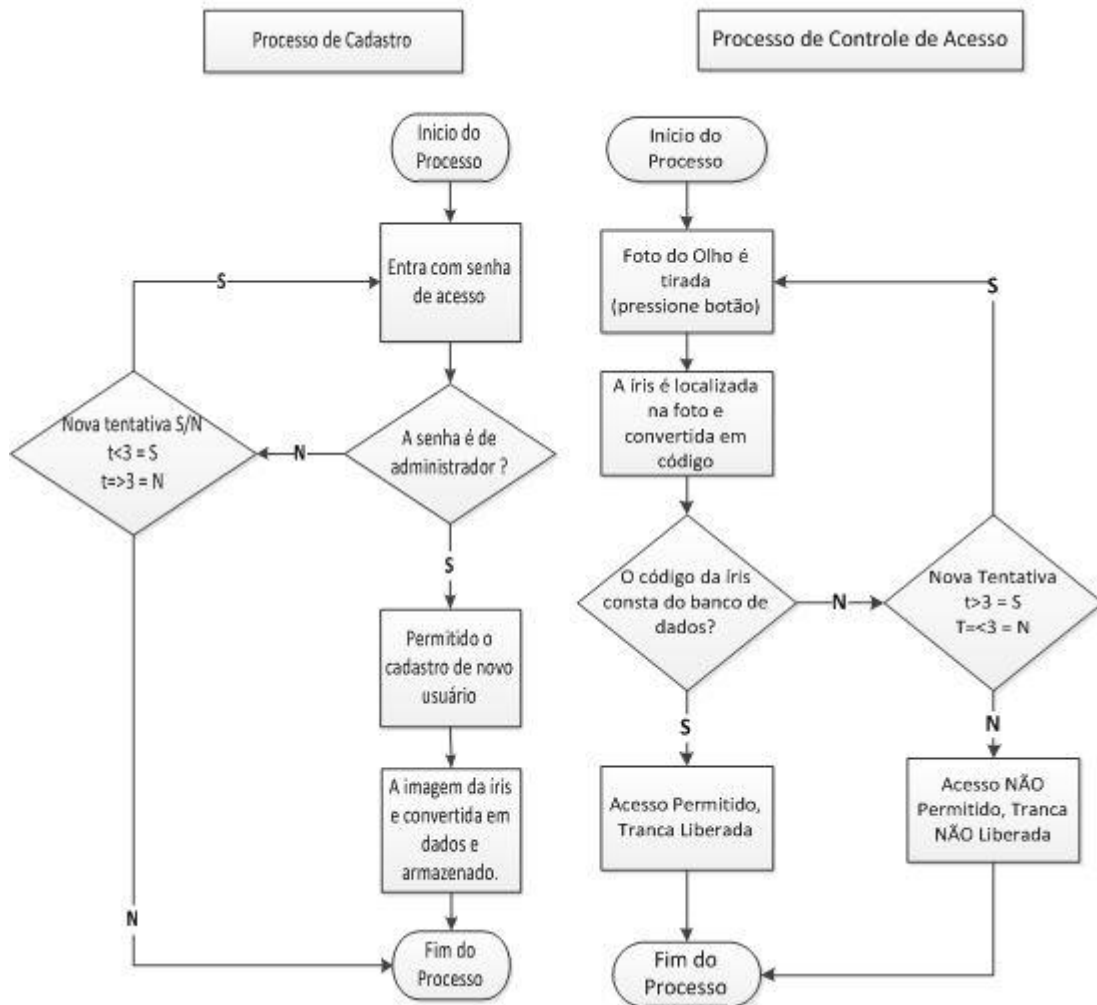


Figura 1 – Fluxograma da execução do software que será desenvolvido
 Fonte: Autoria própria

1.1.1 OBJETIVO GERAL

Desenvolver e implementar um sistema, baseado em microcomputador, que efetue o controle automático de acesso por meio do reconhecimento da íris.

1.1.2 OBJETIVOS ESPECÍFICOS

Tratamento de imagem do olho, a fim de utilizar apenas a íris para o restante do processo de reconhecimento; Criação de um banco de dados para o

acesso das características das íris das pessoas autorizadas; Baixo custo de montagem.

1.2 ORGANIZAÇÃO DO TRABALHO

Esta monografia está organizada em cinco capítulos:

No segundo capítulo – é realizado um estudo de referências sobre o assunto, define-se conceitos básicos de aquisição e processamento de imagens, a biblioteca *OpenCV*, assim como visão computacional, programação em C++, reconhecimento de íris do olho, comunicação através da porta serial do computador, e fechaduras eletromagnéticas e produtos já existentes.

No terceiro capítulo – é apresentado o sistema SARO em desenvolvimento, onde são descritas as câmeras *webcams* utilizadas, o microcomputador, e o acionamento do fecho e os materiais e recursos utilizados, para que o sistema fosse implementado.

No quarto capítulo – São apresentados os problemas e discussões assim como resultados obtidos, sistema operacional utilizado e compiladores e bibliotecas.

No quinto capítulo – São apresentadas as conclusões e sugestões para futuras melhorias.

2. FUNDAMENTAÇÃO TEORICA

Neste capítulo apresenta-se uma breve pesquisa sobre os assuntos abordados durante o desenvolvimento do sistema SARO. Foi estudada, inicialmente, a fundamentação teórica sobre aquisição e processamento de imagens, os métodos de captura através de webcams, câmeras infravermelho, câmeras com adaptações simples a fim de focar apenas o olho de uma pessoa, além do processamento e tratamento dessa imagem para convertê-la em dados, que o microcomputador possa utilizar para o restante do processo.

Juntamente com o processo de captura e tratamento da imagem foi visto sobre Visão computacional e bibliotecas de reconhecimento gráfico como o *OpenCV*, além de um breve conceito sobre biometria e os métodos mais utilizados em sistemas similares ao do SARO.

Conclui-se o estudo com um estudo sobre as portas de comunicação de um microcomputador e seus periféricos, para que fosse possível fazer a conexão mais segura do *software* do SARO com o *hardware* utilizado, que por sua vez, utilizou um fecho eletromagnético montado em um circuito desenvolvido para o desenvolvimento do sistema, sem prejudicar a saída do computador utilizada.

2.1 AQUISIÇÃO E PROCESSAMENTO DE IMAGENS

Neste item é visto então um estudo sobre a aquisição e processamento de imagens vêm crescendo muito, sendo utilizada por órgãos de trânsito, polícia, segurança em geral de estabelecimentos e residências. Na área da medicina, auxiliando profissionais em cirurgias e outros procedimentos, e também para monitoramento de nosso planeta, relevo, vegetação e meteorologia (GAZZONI; MARCON; FRANCISCON, 2006). Assim como visão artificial em máquinas e robôs, em empresas de diversas áreas visando melhorias na produção.

O processo de captura de uma imagem do olho de uma pessoa pode ser obtido através de câmeras, ou webcams, como no caso do sistema SARO. O

conjunto de imagem pode ter suas informações armazenadas de duas maneiras (imagem bidimensional ou imagem tridimensional).

No caso da imagem digital bidimensional é usada uma estrutura em forma de matriz, onde é armazenada toda informação da imagem. A cada valor destas posições na matriz dá se o nome de pixel, estes podem ser valores referentes à intensidade de luz ou ao valor das cores contidas na imagem. (GRANCHO, 2011)

Com relação ao processamento de imagens pode-se dizer que é uma área que vem crescendo muito nos últimos anos. Permitindo viabilizar grandes aplicações em duas categorias, o aprimoramento de informações expressivas para interpretação humana, e a análise automática por computador (visão computacional) de informações de uma imagem (FILHO; NETO, 1999).

Um sistema de processamento de imagens em geral é composto por alguns elementos. O diagrama da figura 2 permite representar estes elementos, ele mostra as principais formas que se pode realizar em uma imagem, como aquisição, armazenamento, processamento e exibição.

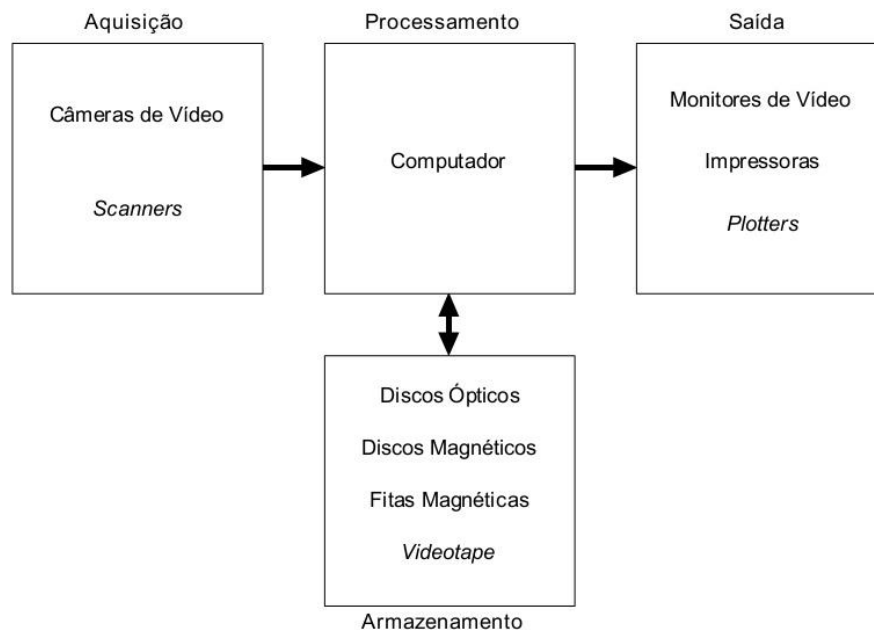


Figura 2 – Elementos de um sistema de processamento de imagens
 Fonte: (FILHO; NETO, 1999)

2.1.1 VISÃO COMPUTACIONAL

Visão computacional é "o conjunto de métodos e técnicas para que os sistemas computacionais possam ser capazes de interpretar imagens"

(WANGENHEIM; COMUNELLO, 2005), ou seja, seria como fazer um computador "enxergar", ou olhar para uma figura ou foto e transformá-la em dados úteis para usos futuros.

