

Processamento de Imagem e Vídeo no Controle de Tráfego

Hae Yong Kim,

Dept. Eng. Sist. Eletrônicos, Escola Politécnica, USP

Seminário 15/maio/2008

Objetivos:

- Mostrar as possibilidades que processamento de imagem e vídeo (PIV) pode oferecer no controle de tráfego.
- Verificar possíveis áreas e assuntos de cooperação LEMT - LPS.

Índice:

1. Sistemas Proeikon, FFMPEG, OpenCV
Exemplo: Segmentação pelo crescimento de semente usando cor.
2. Subtração do fundo
 - 2.1. Contagem de carros e detecção de velocidade
 - 2.2. Controle de vagas em estacionamentos
 - 2.3. Cálculo de comprimento e altura de veículos e caminhões
3. Transformada de Hough
 - 3.1. Detecção de placa limite de velocidade
4. Template matching
 - 4.1. Leitura da placa de carro
 - 4.2. Reconhecimento do modelo do carro
 - 4.3. Contagem de número de eixos em caminhões
 - 4.4. Detecção de face
 - 4.5. Outras aplicações
5. Marca d'água de autenticação
6. Hardware especial
 - 6.1. Processamento de webcam
 - 6.2. DSP/FPGA

1 Sistemas Proeikon, FFMpeg, OpenCV

Proeikon consiste de:

- a) Um conjunto rotinas para processamento de imagem e vídeo (PIV) escrito em C++;
- b) Um conjunto de programas para PIV escritos utilizando as rotinas acima.

Proeikon incorpora algumas rotinas de outros autores, por exemplo:

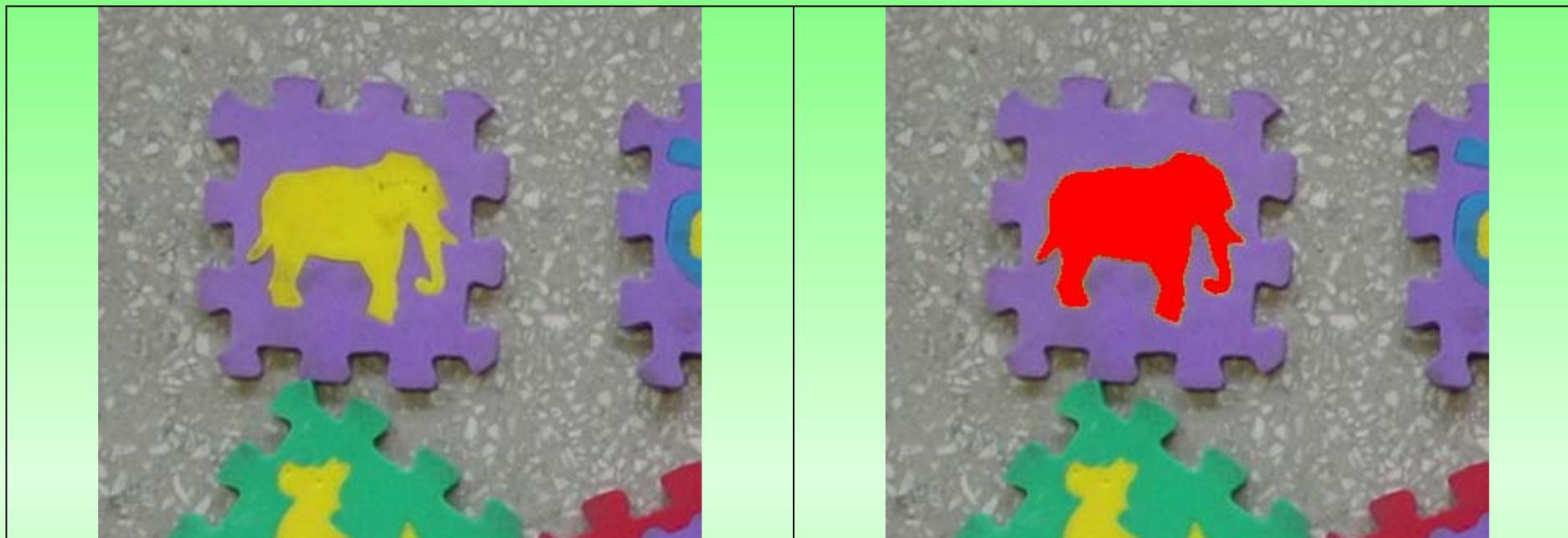
- a) As rotinas de leitura/escrita de JPEG.
- b) As rotinas de FFT.

