

CONIC-SEMESP 13º Congresso Nacional de Iniciação Científica

Anais do Conic-Semesp. Volume 1, 2013 - Faculdade Anhanguera de Campinas - Unidade 3. ISSN 2357-8904

TÍTULO: RECONHECIMENTO DE GESTOS - OPENCV

CATEGORIA: EM ANDAMENTO

ÁREA: ENGENHARIAS E TECNOLOGIAS

SUBÁREA: COMPUTAÇÃO E INFORMÁTICA

INSTITUIÇÃO: FACULDADE ANHANGUERA DE BELO HORIZONTE

AUTOR(ES): MARGHEL RAIMUNDO DA COSTA VASCONCELOS, BRUNO ALEXANDRE PEREIRA,
SERGIO DIAS VIEIRA

ORIENTADOR(ES): LINDEMBERG NAFFAH FERREIRA

Realização:



Apoio:



RECONHECIMENTO DE GESTOS – OPENCV

Resumo

Este projeto tem por objetivo produzir conteúdo que possa auxiliar futuros pesquisadores que estejam realizando seus trabalhos com foco em reconhecimento de gestos, mapeamento de imagens, e aprendizado de máquina, utilizando o framework OpenCv para seu desenvolvimento. Seu propósito é auxiliar, especialmente, aqueles projetos que tornem possível o reconhecimento da linguagem brasileira de sinais –LIBRAS

Introdução

A falta de conhecimento da população na interpretação de sinais (LIBRAS) causa exclusão não intencional da comunidade dos surdos na sociedade.

Visando facilitar a interação entre portadores de deficiência auditiva e a população, empregaremos os resultados obtidos nas pesquisas em andamento, no intuito de produzir um material para auxiliar futuros pesquisadores na busca de novas tecnologias com o foco voltado para o uso da inteligência artificial no reconhecimento de gestos utilizando o framework Opencv para o desenvolvimento. De acordo com a LEI Nº 10.436, DE 24 DE ABRIL DE 2002, pode-se dizer que a Língua Brasileira de Sinais – Libras é ' a forma de comunicação e expressão, em que o sistema linguístico de natureza visual-motora, com estrutura gramatical própria, constituem um sistema linguístico de transmissão de ideias e fatos, oriundos de comunidades de pessoas surdas do Brasil.”

/.

1. Objetivos

O objetivo geral do projeto é investigar a possibilidade da utilização do framework Opencv associado a técnicas de inteligência artificial, a fim de produzir conteúdo sobre o reconhecimento de gestos utilizando as técnicas descritas. É possível encontrar na literatura diversas formas para o reconhecimento de gestos. Entretanto,

apesar dos resultados obtidos, há desafios relacionados ao tratamento dos dados coletado, uma vez que pode haver distorção do resultado final, limitando o sucesso desse processo.

2. Metodologia

A metodologia contempla a utilização da linguagem de programação python para a implementação dos algoritmos a serem utilizados juntamente com o ambiente de desenvolvimento integrado - (cuja sigla em língua inglesa é IDE) Eclipse, que permite a configuração da biblioteca Opencv. Devido ao baixo custo e ampla utilização do dispositivos móveis que possuem o sistema operacional Android, para obtenção das imagens em tempo real, utilizou-se o software IPCAMERA que captura os *frames* através de celulares e *tablets* e os compartilha, empregando para esse fim protocolos de rede da pilha TCPIP, possibilitando a visualização dos dados capturados por uma conexão local via cabo ou via rede sem fio local (*wi-fi*).

3. Desenvolvimento

Sistema Operacional

O sistema operacional utilizado foi o Ubuntu devido a sua extensa comunidade de usuários e facilidade de adaptação. Não será abordada neste artigo a instalação do sistema operacional, valendo dizer que, tão somente, que as instruções para realizá-la pode ser encontrada no endereço <http://www.ubuntu-br.org/>.

Eclipse

Eclipse é uma comunidade de indivíduos e organizações que desejam colaborar em software de código aberto comercialmente amigável. Seus projetos são focados na construção de uma plataforma de desenvolvimento aberta composta de estruturas extensíveis, ferramentas e tempos de execução para a construção, implantação e gerenciamento de software em todo o ciclo de vida. A Fundação Eclipse é uma

organização sem fins lucrativos, apoiada em corporações-membro que hospeda os projetos Eclipse e ajuda a cultivar tanto uma comunidade de código aberto e um ecossistema de produtos e serviços complementares.