Os atuais métodos de processamento de imagens e visão computacional vêm sendo utilizados em muitas aplicações e trabalhos, podem ser citados trabalhos nos campos da medicina, biometria, sensoriamento remoto, entretenimento, robótica, sistemas produtivos, entre outras.

Com o avanço das tecnologias de informação (*hardware* e *software*) desenvolveram-se computadores com maior capacidade de processamento gráfico e ferramentas cada vez mais poderosas para trabalhar com visão computacional (REHEM; TRINDADE, 2009).

Além disso, a necessidade da utilização de métodos baseados em descrição textual tem introduzido o desenvolvimento de mecanismos para consulta de bases de dados através de imagens-exemplo ou pela especificação da imagem desejada, em termos de cores dominantes, textura e formato dos objetos. (RUDEK; COELHO; JUNIOR, 2001)

Assim, os sistemas de visão computacional, tornam-se muito importantes em várias etapas do ciclo de produção de um produto, como por exemplo, a direção e deslocamento de um robô ou máquina, automação de trabalhos em linhas de produção e controle de qualidade.

Para trabalhar com visão computacional faz-se o uso de bibliotecas virtuais. No desenvolvimento das bibliotecas virtuais existe a necessidade de gerenciamento de bancos de dados, que armazenem os dados dos usuários cadastrados, assim como o sistema SARO, ou até mesmo imagens se necessário. Para que o gerenciamento tenha sucesso é importante o desenvolvimento de técnicas que englobem metodologias de acesso para o arquivamento e leitura da informação.

O desenvolvimento de sistemas de visão computacional requer uma imagem, que servirá como a entrada do sistema e será processada, binarizada e transformada dados (processamento). Este processo de transformação da imagem é realizado por bibliotecas de processamento gráfico. Entre as mais conhecidas: *OpenCV*, *OpenGL* e *tec*.

A organização de um Sistema de Visão Computacional (SVC) é definida de acordo com o projeto, qual sua função, se já está predefinido ou se exige algum

processo de aprendizagem. Entretanto há funcionalidades que são comuns em vários projetos, de acordo com REHEM; TRINDADE (2009) entre elas destacam-se: aquisição de imagens, pré-processamento, extração de características, detecção e segmentação, e processamento de alto nível.

2.1.2 LINGUAGEM DE PROGRAMAÇÃO C++

Antigamente os softwares de computadores eram programados por chaves no painel. Com o passar dos anos, os programas foram crescendo e ficando mais complexos, foi aí então que surgiram os montadores assembler onde o programador podia se valer de mnemônicos para facilitar o trabalho de programação. Como os programas continuaram crescendo, surgiram as linguagens de alto nível que ofereceram melhores ferramentas para o programador enfrentar a crescente complexidade dos programas. Ou seja, que oferecem ferramentas para facilitar trabalho do programador.

De acordo com FRIGERI; COPSTEIN; PEREIRA (1996) o Fortran certamente foi à primeira linguagem de alto nível de grande utilização. Nos anos sessenta surgiu a programação estruturada, base de linguagens como o "C" e o Pascal.

Através da programação estruturada foi possível, criar diversos programas com complexidade moderada. Porém, o programa ficava muito confuso, dificultando o entendimento do mesmo (FRIGERI; COPSTEIN; PEREIRA, 1996). Os recursos da programação estruturada eram escassos para o desenvolvimento da programação, surgiu então a programação orientada a objetos, com novas funções e métodos para facilitar a programação.

Com a programação orientada a objetos preservaram-se os recursos oferecidos pela programação estruturada e foram acrescentados novos recursos que permitiram uma nova estrutura de programação.

Uma modificação da programação orientada a objetos em relação à estruturada diz que o projeto do sistema deve ser feito a partir de associações das variáveis com o mundo real e não apenas a partir de dados ou procedimentos (FRIGERI; COPSTEIN; PEREIRA, 1996), uma vez que os dados e procedimentos

mudam com muita frequência, e variáveis do mundo real visam a ser mais constantes.

Na modelagem de objetos do mundo real, existem blocos ou conjuntos de objetos com características comuns, estes blocos ou grupos são definidos como "classe". Na linguagem orientada a objetos, cada classe corresponderá a um tipo abstrato de dados, ou seja, um grupo de atributos que mostram a classe e os comportamentos sobre esses atributos, o conjunto de atributos e procedimentos são as propriedades da classe. Os objetos do mundo real são então demonstrados por instâncias dessas classes. Outros atributos importantes das linguagens de programação orientadas a objetos são subclasses e polimorfismo.

Para resolver os problemas de manutenção dos grandes sistemas, em 1980, Bjarne Stroustrup¹ criou o conceito de classes e de verificação de parâmetros de funções, e mais algumas outras facilidades para a linguagem "C". Esta nova linguagem foi chamada "C com classes". Três a quatro anos mais tarde a nova linguagem passou a ser conhecida como "C++". Após mais alguns refinamentos serem adicionados na linguagem C++, tornou-se disponível ao público (SCHILD, 1992).

A linguagem C++ pode ser considerada como uma extensão da linguagem C (FRIGERI; COPSTEIN; PEREIRA, 1996). A figura 3 mostra o relacionamento entre as duas linguagens.

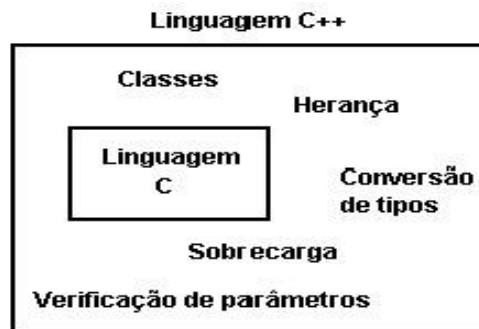


Figura 3 – Relação entre a linguagem "C" e a linguagem "C++"
Fonte: (FRIGERI; COPSTEIN; PEREIRA, 1996)

A linguagem C++ adiciona na linguagem C facilidades que deixam o desenvolvimento de uma programação orientada a objetos sem alterar as características da linguagem C.

¹Bjarne Stroustrup (Århus, 11 de junho de 1950) é um cientista da computação dinamarquês e professor catedrático da Universidade do Texas A&M. É conhecido como o pai da linguagem de programação C++.

O C++ permite operações que normalmente só são assumidas em linguagens de baixo nível e ao mesmo tempo em que oferece facilidades em programação orientada a objetos. Isso quer dizer que a linguagem oferece as condições, mas não deixa de exigir programadores bons e disciplinados. Assim como qualquer linguagem de programação, todos os recursos oferecidos e utilizados são responsabilidade por parte do programador. Como C++ é uma extensão da linguagem C é possível que um programa em C seja compilado sem problemas em um compilador C++, pois este compilador deve é capaz de compilar os comandos da linguagem C e os comandos adicionados que constituem o C++, uma das linguagens mais conhecida do mercado, e a opção de escolha da maioria dos programadores e desenvolvedores de sistemas.

É uma linguagem de baixo nível, e pode usar a orientação a objetos para aumentar o nível de programação. Além de que existe uma quantidade gigantesca de bibliotecas e componentes aproveitáveis para aplicações em C++ (VILLAS-BOAS, 2006).

2.1.3 OPENCV (OPEN SOURCE COMPUTER VISION)

O *OpenCV* é uma biblioteca *free* (livre) de uso acadêmico e comercial, criada pela Intel e utilizada para a criação de softwares nas áreas de processamento de imagens e visão computacional.

Foi desenvolvida em linguagem C e C++, podendo ser executada em diversos sistemas operacionais como o Linux, Windows, Mac OS, Unix, Android e até GPUs Nvidia (CUDA) (OpenCVWiki, 2008).

Ela traz consigo vários algoritmos para a aquisição e ao tratamento de imagens, o que facilita o desenvolvimento de trabalhos com visão computacional, tais como: segmentação, reconhecimento de faces, aprendizado de máquinas, filtragem de imagens e vídeo, reconhecimento de padrões, entre outros. Por se tratar um código aberto, vários algoritmos já implementados podem ser utilizados, adaptados, ou reescritos.

Existem ainda outras bibliotecas de processamento gráfico semelhantes ao OpenCV, como o Emgu CV², porém foi desconsiderado então o uso do Emgu CV no projeto SARO, já que este foi desenvolvido para funcionar via computador e não através de interfaces gráficas, fica aí uma opção para futuros projetos acadêmicos.

Outra biblioteca é o OpenGL (*Open Graphics Library*), uma biblioteca livre de computação gráfica, para desenvolvimento de aplicativos gráficos, ambientes 3D, jogos, entre outros. É um conjunto de funções, que fornecem acesso a praticamente todos os recursos do hardware de vídeo (GROUP, 1997). Também foi projetada para funcionar mesmo que uma máquina que esteja exibindo gráficos não seja a mesma que contém o programa gráfico. Trabalha com medida de ângulos em graus e não em radianos, como ocorrem com a maioria das linguagens, suas coordenadas gráficas segue o eixo cartesiano, assim o número da linha é decrescente, e não crescente, como em boa parte das implementações. E também as matrizes no OpenGL são calculadas com base nas colunas, e não nas linhas.

Com as mesmas características da OpenGL existem outras bibliotecas, bastante conhecida também é a DirectX, desenvolvida para ambiente Windows, mas ambas são mais dedicadas para jogos e softwares que exigem um excelente processamento gráfico.

2.1.4 BIOMETRIA E RECONHECIMENTO DE ÍRIS

A Biometria é uma tecnologia para identificação de pessoas, é utilizada para diferenciar um ser humano de outro se baseando nas diferenças das suas características físicas e biológicas (VALADÃO, 2008).

Em qualquer sistema que exija uma senha, a pessoa cadastrada, ou autorizada, é a única que possui as suas informações de acesso em relação a outras pessoas ou usuários. Já na Biometria essa singularidade ocorre por meio de características físicas dessa pessoa que são únicas e individuais para cada pessoa. As técnicas mais conhecidas de biometria são: impressão digital, reconhecimento de face, de voz, de assinaturas, de íris, de retina.