Proeikon pode trabalhar em conjunto com as bibliotecas:

- a) As rotinas FFMPEG para leitura/gravação de vídeos.
- b) As rotinas de OpenCV (inclui uso de webcam, FFT acelerado por MMX, interface gráfica simplificada).

Proeikon vem em formato DLL. Isto permite utilizá-lo juntamente com os mais variados compiladores e linguagens.

Um exemplo: Segmentação de regiões com cores semelhantes pelo crescimento de semente.



```

#include <proeikon>
#include <queue>

int distancia(COR x, COR y)
{ return abs(x.r()-y.r()+abs(x.g()-y.g()+abs(x.b()-y.b())); }

void pintacffi(IMGCOR& b, int lsemente, int csemente)
{ const int limiar=150;
  COR inicial=b(lsemente,csemente);
  queue<int> q; q.push(lsemente); q.push(csemente);
  while (q.empty()==false) {
    int l=q.front(); q.pop(); int c=q.front(); q.pop();
    if (distancia(b(l,c),inicial)<limiar) {
      b(l,c)=COR(255,0,0);
      q.push(l-1); q.push(c);
      q.push(l); q.push(c-1);
      q.push(l); q.push(c+1);
      q.push(l+1); q.push(c);
    }
  }
}

int main()
{ IMGCOR b; le(b,"elefante.jpg"); b.backg()==COR(0,0,0);
  int lsemente=116; int csemente=136;
  pintacffi(b,lsemente,csemente);
  imp(b,"elefante2.tga");
}

```

2 Subtração do fundo

Muitos problemas podem ser resolvidos simplesmente subtraindo o fundo. Dificuldades:

- Brilho/contraste do fundo varia. É possível corrigir a variação de brilho/contraste do fundo.
- A sombra pode atrapalhar. Possível solução: usar invariância de cores (a ser testado).



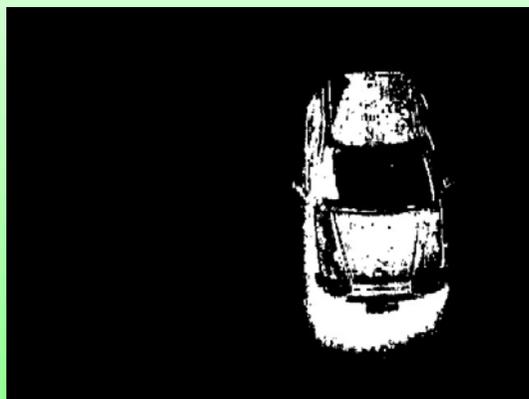
Frame 280 do trafego.avi



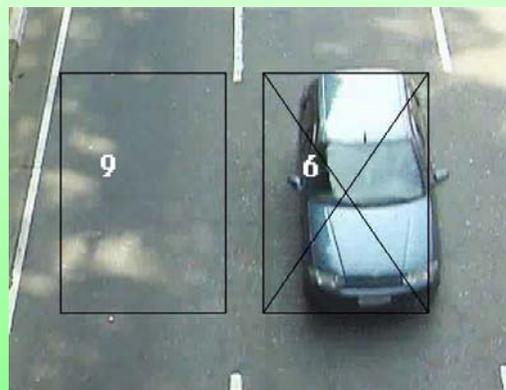
Após correção do brilho e contraste



Subtração de fundo



Binarização



Passaram 9 e 6 carros desde o início do vídeo nas duas faixas.

2.1 Contagem de carros e detecção de velocidade

Projeto desenvolvido com auxílio dos alunos do segundo ano e do prof. Hugo.

Objetivo: Substituir detector de carros indutivo pelo processamento de vídeo.

É possível distinguir moto, carro e caminhão.