[<http://www.eclipse.org/org/> Acessado em : 10/07/2013]

O repositório padrão do Ubuntu já contém os pacotes necessários para instalação do Eclipse. Para efetuá-la basta acessar o terminal e digitar os comandos abaixo.

```
sudo apt-get update  
sudo apt-get install eclipse
```

Após concluir a instalação para iniciar a ide utilizamos o comando “eclipse”.

Python

Python é uma linguagem de programação orientada a objetos utilizada por grandes empresas da área de tecnologia da informação, tais como Google, porém não muito utilizada nas demais empresas. A opção pela sua utilização deve-se à facilidade de implementação dos métodos disponibilizados pelo framework Opencv.

O interpretador python já está incluso nas distribuições Linux. Para que ele possa ser utilizado em conjunto com o IDE Eclipse, faz-se necessário instalar um plugin “Pydev” que permite desenvolver a aplicação de forma rápida e precisa. A instalação e configuração do Pydev na IDE Eclipse está descrita abaixo. Deve-se iniciar a IDE Eclipse pelo terminal, mediante execução do comando

```
sudo eclipse
```

Com a IDE em execução deve-se acessar através do menu “Help” o sub-menu “Install New Software” . Após visualizar a tela conforme a Figura 1, deve-se clicar no botão “Add” .

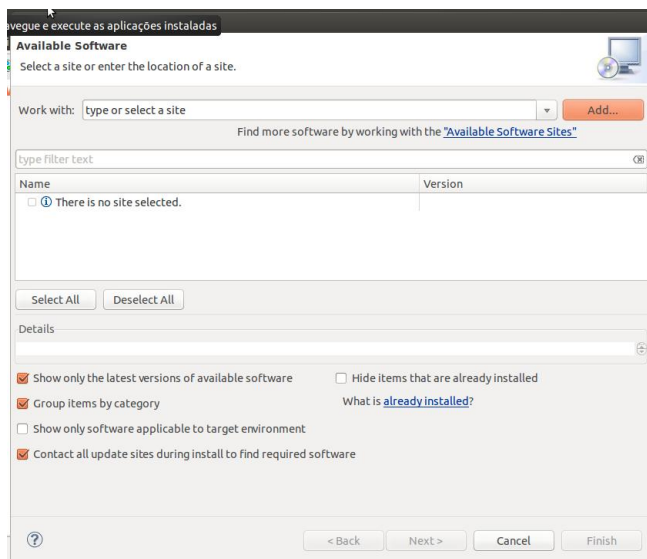


Figura 1 – Tela de plugins disponíveis no Eclipse

Deve-se adicionar um novo repositório. Na propriedade “Name”, deve-se preencher como “PyDev” e, em “Location”, adicionar a URL :

<http://update-production-pydev.s3.amazonaws.com/pydev/updates/site.xml>

Na tela onde os resultados são exibidos, deve-se selecionar os dois itens disponíveis e clicar no botão “Next”.

Será exibida uma janela com licença do Pydev, devendo-se assinalar “agreement” e clicar em “Next”.

Para configurar o interpretador python, deve-se acessar o menu “Window|Preferences” , localizar na aba a esquerda “Pydev” clicar para expandir e selecionar a opção “Interpreter –python” e clicar no botão “Auto config”, como pode ser visualizado na Figura 2.

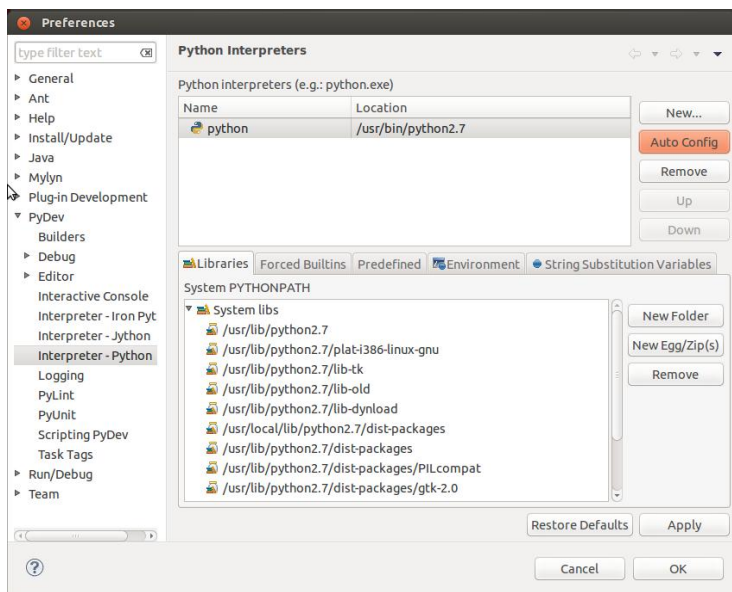


Figura 2 – Tela para seleção do ambiente de desenvolvimento integrado - Eclipse

OpenCv

OpenCV (Open Source Computer Vision) (OpenCV, 2013) é uma biblioteca de funções de programação de visão computacional em tempo real.