² EmguCV é uma biblioteca de processamento gráfico desenvolvido pela Intel, porém é desenvolvido em linguagem C#, para desenvolvimento de aplicações envolvendo interfaces gráficas por simplificar a tarefa do programador (EmguCVWiki, 2012).

O sistema biométrico SARO utiliza a íris como acesso. A íris é um músculo do olho, a parte visível do olho humano responsável pela coloração do olho, situada em torno da pupila (Figura 4), é uma superfície quase plana, com uma abertura em formato de círculo no centro, é composta por de fibras musculares lisas (VALADÃO, 2008).

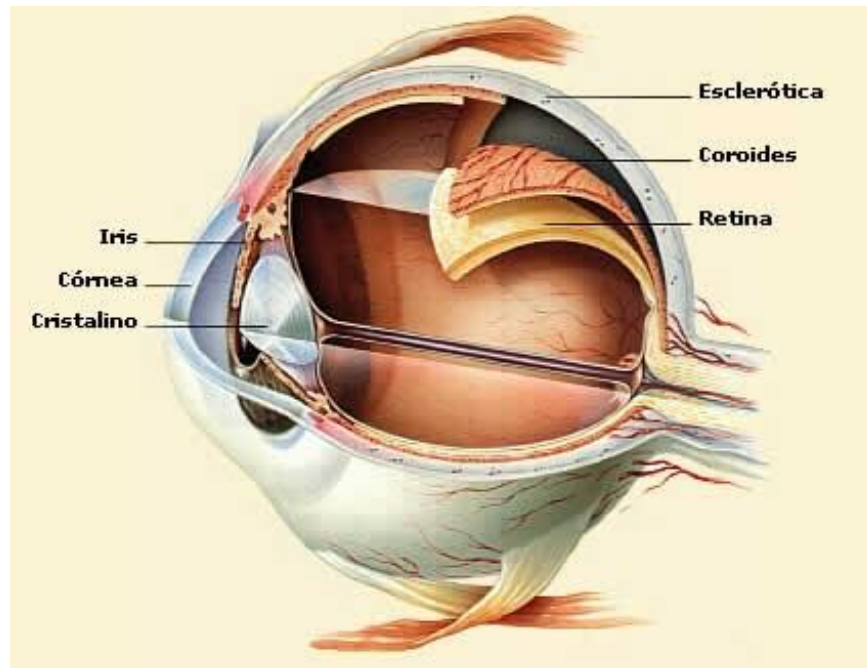


Figura 4 – Olho humano em corte
Fonte: (VALADÃO, 2008)

Ao ser feita uma análise das camadas do olho de fora para dentro, a íris seria a terceira “camada”, atrás apenas da córnea e do humor aquoso (JÚNIOR, 2005). Sua função é regular a quantidade de luz que entra nos olhos, através da contração e dilatação da pupila.

A íris possui características que são próprias de cada pessoa. Durante o envelhecimento de uma pessoa, a íris não se altera biometricamente, sendo uma das suas características fisiológicas mais expressivas (CHAVEZ; IANO; SABLON, 2006).

Cada pessoa tem uma íris diferente das outras pessoas, mesmo que sejam gêmeos univitelinos, e também a íris direita e a esquerda de uma mesma pessoa são diferentes uma da outra. Porém existem algumas maneiras da íris se modificar durante a vida de uma pessoa, é o caso de um trauma (batida de carro, por exemplo) ou uma cirurgia de glaucoma. Já cirurgias de miopia ou astigmatismo e

o uso de óculos ou lentes não alteram o relevo da íris (TISSE; MARTIN; ROBERT, 2003). O reconhecimento de íris não a prejudica por se tratar de uma técnica sem contato com o olho, dependendo apenas da captura da imagem e do posterior processo de reconhecimento (CHAVEZ; IANO; SABLON, 2006). Na figura 5 pode-se ver a diferença entre nove íris diferentes.

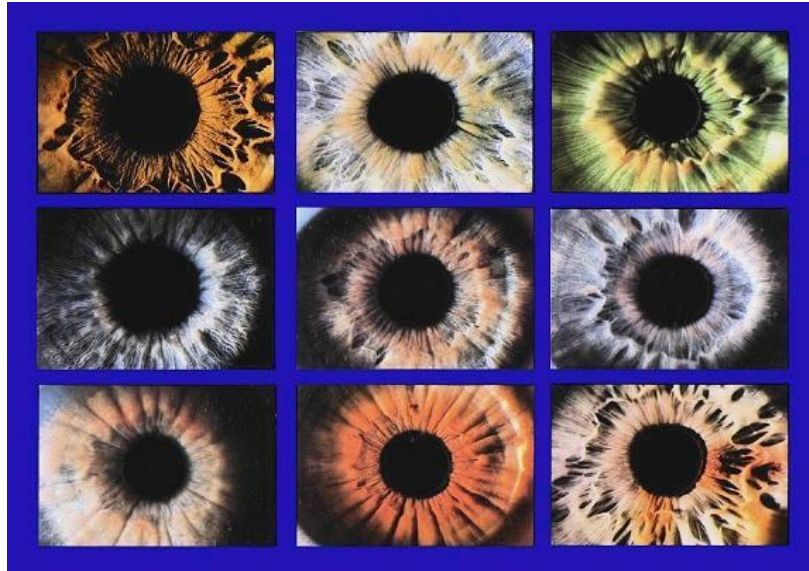


Figura 5 – Diferenças entre nove íris
Fonte: (VALADÃO, 2008)

Os algoritmos de reconhecimento da íris, geralmente, são decompostos em blocos ou subseções (CHAVEZ; IANO; SABLON, 2006), para facilitar a análise da íris. Um modelo de algoritmo bastante utilizado segue os seguintes passos: localização de íris, correções matemáticas para robustez do modelo, como a distância de Hamming e a Transformada de Hough, a codificação da íris e comparação dos códigos; outro modelo semelhante, porém mais eficiente segue com: localização de íris, normalização, realce, extração de características e comparação.

Após adquirir a imagem da íris, o próximo passo é processar os dados da imagem adquirida e efetivamente “encontrar” uma íris, ou seja, separar a imagem efetiva da íris das outras imagens, como o resto do olho (VALADÃO, 2008). Para retirar as partes que não interessam para a imagem são aplicados filtros visando deixar apenas a imagem da íris. A figura 6 abaixo mostra uma parte do código da captura da imagem.

```

//CvCapture* capture = 0;
capture=cvCaptureFromCAM( CV_CAP_ANY );
printf("Identificação de Pessoas\n");
//Cria janela
//cvNamedWindow( "Imagem", CV_WINDOW_AUTOSIZE );

while(1){
// comando de captura
IplImage* frame = cvQueryFrame(capture);
if( !frame ) {
    fprintf( stderr, "ERROR: frame is null...\n" );
    getchar();
    break;
}

// Salvando arquivo
char file[255];
printf("Informe o nome do usuario para acesso:\n");
scanf("%s",file);
strcat(file,".bmp");
cvSaveImage(file,frame);
(((cvWaitKey(10) & 255) == 27) || fechado == 1 );
fechado = 0;
break;
}

```

Figura 6 – Código fonte: Captura da Imagem
Fonte: Autoria própria

A avaliação do centro da íris é feita através de uma estimativa do raio da íris e da pupila com as coordenadas do centro da imagem (Figura 7). Pode-se pensar então que o centro da pupila é o centro da íris, mas isso não deve ser considerado, pois o centro da pupila é inferior ao centro da íris, podendo variar (VALADÃO, 2008).

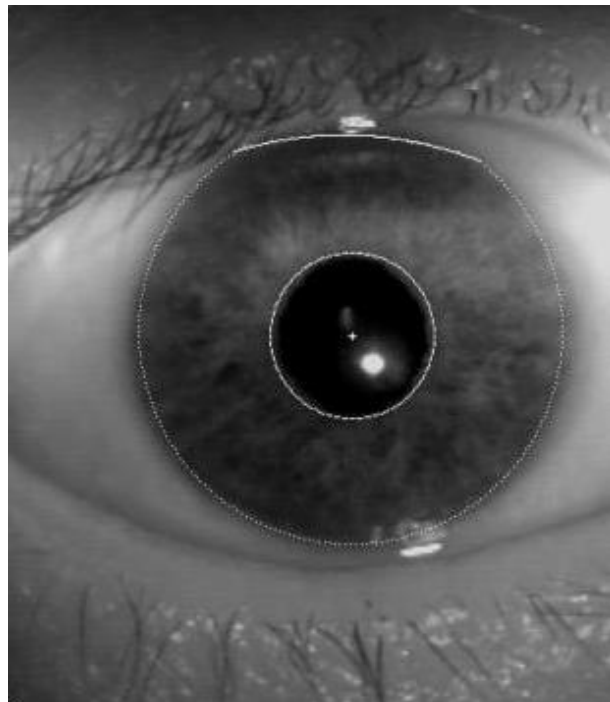


Figura 7 – Centro da Pupila e da Íris
Fonte: (VALADÃO, 2008)

Segundo o Daugman (2004), renomado no estudo de Reconhecimento de Íris, criou diversos algoritmos e fórmulas para o reconhecimento de íris. Por exemplo, para adquirir as fronteiras da íris e separá-la da pupila, Daugman (2004) propôs a seguinte integral parametrizada (Equação 1):

$$\max_{(r, x_0, y_0)} \left| G_{\sigma}(r) * \frac{\partial}{\partial r} \oint_{r, x_0, y_0} \frac{I(x, y)}{2\pi r} ds \right| \quad (1)$$

Onde $I(x, y)$ é uma imagem contendo um olho e $2\pi r$ é a medida do contorno da imagem adquirida. Os parâmetros são: r , que é o raio da imagem (não apenas da íris) e x_0 e y_0 que são as coordenadas do centro da imagem. Essa fórmula permite calcular tanto os limites da íris, quanto os limites da pupila, podendo separá-las e utilizar apenas a imagem da íris nos cálculos posteriores. A função G_{σ} serve para suavizar ruídos, isto é, diminuir qualquer fator na imagem que prejudique o reconhecimento da íris, por exemplo, reflexos (DAUGMAN, 2004).