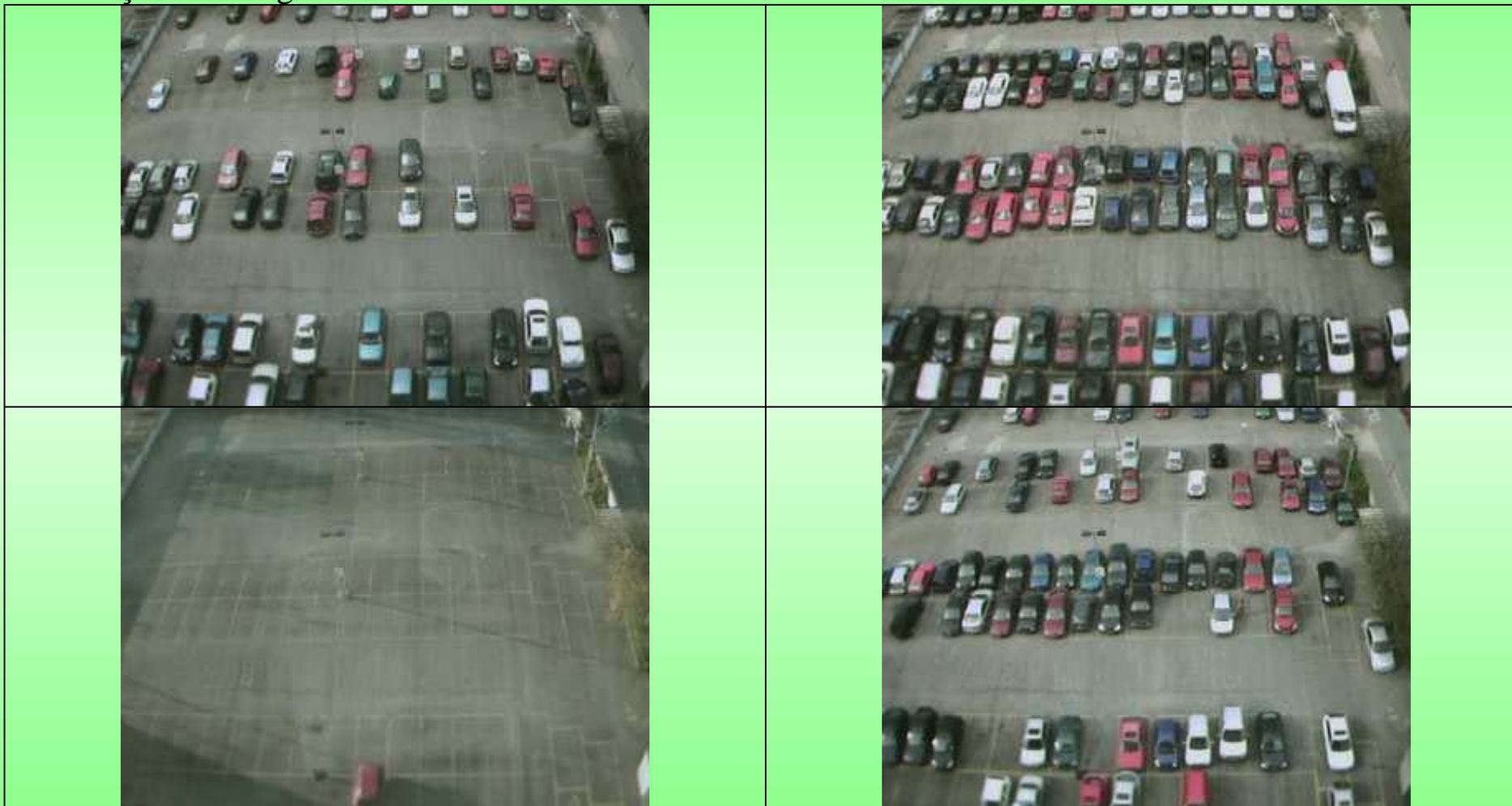
É possível detectar a velocidade aproximada.

Problemas:

- Carros e motos que passam no meio das duas faixas: É possível seguir a trajetória do carro/moto.
- É possível detectar ultrapassagens proibidas, carro que anda pelo acostamento, etc.

2.2 Controle de vagas em estacionamentos.

Projeto de formatura. Com câmera fixa e usando subtração de fundo, é possível reconhecer a quantidade e as localizações das vagas livres num estacionamento.