<http://opencv.willowgarage.com/wiki/> Acessado em : 10/07/2013]

Os passos para instalação do OpenCv estão descritos abaixo .

Deve-se acessar o terminal e digitar o comando::

```
sudo apt-get install g++ libopencv-dev
```

Para utilizar a biblioteca com o interpretador python, deve-se instalar a lib “python-opencv”.

```
sudo apt-get install python-opencv
```

IP Webcam

È um software livre, disponibilizado na playstore, que transmite via IP as imagens capturadas por um smartphone Android, permitindo que sejam visualizadas por aplicativos que utilizam vídeo streaming MJPG, ou pelo WebBrowse.

Para a instalação deve-se acessar o repositório de software para o ambiente operacional Android através do *smartphone* ou *tablet* e procurar pelo app “Ip Webcam”.

Captura e processamento de imagens

Nesta etapa procura-se compreender o funcionamento dos métodos utilizados para obtenção das imagens, tendo sido aplicados alguns filtros para observarmos as mudanças obtidas.

A captura das imagens foi realizada através de um *smartphone*, transmitindo-as em tempo real pelo protocolo de rede TCP/IP, como pode ser visualizado na figura 3.

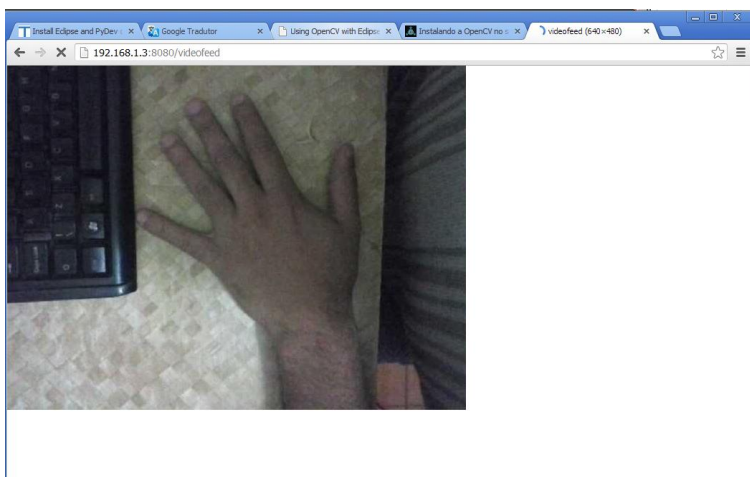


Figura 3 – Imagem capturada pelo Ip Webcam.

A implementação do algoritmo utilizado para captura das imagens utilizando a biblioteca OpenCv é simples e, se necessário, a biblioteca disponibiliza toda a documentação sobre os métodos utilizados.

Segue abaixo descrição da implementação do algoritmo utilizado para capturar as imagens transmitidas pelo IP WebCam.

```
import cv2.cv as cv
```

```

print "Iniciando Captura"
capture = cv.CaptureFromFile("http://192.168.1.3:8080/videofeed?
something.mjpg")
while True:
    image = cv.QueryFrame(capture)
    cv.ShowImage("Teste", image)
    k = cv.WaitKey(10);
    if k == 'f':
        break

```

A implementação de algoritmo utilizado para aplicar a escala de cinza na imagem carregada, pode ser visualizado na Figura 4.

```

import cv2.cv as cv
image = cv.LoadImage("mao.jpeg")
image_cinza =cv.LoadImage("mao.jpeg",cv.CV_LOAD_IMAGE_GRAYSCALE)
while True:
    cv.ShowImage("Imagem Natural", image)
    cv.ShowImage("Imagem Escala Preto e branco", image_cinza)
    k = cv.WaitKey(10);
    if k == 'c':
        break

```



Figura 4 – Imagem Original e imagem na escala de cinza.