Para a comparação das imagens é feito um teste de compatibilidade entre dois códigos de íris diferentes. Esse teste é montado com uma lógica XOR (XOU, OU exclusivo).

O XOR é aplicado os bits do código de íris, garantindo que não haverá diferença entre dois bits correspondentes, enquanto que um AND (E) é aplicado nos bits de máscara correspondentes garantindo que não haverá problemas com cílios, pálpebras, reflexos e outros fatores externos (DAUGMAN, 2004).

Os resultados obtidos são utilizados para o cálculo da distância de Hamming, utilizada para achar o número de erros ou dados diferentes entre dois arquivos ou processo, no caso do processo do sistema SARO define a porcentagem de diferença entre o código de duas íris. Na equação 2 segue a fórmula para o cálculo da distância de Hamming, a onde *Mask* e *Code* são os códigos de máscara de um olho e o código de íris para esse mesmo olho A ou B.

$$HD = \frac{\| (codeA \otimes codeB) \cap maskA \cap maskB \|}{\| maskA \cap maskB \|} \quad (2)$$

Em suma, quanto maior é a distância de Hamming, mais chance de duas íris diferentes serem iguais e quanto menor a distância de Hamming, mais chance

de se comparar errado a mesma íris, contudo as probabilidades são altamente a favor do reconhecimento de íris.

Ainda há muito a ser desenvolvido em todas as áreas da biometria para que ela seja aplicada em larga escala no mundo. Apesar disso, ela já pode ser considerada uma evolução drástica nos meios de identificação e promete substituir senhas, logins e cartões de identificação, além de ser um ótimo meio para controle e organização de pessoas, seja em áreas militares, aeroportos ou qualquer outra área que necessite. (VALADÃO, 2008)

Uma das áreas de pesquisa mais destacadas em Processamento de imagens é a segmentação, a qual busca separar as regiões da imagem de interesse para um determinado processo.

As técnicas de segmentação podem ser classificadas a partir do nível de abstração, conforme ilustrado no diagrama da Figura 8.



Figura 8 – Níveis de abstração e técnicas de segmentação de imagens.
Fonte: (FERNANDES, 2009)

A transformada de Hough (HT) é utilizada para descobrir padrões de linhas em imagens desfocadas, sem nitidez, junto com suas variações formam um conjunto de técnicas de alto nível de abstração para segmentação de imagens baseadas em modelos geométricos (linhas, círculos, elipses) que descrevem uma região de interesse que se deseja segmentar (FERNANDES, 2009). Utilizou-se neste trabalho a HT para a detecção da região de interesse circular que define a pupila.

Para a detecção de regiões circulares, são dadas as equações ($x = x_c + r * \cos \alpha$), e ($y = y_c + r * \sin \alpha$), que são as equações paramétricas de uma circunferência, onde através das coordenadas de um ponto qualquer da

circunferência é obtido um conjunto de parâmetros (x_c, y_c, r, α) (FERNANDES, 2009) como pode ser visto na figura 9.

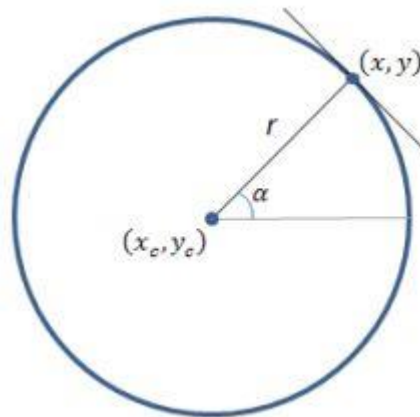


Figura 9 – Parâmetros da circunferência
Fonte: (FERNANDES, 2009)

Onde (x, y) são as coordenadas de um ponto da circunferência, (x_c, y_c) o centro da circunferência, r é o raio e α é o ângulo de direção à tangente no ponto (x, y) , em relação ao eixo de referência (FERNANDES, 2009). Uma vez que o ângulo α é equivalente ao ângulo φ de direção do gradiente no ponto (x, y) , infinitas circunferências podem passar pelo ponto (x, y) , mas somente duas em cada valor do parâmetro, e apenas uma se o interesse seja apenas ao gradiente positivo ou negativo, conforme a Figura 10.

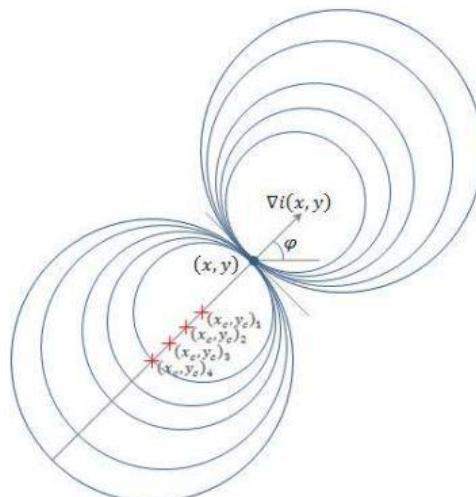


Figura 10 – Princípio da Transformada de Hough para circunferências.
Fonte: (FERNANDES, 2009)

O espaço de Hough, ou espaço de parâmetros, para circunferências define uma matriz tridimensional em que cada célula representa uma circunferência

demonstrada por (x_c, y_c, r) do qual o gradiente aponta para um centro (x_c, y_c) à uma distância r deste centro. Os resultados são armazenados nas células da matriz, ou matriz acumuladora, onde R é o raio máximo esperado para uma circunferência e W, H são a largura e a altura da imagem, como pode ser visto na figura 11 (FERNANDES, 2009).

Se existem pontos de máximos proeminentes na matriz, isto indica que existem circunferências na imagem de entrada, definidas pelos parâmetros associados às coordenadas destes pontos. A detecção de circunferências é então resumida à procura de máximos locais na matriz acumuladora (FERNANDES, 2009).

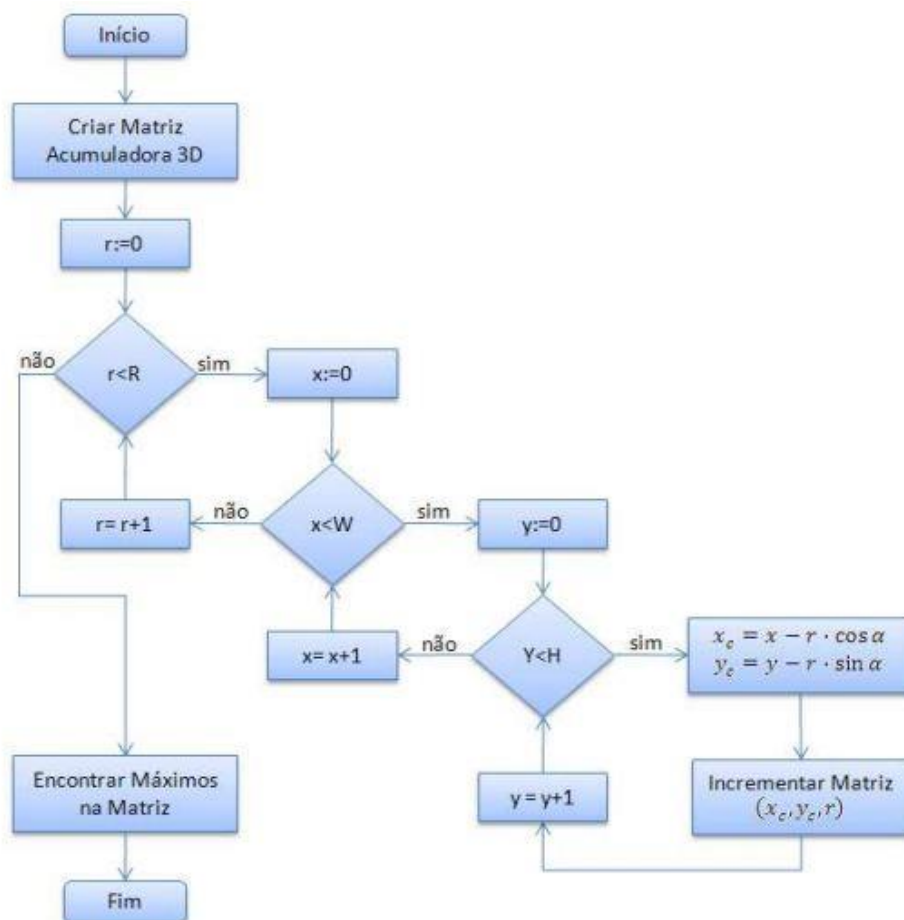


Figura 11 – Algoritmo da Transformada de Hough para círculos
Fonte: (FERNANDES, 2009)

Para realizar o conceito de Hough, é apresentado na figura12 um trecho do código fonte, onde se determinam os postos de escala de cinza e o calculo do raio e da circunferência.


```

// Tratamento de escala de cinza
while(1) {
    frame = cvQueryFrame(capture);
    if(!frame) break;
    cvSmooth(frame,frame);
    cvCvtColor(frame, gray_frame, CV_RGB2GRAY);
    CvMemStorage* storage = cvCreateMemStorage(0);
    if(gray_frame->origin != IPL_ORIGIN_TL){
        cvFlip(gray_frame, gray_frame, 0);
    }
    // Detecção do Circulo
    CvSeq* results = cvHoughCircles(gray_frame, storage, CV_HOUGH_GRADIENT, 1, gray_frame->width,100,1,10,50);
    for(int i = 0; i<results->total; i++){
        float* p = (float*) cvGetSeqElem(results, i);
        // Analise dos Ponto em escala de cinza
        CvPoint pt = cvPoint(cvRound(p[0]), cvRound(p[1]));
        cvCircle(gray_frame,pt,cvRound(p[2]), CV_RGB(0xff,0xff,0xff));
    }
}

```

Figura 12 - Código Fonte: Conceito de HOUGH
Fonte: Autoria própria

2.2 COMUNICAÇÃO SERIAL E PARALELA

A comunicação de dados é a transferência de números binários entre circuitos eletrônicos. Dependendo da comunicação, esses bits poderão codificar a informação que está trafegando de acordo com um software, e fazer com que algum periférico execute uma função (FERNANDO ALVES, 2003).