2.3 Cálculo de comprimento e altura de veículos e caminhões

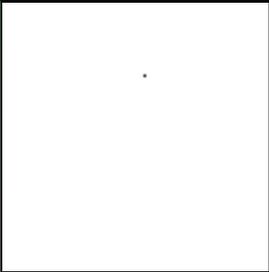
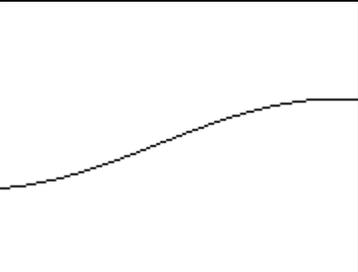
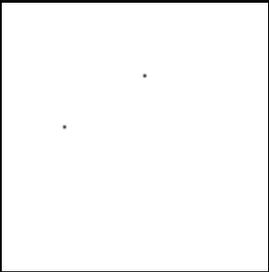
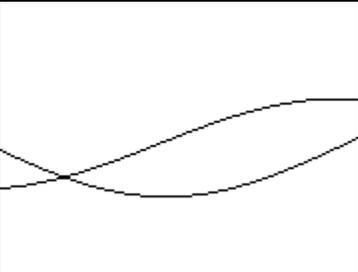
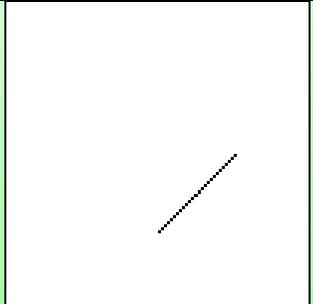
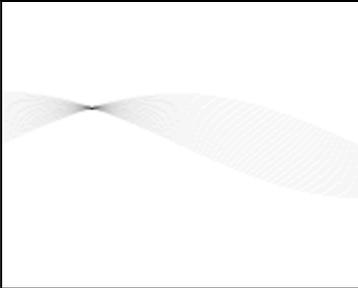
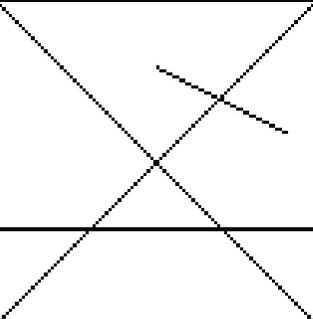
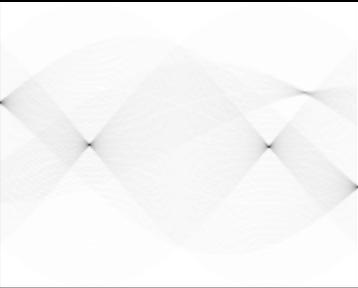
Com subtração de fundo e câmeras colocadas em posições estratégicas, é possível calcular as dimensões dos caminhões.

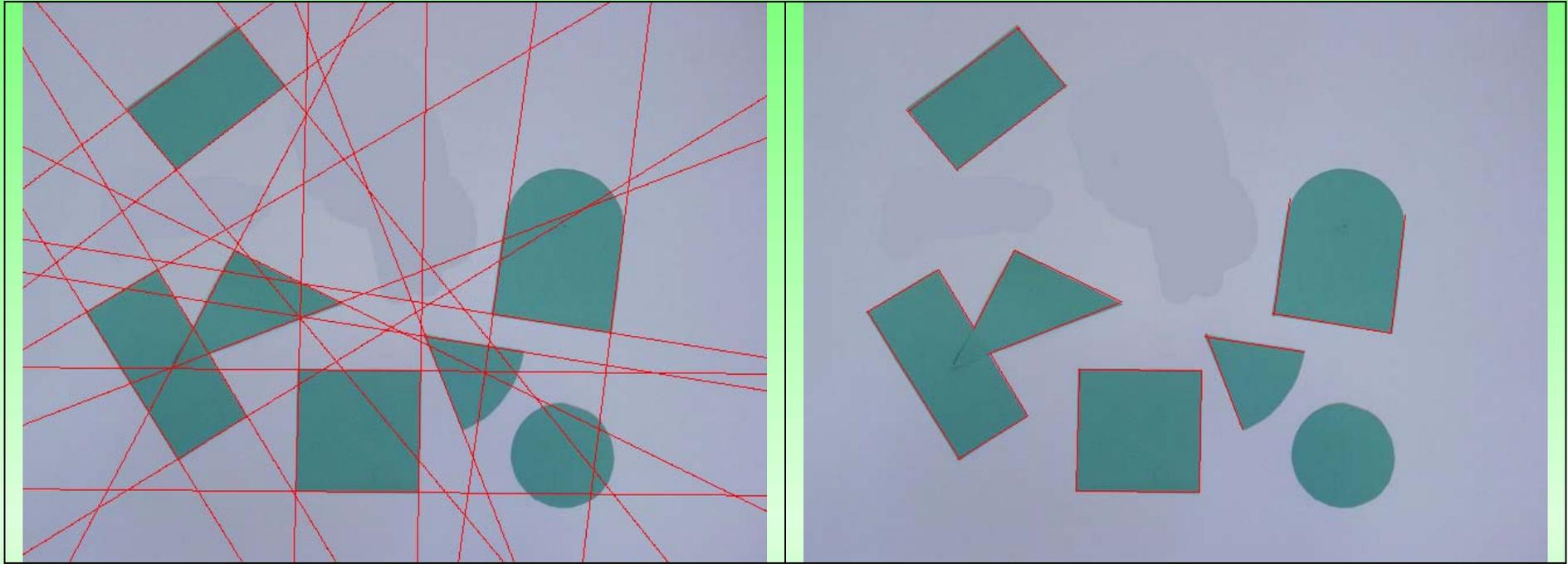
Possivelmente, é possível calcular as dimensões até com caminhões em movimento.