Extração de contorno

Nesta etapa são analisados os métodos utilizados em busca de extração e redução do número de coordenadas emitidas a cada *frame*

Uma implementação do algoritmo utilizado para extração do contorno pode ser visualizada na Figura 5.

```
import cv2
import numpy as np
imagem = cv2.imread('mao.jpeg')
imagem_cinza = cv2.cvtColor(imagem,cv2.COLOR_BGR2GRAY)
imagem_nova = np.zeros(imagem.shape,np.uint8)
ret,thresh = cv2.threshold(imagem_cinza,127,255,0)
contours, hierarchy = cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
for cnt in contours:
    color = np.random.randint(0,255,(3)).tolist()
    cv2.drawContours(imagem_nova,[cnt],0,color,2)
while True:
    cv2.imshow("Contorno da imagem",imagem_nova)
    cv2.imshow("Imagem Original",imagem)
    k = cv2.waitKey(10);
    if k == 'c':
        break
```

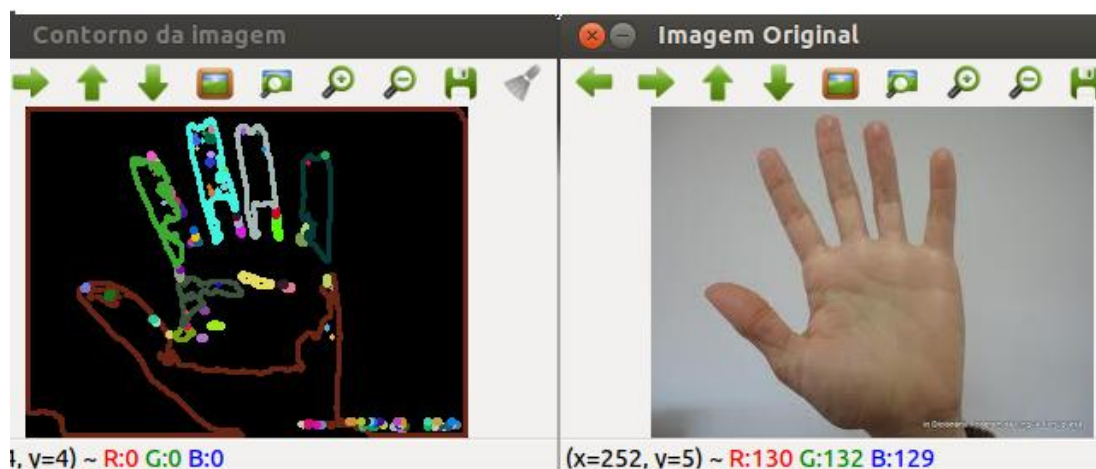


Figura 5 – Imagem com contorno e imagem original

4. Resultados

Efetuar o reconhecimento de gestos, ou seja, mapear as coordenadas de uma imagem, tornou-se um desafio. Através da extração do contorno de uma imagem, foi possível identificar os pares ordenados, tornando possível reproduzir o traçado realizado para a formação do desenho. Durante os testes realizados, não foi possível definir um número fixo de pontos a serem verificados. A princípio, definiu-se como objetivo alinhar um ponto em cada dedo e reproduzir a trajetória do momento em que se inicia um movimento até sua conclusão. Porém ao sobrepor as mãos, não foi possível distinguir os pontos e perdeu-se a referência. O que se propõe como trabalho futuro, a ser realizado nas próximas etapas deste projeto, é avançar na extração destas coordenadas e conseguir sintetizar o resultado a ser comparado. Todas as verificações ocorreram com base em apenas 1 (um) quadro. O que está sendo investigado é a obtenção de uma solução que minimize o número de dados a serem produzidos em cada quadro, pois na captura em tempo real obtém-se cerca de 25 quadros por segundo. Ainda tem-se como objetivo o emprego de técnicas de inteligência artificial para a comparação dos dados. Vale dizer que a

biblioteca OpenCv já possui métodos capazes de permitir a realiza aprendizado de máquina e reconhecimento de imagens. Entretanto grande parte das implementações realizadas usando esses recursos buscam o reconhecimento da face humana.

5. Considerações Finais

Mesmo com os desafios encontrados durante a fase inicial da pesquisa, os resultados atingidos são satisfatórios e estão dentro do esperado. Constatou-se que as ferramentas utilizadas são de grande potencial. Ademais, o assunto que objeto desta pesquisa é de grande interesse, particularmente para a comunidade usuária de LIBRAS, o que serve de estímulo para a busca de soluções para os problemas encontrados.

6. Fontes Consultadas

Python Programming Language - <http://www.python.org/>

Acessado : 10/07/2013.

Ubuntu Brasil - www.ubuntu-br.org Acessado : 10/07/2013

CookBook OpenCv

<http://opencv.willowgarage.com/documentation/python/cookbook.html>

Acessado: 09/07/2013.

Python Linux Núcleo

<http://linuxcpdti.blogspot.com.br/2013/07/script-de-instalacao-do-opencv-246-no.htm>

Acessado :10/07/2013.

OpenCv Python Examples

<http://www.betasix.net/opencv-2-2-python-examples>

Acessado:08/07/2013.

Eclipse - www.eclipse.org/

Acessado 10/07/2013.

PyDev - <http://pydev.org/>

Acessado: 07/07/2013.