A transmissão de dados pode acontecer por portas paralelas ou por portas seriais. As portas seriais são portas de comunicações, que transmitem e recebem informações em série isto é, bit a bit. É usada entre os periféricos de um computador, tem uma função simples: uma linha envia um dado (pino Tx), e outra recebe dados (pino Rx), enquanto os outros pinos servem para estabelecer como os dados são enviados, além da alimentação Vcc e o GND.

Mesmo sendo mais lenta em relação à porta paralela, a serial é muito usada em conexão com modems, os quais podem obter sinais por uma linha telefônica e também para mouse, que não precisa enviar grandes quantidades de dados (FERNANDO ALVES, 2003).

O termo oficial definido pela IBM (*International Business Machines*) para a porta serial é porta de comunicações de dados assíncrona (FERNANDO ALVES, 2003). A maioria das ligações seriais aceitas pela indústria de computadores PC operam sob um padrão chamado RS-232 definido pela EIA (*Electronics Industry*

Association). Essa forma de comunicação é chamada serial porque os bits de informação são transferidos em longas séries (FERNANDO ALVES, 2003).

Para estabelecer comunicações seriais confiáveis é preciso contornar problemas de erro de bit e muitos outros mais. Mesmo assim, as comunicações seriais funcionam bem e, portanto são indispensáveis para a comunicação de um computador com um periférico.

O RS-232 atribui funções particulares aos pinos em uma porta serial. Além de dois pinos responsáveis por enviar (pino 3) e receber os dados (pino 2), outros são necessários para fazer com que tudo funcione corretamente, sem perder dados, bits durante uma transmissão, de acordo com o que foi solicitado.

No quadro 1 pode-se ver a pinagem, seus nomes e funções nos conectores seriais da IBM com 9 pinos chamado de conector DB-9.

PINO	FUNÇÃO	NOME
1	Detecção de portadora	CD
2	Recepção de dados	RXD
3	Transmissão de dados	TXD
4	Terminal de dados pronto	DTR
5	Terra do sinal	GND
6	Data set pront	DSR
7	Solicitação de envio	RTS
8	Pronto para enviar	CTS
9	Indicador de chamada	RI

Quadro 1: Pinagem porta Serial, conexão DB – 9.
Fonte: (FERNANDO ALVES, 2003)

Já as portas paralelas eram utilizadas, principalmente, para conexão de computadores com impressoras, com passar dos tempos mais periféricos começaram a utilizar a conexão paralela, por ser rápida e segura. Conduz os sinais por meio de oito fios separados, um para cada bit de um byte de dados. São simples, pois lidam com os dados da forma como o computador faz, em bytes ao invés de bits. Continua sendo a forma fácil e confiável de ligar uma impressora a um computador.

Para compensar a falta de um circuito de porta extra, a interface paralela aumenta o custo com um cabo mais grosso. O custo do cabo não é a única desvantagem de uma ligação paralela, (FERNANDO ALVES, 2003).

Com relação à pinagem da porta, a IBM decidiu criar seu próprio padrão de conectores de porta paralela. Enquanto a Centronics³ usa um conector de 36 condutores, a IBM escolheu um conector de 25 pinos.

Inicialmente, as portas paralelas foram criadas a princípio para servirem como saídas de dados, ou seja, apenas um sentido, do PC para a impressora. Somente alguns sinais de controle de porta paralela precisavam seguir o caminho contrário. Por isso, todos os primeiros PCs foram equipados com portas paralelas unidirecionais, ou seja, poderiam enviar dados, mas não recebê-los. Evitando que qualquer equipamento alterasse as saídas da porta, a IBM impediu o uso da porta paralela para a aquisição e recepção de dados.

Porém a IBM estudou novamente esse processo e começou a montar portas paralelas capazes de realizar operações bidirecionais, (FERNANDO ALVES, 2003).

Hoje você pode ligar uma porta paralela a uma série de dispositivos. Alguns deles são adaptadores de rede e sistemas de intercâmbio de dados.

Dos 25 pinos de um conector de porta paralela, 19 são usados para a função aparentemente trivial de transferir dados. O quadro abaixo mostra onde cada conexão recebe o conector de porta paralela no padrão IBM.

Conectores de 25 pinos	Função	Conectores de 25 pinos	Função
1	Estrobo	16	Inic. Da Impressora.
2	Bit de dados 0	17	Seleção de entrada
3	Bit de dados 1	18	Terra
4	Bit de dados 2	19	Terra
5	Bit de dados 3	20	Terra
6	Bit de dados 4	21	Terra
7	Bit de dados 5	22	Terra
8	Bit de dados 6	23	Terra
9	Bit de dados 7	24	Terra
10	Confirmação	25	Terra

³ A Centronics Data Computer Corporation foi uma fabricante pioneira de impressoras estadunidense, agora somente lembrada pela interface paralela que leva seu nome.

Conectores de 25 pinos	Função	Conectores de 25 pinos	Função
11	Ocupada		
12	Fim de papel		
13	Seleção		
14	Avanço Automático		
15	Erro		

Quadro 2: Pinagem porta Paralela, conexão 25 pinos
Fonte: (FERNANDO ALVES, 2003)

A linha de estrobo serve exatamente para isso. O sinal de estrobo do sistema paralelo da IBM é negativo. Quando os bits de dados não estão prontos, ele passa para o estado alto. Quando um byte de dados deve ser transmitido, ele passa ao nível baixo. O sincronismo entre os sinais de dados e o sinal de estrobo é muito importante.

Foi testado também o uso da porta USB (Universal Serial Bus) no projeto, porém acabou não sendo utilizada.

A porta USB surgiu em 1995, através de uma parceria entre várias companhias da área de tecnologia, a Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC e Philips, (SILVA, 2006).

A primeira versão foi a 1.0 com velocidade de transmissão de dados de 1,5Mbs (*Low-speed*), sendo substituída em seguida pela 1.1 com velocidades que iam de 1,5 Mbps a 12 Mbps. Essa versão ficou conhecida como *Full-speed*. Até que no final de 2000 foi lançada a versão 2.0 (*high-speed*), com vários aperfeiçoamentos chegando a 480 Mbps, equivalente a cerca de 60 MBps (60 milhões de Bytes por segundos).

O cabo USB é composto por 4 fios e uma malha para eliminação de ruídos simples. Dois fios transportam energia para alimentar dispositivos, o fio de cor vermelha é chamado Vbus (5v), ou seja, é o fio positivo de fornecimento de energia. O fio de cor preta é o GND (0v).

O barramento USB pode fornecer no máximo 5 Volt de tensão e 500mA de corrente elétrica, já a quantidade de corrente que seu dispositivo irá precisar para funcionar pode ser configurada via software, (SILVA, 2006).

Os outros dois fios chamados de D+ (dado+) e D- (dado-) são usados para transferência de dados entre o computador e os dispositivos periféricos.

Ao contrário da Porta Serial e Paralela onde através dos sinais elétricos dos pinos e um programa básico é possível se comunicar com um dispositivo, na USB isso só é possível se o dispositivo carregar o protocolo USB num chipset ou mesmo dentro de um microcontrolador.

2.2.1 FECHO ELETROMAGNÉTICO

O fecho eletromagnético (figura 13) utilizado para a realização do *hardware* do sistema SARO, é um modelo da THEVEAR, que diferente dos modelos convencionais, para manter o fecho destravado depois de aberto, não é necessário que o usuário fique pressionando o botão de acionamento até que alguém abrir a porta; apenas basta o operador pressionar uma única vez o botão para o fecho abrir e permanecer destravado até ser novamente travado com o retorno do portão.

Os fechos são fáceis e práticos de serem instalados desde que sejam respeitadas as especificações técnicas e exigências de cada modelo. O espelho do fecho é reversível, permitindo o uso em portas que abrem tanto à direita quanto à esquerda.

Funcionam ligados a: kits residenciais, centrais de portarias coletivas e centrais de comunicação telefônica. Proporcionando o controle de acesso em locais que exige segurança com conforto, qualidade e alta eficiência. Com o acionamento eletromagnético feito com 12V/1A (THEVEAR, 2008).

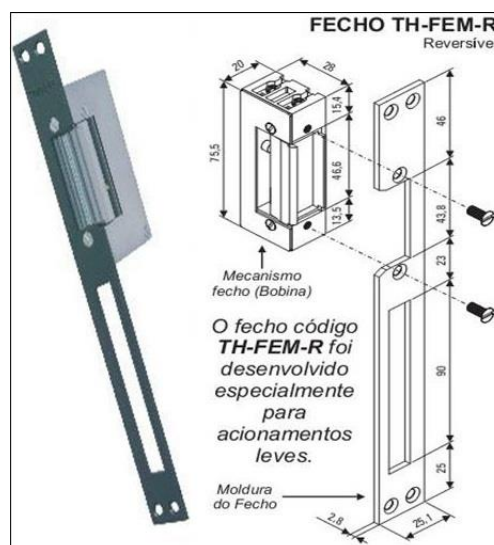


Figura 13 – Fecho eletromagnético THEVEAR
Fonte: (THEVEAR, 2008)

2.2.2 ALGUNS PRODUTOS JÁ EXISTENTES

Grandes empresas também desenvolveram seus sistemas para reconhecimento de íris, com mais recursos e *hardware* desenvolvido com câmeras próprias, direcionados a instituições financeiras ou empresas preocupadas com a segurança no acesso a áreas restritas.

Lançado pela LG *Electronics*⁴, o *IrisAccess* (ID IRIS, 2000)(figura 14) usa um sensor que faz a leitura da íris do olho humano. Já disponível no mercado brasileiro, o produto ajuda instituições financeiras, entre outras, a obterem mais segurança em transações como controle de acesso a áreas restritas, substituição de senhas e cartões magnéticos em transações bancárias, segurança na Internet, etc (MENDES, 2000).



Figura 14– IrisAccess da LG
Fonte: (MENDES, 2000)

A LG tem sido líder na comercialização da tecnologia de reconhecimento de íris nos últimos 15 anos. IrisAccess continua a liderar o mercado como plataforma mais avançada e mais amplamente utilizado do mundo o reconhecimento de íris (MENDES, 2000).

⁴A LG Electronics é a uma empresa coreana que iniciou suas atividades em 1947 na Coréia do Sul, atua em vários setores, como química e energia, maquinaria e metais, finanças, serviços, trade, indústria elétrica e eletrônica.

Para que o reconhecimento seja consumado, o usuário deve ser previamente cadastrado no sistema, por meio da gravação da íris. Esta imagem é transformada em códigos complexos de matrizes, e após este cadastramento, o usuário já está apto a utilizar o sistema.

Aproximando-se o olho do leitor, ele é focado automaticamente, mesmo que o usuário do sistema esteja de óculos, lentes de contato ou em um ambiente sem iluminação. Em no máximo dois segundos o reconhecimento é efetuado, autorizando-se ou não a abertura de uma porta ou a entrada em um sistema, por exemplo.

Já a Panasonic⁵ desenvolveu o BM-ET200 (Figura 15), que segundo a descrição da empresa oferece mais precisão e mais rapidez tecnologia biométrica de reconhecimento da íris.

O sistema pode ser usado para verificar IDs em uma variedade de aplicações, incluindo aeroportos e de entrada ou saída de controle para diversos tipos de indústrias (PANASONIC, 2012).

Com uma câmera de vídeo especial da Panasonic, um *close-up* detalhado da íris de cada pessoa autorizada é capturado no sistema. O software biométrico faz então um modelo ou "mapa" do padrão da íris de cada pessoa, e armazena-la no sistema.



Figura 15 – BM-ET200 da Panasonic
Fonte: Folheto Panasonic BM-ET200 (PANASONIC, 2012),

⁵A Panasonic é uma empresa japonesa fabricante de eletroeletrônicos. Foi fundada em 1918 com o nome de Matsushita Electric Industrial Co. Ltd. atua com uma ampla gama de produtos desde Pilhas e Baterias, Áudio e Vídeo, Telecomunicações, Broadcasting, Cuidados pessoais, Industrial e Acessório.

3 O SISTEMA SARO

Para a realização do sistema SARO, montou-se um diagrama de blocos para facilitar a compreensão do projeto, onde o usuário posicionará o olho em frente à uma webcam, acionando uma tecla através do teclado do computador que fará a execução do Software, acionando o fecho eletromagnético. Um diagrama de blocos na figura 16 demonstra como será montado o projeto SARO.

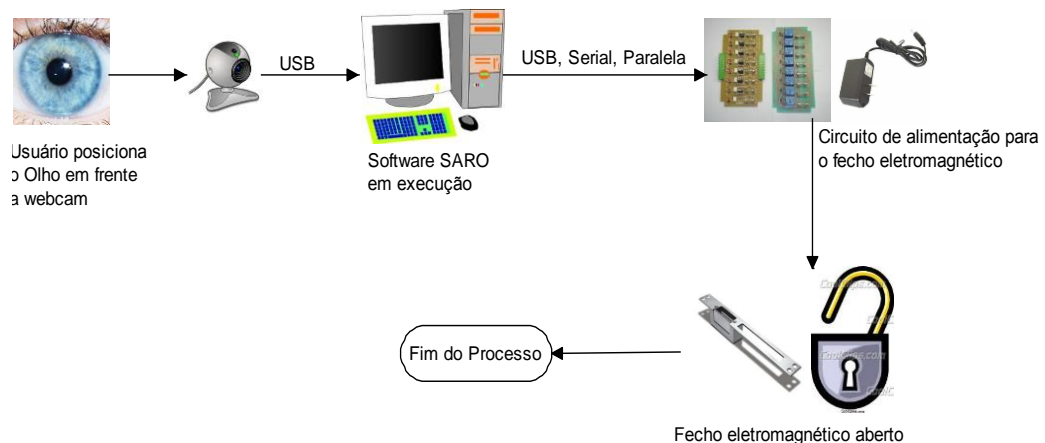


Figura 16 – Diagrama de blocos do sistema SARO
Fonte: Autoria própria

3.1 CÂMERA

Durante os testes de captura da íris foram utilizados dois modelos de câmeras conforme a figura 17, a primeira de 2mp que apresentava problemas de foco, não possuía bom foco deixando a imagem manchada.

A Segunda câmera apesar de ser de 1.3 mp seu foco era melhor, mas por conta dos leds da câmera apresentou-se reflexo nos olhos, além do problema de manchas nas imagens.

Com esses problemas o programa ficava incapaz de capturar e tratar a imagem para gerar o código. A figura 18 mostra o reflexo que apresenta nas fotos capturadas pela câmera.



Figura 17 – Webcams utilizadas no sistema SARO
Fonte: Autoria Própria



Figura 18 - Imagem da íris capturada pela câmera
Fonte: Autoria Própria

3.1.1 MICROCOMPUTADOR

A proposta apresentada solicitava a utilização de micros ditos obsoletos, onde a função seria reaproveitar esses computadores. Para desenvolver o programa foi criado em computadores com processamento dual core, sendo assim não teve problema com primeiros testes, porém ao migrar o sistema para o computador onde sistema seria implementado começaram a apresentar problemas. Primeiro deles foi a questão do processamento o computador utilizado era um

processador Pentium 4 com 1giga de memoria DDR, o programa ficava lento, e muitas vezes chegando a travar quando chegava a etapa captura da imagem.

Utilizando uma placa de vídeo dedicada de 64 mbps não suportava o nível de processamento que necessária para a imagem. Onde em alguns projetos que se conheceu durante a pesquisa utilizavam placas dedicada com quatro saídas de vídeo e memoria dedicada de 2gbts. A figura 19 abaixo mostra o microcomputador utilizado na realização do programação.



Figura 19 – Computador Utilizado no SARO
Fonte: Autoria Própria

3.1.2 ACIONAMENTO DO FECHO.

Para o acionamento do fecho através do sistema SARO, foi necessário analisar a melhor possibilidade de comunicação, uma vez que proposto um circuito simples e barato.

Primeiro foi verificado as opções de comunicação que o micro computador poderia nos oferecer, através de seus periféricos. Portas de comunicações e de reconhecimento DRIVER USB, SERIAL e PARALELA.

Para acionar um circuito através da porta de comunicação USB é necessário que você utilize um conversor serial - USB, um conversor *receiver* (MAX 232) um microcontrolador, além de outros componentes.

Já a porta serial não precisaria do conversor serial USB, mas para o funcionamento do circuito teria que utilizar os mesmos componentes que utilizaria para a porta USB, exceto o conversor serial USB.

A porta paralela tem a comunicação mais rápida e possui um sinal em nível TTL, onde possibilita o acionamento de um circuito qualquer seja ele simples ou micro controlado.

O circuito só ira funcionar assim que o *software* validar a opção de acesso, enviando um pulso de tensão através da porta paralela que acionará um opto acoplador 4N25A e um relé permitindo que a fonte externa 12v/1A alimente o fecho, liberando o acesso ao local.

Este circuito foi desenvolvido visando também à proteção da porta de comunicação utilizada, evitando retorno de corrente e tensão proveniente dos componentes utilizados na montagem do circuito. É um circuito de simples implementação e de baixo custo, com autoria própria facilitando a operação e futuras manutenções do mesmo. Abaixo segue a figura 20 com o circuito montado.

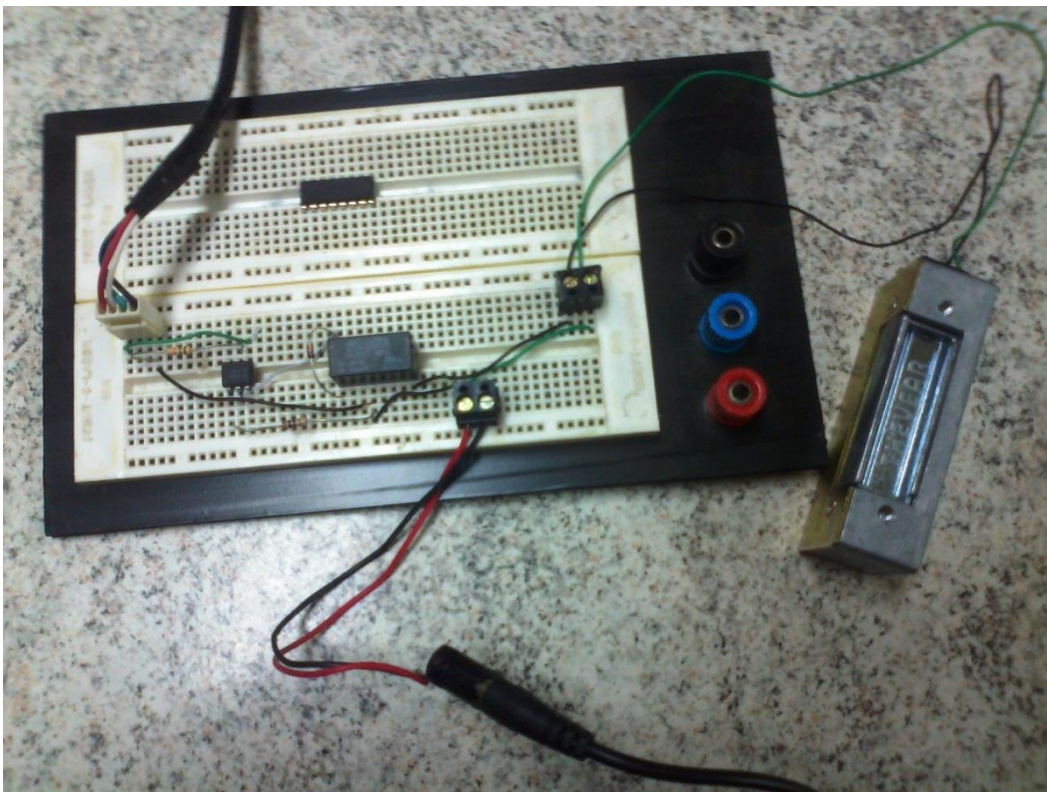


Figura 20 - Circuito SARO
Fonte: Autoria própria

4 RESULTADOS E DISCUSSÃO

4.1 RESULTADOS OBTIDOS

Para realizar os testes do sistema SARO e analisar os resultados obtidos, dividiu-se o sistema em tópicos, para facilitar o entendimento do mesmo, e achar os pontos críticos no momento de sua execução.

O cadastro de usuários: No processo de cadastro de novos usuários, o *software* criado solicita a opção de cadastro, sendo escolhida então a opção de cadastro. Solicita-se uma senha para acessar essa opção, tendo a senha negada o programa se fecha, senha autorizada passa para a fase de cadastro onde se determina identificação de acesso, após essa fase o cliente se posiciona e sua foto é tirada.

Nesta fase de teste de cadastro, onde ao tirar a foto do usuário, a imagem é armazenada em uma determinada pasta junto ao programa, em relação a este procedimento foram realizados 23 cadastros com êxito. A figura abaixo representa o processo de cadastro

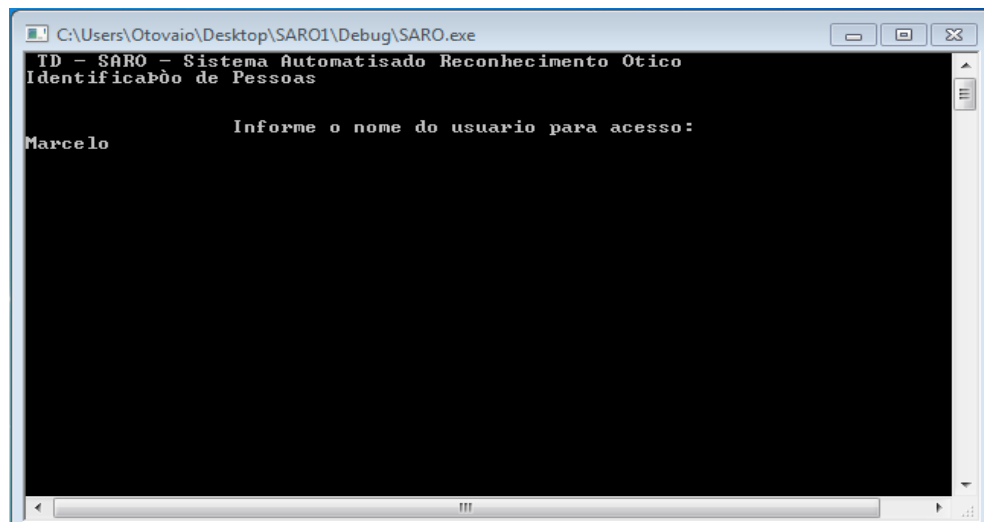


Figura 21 – Cadastro do Sistema SARO
Fonte: Autoria própria

A comparação dos dados: Nesta etapa do processo, o usuário terá ou não a permissão (acesso) para o acionamento do *hardware*, é nesta fase que se tem o ponto principal do projeto. Iniciando o *software* novamente seleciona-se a opção

de acesso, o usuário direciona o olho para a *webcam* e novamente é tirada uma nova foto, que iniciara uma comparação das mesma, essa comparação tem que ser acima de 98% de equivalência, figura 22. Para que o acesso do individuo seja permitido sendo então acionado o fecho magnético via porta paralela. Caso a isso não ocorra quer dizer que a comparação não atingiu o valor necessário ficando abaixo de 98% de equivalência não permitindo a entrada ou acesso ao local. Conforme a figura 23.

A comparação dos dados se a estudo realizado com a variável *hough*, onde nesta comparação o programa trabalha com o escalonamento de cinza, onde a partir de uma tonalidade, tonalidade essa determinada pela biblioteca do OpenCV, onde cor mais escura recebe valor binário 1 e cor mais claro recebe valor 0.

De acordo com a proposta testou-se duas câmeras, uma de 2 Mp e uma de 1.3 Mp, ajuste de foco manual. Toda vez que se inicia a câmera é necessário fazer o ajuste de foco manual na câmera, pois os ajustes anteriormente feitos, se “perdem”, outros problemas encontrados foram, chuviscos e ruídos na imagem. por não ser uma conexão exclusiva para vídeo.

```
Resultado de compatibilidade
0.999006
acesso permitido
```

Figura 22 – Acesso Permitido
Fonte: Autoria Própria

```
Resultado de compatibilidade
0.921636
Dados incompatíveis acesso negado
```

Figura 23 – Acesso negado
Fonte: Autoria Própria

O acionamento do *hardware*: Devido aos problemas com as *webcam* em relação a captura de imagem, montou-se uma própria biblioteca, comparando suas imagens e sua equivalência a parti de 98% o programa instrui o envio do sinal via porta paralela para o acionamento o fecho magnético onde o mesmo volta a travar assim que a porta abre.

4.2 DISCURSÕES

A demora em definir qual sistema operacional a ser usado nos custou muito tempo, depois de muitas tentativas e nenhum sucesso, concluímos que teríamos que mudar de sistema operacional deixar de utilizar o sistema *Linux* e começa a utilizar o sistema *Windows*.

Com o sistema operacional definido, começamos com o processo de programação, o primeiro *Software* a ser usado foi DEV C++, mas os conflitos com a biblioteca `highgui.h` da OpenCV e suas versões, tivemos que mudar de *Software*. Passando então a utilizar o Visual Studio, e mesmo assim tivemos dificuldades em configurar a biblioteca.

Depois de muitas pesquisas, vimos que para realizar a configuração da biblioteca OpenCV, no Visual Studio, era necessário a instalação de um programa chamado Cmake. Com a biblioteca configurada começamos com a estrutura de programação em C.

Finalizado o programa de captura, deu-se início ao programa de acesso, onde se teve vários conflitos e ruídos, impossibilitando a comparação de uma imagem adquirida da íris de um olho humano. Com não foi possível realizar a comparação, através do programa. Foi necessário criar um programa que comunicasse através da porta paralela, certificando o acionamento do *Hardware*.

Resolvido à etapa do Hardware, continuamos com o processo de comparação, onde foi necessário a utilização de uma foto qualquer da íris. Utilizando essa mesma foto para o cadastro e comparação, chegou-se a etapa final do processo, tendo o acesso liberado.

4.2.1 SISTEMA OPERACIONAL WINDOWS OU LINUX

No princípio do projeto tentou-se utilizar o sistema operacional Linux, mas surgiram problemas por conta de sua arquitetura, no início usou-se o Ubuntu 10.0

que utiliza arquitetura KDE 64bits, mas ocorreram problemas com a biblioteca, onde ocorria erro fatal, estudando melhor o assunto e pesquisando em fóruns especializado no sistema operacional Linux, foi visto que a versão OpenSuse 12.0 era elogiada para se trabalhar com a biblioteca Opencv, a vantagem que OpenSuse trabalhava com ambas arquiteturas KDE e Gnome. Novamente enfrentou-se problema por conta de erra falta ao adicionar as variadas bibliotecas que se necessita no OpenSuse, verificando a perda de tempo optou-se a utilizar o sistema operacional Windows xp, devido sua estabilidade, com arquitetura 32bits. Outro motivo que a biblioteca Opencv que nos foi sugerido trabalhava com 32bits e o Windows vista não apresentava confiabilidade.

4.2.2 COMPILADORES X BIBLIOTECA

Com a escolha do sistema operacional, o próximo passo foi escolher o compilador e a biblioteca, pesquisando sobre o OpenCV e verificou-se que vários projetos universitários se destacavam dois compiladores DEV C++ e Visual Studio.

Infelizmente por se tratar de uma biblioteca de código aberto, ocorreu o problema de compatibilidade de versão, ao usar o DEV C++, utilizou-se a versão 1.0 e 1.1, mas conflitos com as bibliotecas e o compilador nos custaram muito tempo, vários erros ocorriam e o projeto não andava. Mudou-se para outro compilador, utilizando versão free do compilador Microsoft Visual Studio 2010.

Definido compilador pesquisou-se novamente melhor maneira de trabalhar na criação do software, em fóruns e no próprio site da OpenCV vimos que apesar de teste de varias versões novas, a versão 2.1 para sistema 32bits era mais estável. Porém muitas informações de fórum eram picadas e do próprio site do OpenCV vários usuários modificavam as versões criando conflito com compilador.

5 CONCLUSÕES

Devido a vários problemas encontrados durante o projeto, teve que se criar dois programas, um para cadastro sendo executado com sucesso, correspondendo a nossas expectativas, já o outro programa criado foi para o acesso do usuário, sendo esse a parte fundamental para o funcionamento do projeto. Mas devido aos problemas com a *webcam*, não foi possível a excursão perfeita do programa. Tendo em vista dos problemas teve-se que executar o programa de acesso utilizando uma mesma imagem para cadastro e acesso, obtendo com êxito o proposto do projetos.

Tendo em vista os problemas ocorridos, foi verificado que a criação de um sistema de captura de imagem da íris para geração de uma identificação única, é inviável utilizando materiais limitados. Utilização de *webcam* não é viável, devido sua baixa resolução em trabalha com zoom e macro e o fato de ser uma câmera que trabalha com a aquisição “branca” proporciona erros variados.

O computador com processador de um único núcleo apresenta dificuldade em processar o sistema, sua placa de vídeo não possui configuração suficiente para trabalhar com gráficos.

O sistema criado utilizando a biblioteca *OpenCV* corresponde a expectativas e onde mostrou precisão na diferenciação de imagens, não apresentando falhas no presente momento.

5.2 SUGESTÕES E FUTURAS MELHORIAS

Para um primeiro passo foi a utilização de uma câmera infravermelho, onde a luz não apresentaria reflexo, a câmara infravermelha trabalha diretamente com escala de cinza, facilitando a transformada Hough.

Outra possível solução seria de utilizar uma câmera especial vinculada a uma lupa própria, porem o custo seria muito alto fugindo da proposta do projeto.

Em um segundo passo a utilização de um micro com placa de vídeo com quatro canais de vídeo, onde o tratamento da imagem seria discretizado. Podendo

assim, desenvolver o sistema em com sistema de 64 bits, onde hoje é a realidade atual.

REFERÊNCIAS

CHAVEZ, Roger F. L.; IANO, Yuzo; SABLON, Vicente I. B. Processo de Reconhecimento de Íris Humana: Localização Rápida de Íris. **REVISTA CIENTÍFICA PERIÓDICA - TELECOMUNICAÇÕES**, Campinas, v. IX, n. 01, Novembro 2006.

DAUGMAN, John. How Iris Recognition Works. **IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY**, Cambridge, U.K., XIV, January 2004.

DUARTE, Otto C. M. B.; VALADÃO, Renan B. Biometria - Reconhecimento de Íris. **GTA - Grupo de Teleinformática e Automação**, 2008. Disponível em: <http://www.gta.ufrj.br/grad/08_1/iris/index.html>. Acesso em: 20 ago. 2011.

EMGUCVWIKI. **EmguCV**, 2012. Disponível em: <http://www.emgu.com/wiki/index.php/Main_Page>. Acesso em: maio 2012.

FERNANDES, Tomas. **Identificação de Indivíduos por Biometria da Íris usando Função Densidade de Coincidências de Fase**. 2009. Dissertação de Mestrado do Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. Universidade Tecnológica Federal do Paraná, Curitiba, 2009.

FERNANDO ALVES, Liliane C. **Estudo da Porta Serial e Paralela**. Instituto Superior Politécnico Gaya - ISPGAYA. [S.l.]. 2003.

FILHO, Ogê M.; NETO, Hugo V. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. ISBN 8574520098.

FRIGERI, Alceu H.; COPSTEIN, Bernardo; PEREIRA, Carlos E. **Curso de C++**. Porto Alegre: [s.n.], 1996.

GAZZONI, Jean C.; MARCON, Cesar L.; FRANCISCON, Marcio A. **SARPA Sistema Automatizado de Reconhecimento de Placas Automotivas**. 2006. Trabalho de Conclusão de Curso - Universidade de Tecnológica Federal do Paraná, Pato Branco, 2006.

GRANCHO, Emanuel. **Reconhecimento Biométrico Periocular: Segmentação de dados**. Universidade Da Beira Interior. Covilhã, p. 68. 2011.

GROUP, The K. OpenGL. **The Industry's Foundation for High Performance Graphics**, 1997. Disponível em: <<http://www.opengl.org/about/>>. Acesso em: maio 2012. from games to virtual reality, mobile phones to supercomputers.

ID IRIS, Inc. IRIS ID. **site da LG IRIS**, 2000. Disponível em: <<http://www.irisid.com/>>. Acesso em: jun. 1012.

JÚNIOR, Celso D. A. P. **Biometria com Enfoque em Reconhecimento de Íris**. 2005.UEL - Universidade Estadual de Londrina, Londrina, 2005.

MENDES, Carlos P. LG lança aparelho que reconhece íris do olho. **Jornal Eletrônico Novo Milênio**, 2000. Disponível em: <<http://www.novomilenio.inf.br/ano00/0004a002.htm>>. Acesso em: 25 abr. 2012.

OPENCVWIKI. **OpenCV**, 2008. Disponível em: <<http://opencv.willowgarage.com/wiki/Welcome>>. Acesso em: 15 mar. 2012.

OST, Flavia N. **Visão Computacional e Arte Digital**. 2008.USP, São Paulo, 2008.

PANASONIC. © Panasonic Corporation 2012, 2012. Disponível em: <<http://panasonic.net/security>>. Acesso em: 25 abr. 2012.

REHEM, Almerindo; TRINDADE, Fernando H. V. REAL PORTAL, Portal do Almerindo Rehem. **Técnicas de Visão Computacional para Rastreamento de Olhar em Videos**, 2009. Disponível em: <[http://almerindo.devin.com.br/index.php?option=com_content&view=article&id=78%3Atecnicas-de%20-computacao%20-visual-para%20-rastreamento-de%20-olhar-em-videos&catid=43%3Atrabalhos%20-de%20-alunos&Itemid=86&showall=1](http://almerindo.devin.com.br/index.php?option=com_content&view=article&id=78%3Atecnicas-de%20computacao%20-visual-para%20-rastreamento-de%20-olhar-em-videos&catid=43%3Atrabalhos%20-de%20-alunos&Itemid=86&showall=1)>. Acesso em: 02 abr. 2012.

RUDEK, Marcelo; COELHO, Leandro D. S.; JUNIOR, Osiris C. **VISÃO COMPUTACIONAL APLICADA A SISTEMAS PRODUTIVOS: FUNDAMENTOS E ESTUDO DE CASO**. Curitiba: Pontifícia Universidade Católica do Paraná, PUCPR/LAS/CCET, 2001.

SCHILDT, Herbert. **Turbo C++: Guia do Usuário**. São Paulo: Makron Books, 1992.

SILVA, Thiago R. D. **DESENVOLVIMENTO DE UMA PLACA PARA AQUISIÇÃO DE SINAIS BIOMÉDICOS COM INTERFACE USB**. 2006.Centro Tecnológico da Universidade Federal do Espírito Santo, Vitória, 2006.

THEVEAR. **Thevear Eletrônica Ltda.**, 2008. Disponível em: <<http://www.thevear.com.br/New/VerProdutos.aspx?IDSubCat=143>>. Acesso em: 15 nov. 2011.

TISSE, Lionel T. C.-L.; MARTIN, Lionel; ROBERT, Michel. Person identification technique using human iris recognition. [S.l.]: St Journal of System Research, v. IV, 2003. p. 67-75.

VALADÃO, Renan B. Biometria - Reconhecimento de Íris. **GTA - Grupo de Teleinformática e Automação - UFRJ - Escola Politécnica**, 2008. Disponível em: <http://www.gta.ufrj.br/grad/08_1/iris/index.html>. Acesso em: 15 nov. 2011.

VILLAS-BOAS, Sergio B. **C++ Multiplataforma e Orientação a Objetos. sbVB**. Disponível em: <www.sbv.com.br>. Acesso em: 10 abr. 2012.

WANGENHEIM, Aldo V.; COMUNELLO, Eros. PPGCC - INE - UFSC. **Seminário Introdução à Visão Computacional. The Cyclops Project**, 2005. Disponível em: <<http://www.inf.ufsc.br/~visao/#Introdu%C3%A7%C3%A3o>>. Acesso em: 02 abr. 2012.

APÊNDICE A – CAPTURA DE IMAGEM

```

void cadastro(void){
    CvCapture* capture;
    IplImage* frame;
    CvSize imgSize;
    imgSize.width = 360;
    imgSize.height = 280;
    IplImage* gray_frame = cvCreateImage(imgSize, IPL_DEPTH_8U, 1);
    double reg = 0;
    int j = 0;
    int v = 0;

    system("CLS");
    printf(" TD - SARO - Sistema Automatizado Reconhecimento Otico\n");
    //CvCapture* capture = 0;
    capture=cvCaptureFromCAM( CV_CAP_ANY );
    printf("Identificação de Pessoas\n");

    while(1)
    {
        // comando de captura
        IplImage* frame = cvQueryFrame(capture);
        if( !frame )
        {
            fprintf( stderr, "ERROR: frame is null...\n" );
            getchar();
            break;
        }

        // Salvando arquivo
        char file[255];
        printf("\n\n\t\t Informe o nome do usuario para cadastro /
acesso:\n");
        scanf("%s",file);
        strcat(file, ".bmp");
        cvSaveImage(file,frame);
        (((cvWaitKey(10) & 255) == 27) || fechado == 1 );
        capture = cvCreateFileCapture(file, ".bmp");// tenho que pegar a
foto e já começar a fazer o tratamento neste ponto.
        FILE *f = fopen ("IRIS1.TXT", "wt");
        while(1) {
            frame = cvQueryFrame(capture);
            if(!frame) break;
            cvSmooth(frame,frame);
            cvCvtColor(frame, gray_frame, CV_RGB2GRAY);
            CvMemStorage* storage = cvCreateMemStorage(0);
            if(gray_frame->origin != IPL_ORIGIN_TL)
            {
                cvFlip(gray_frame, gray_frame, 0);
            }

            CvSeq* results = cvHoughCircles(gray_frame, storage,
CV_HOUGH_GRADIENT, 1, gray_frame->width,100,1,10,50);
            for(int i = 0; i<results->total; i++)
            {
                float* p = (float*) cvGetSeqElem (results, i);
                CvPoint pt = cvPoint(cvRound(p[0]), cvRound(p[1]));
                cvCircle(gray_frame,pt,cvRound(p[2]), CV_RGB(0xff,0xff,0xff));
            }
        }
    }
}

```

APÊNDICE B – COMPARAÇÃO DE IMAGEM

```

int _tmain(int argc, _TCHAR* argv[]){

//imagem para cadastrada
IplImage *IMG_Original = cvLoadImage("c:/ttd.jpg");

//imagem tentando acesso manualmente
IplImage *IMG_Template = cvLoadImage("c:/ttd.jpg");

while(1){

//IMAGEM RESULTANTE
IplImage *IMG_Result = cvCreateImage(cvSize(IMG_Original->width - IMG_Template-
>width+1,
                                                    IMG_Original-
>height - IMG_Template->height+1),IPL_DEPTH_32F, 1);

cvMatchTemplate(IMG_Original, IMG_Template, IMG_Result, CV_TM_CCORR_NORMED);

//Determina valor minimo e maximo
double min_val = 0, max_val = 0;
CvPoint min_loc, max_loc;
cvMinMaxLoc(IMG_Result, &min_val, &max_val, &min_loc, &max_loc);

//medição da area
cvRectangle(IMG_Original, max_loc, cvPoint(max_loc.x+IMG_Template->width,
max_loc.y+IMG_Template->height), cvScalar(0,0,255,0), 1);

```