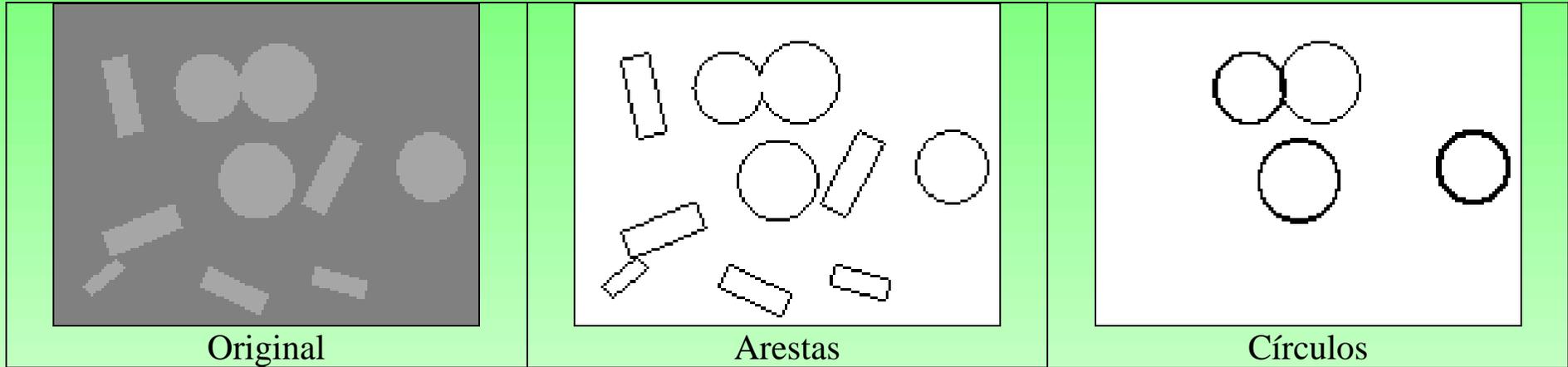
A acuracidade possível de ser obtida depende dos testes.

3 Transformada de Hough

A transformada de Hough é capaz de detectar grupos de pixels que pertencem a uma linha reta.

Imagem original	Accumulator cells (espaço de Hough)	Imagem original	Accumulator cells (espaço de Hough)
 <p>(a) 1 ponto</p>	 <p>(b) 1 senóide</p>	 <p>(c) 2 pontos</p>	 <p>(d) 2 senóides. O ponto de intersecção representa a reta que passa pelos 2 pontos.</p>
 <p>(e) uma reta</p>	 <p>(f) Infinitos senóides que intersectam num único ponto. O ponto de intersecção representa a reta.</p>	 <p>(g) 4 retas</p>	 <p>(h) Infinitos senóides que se acumulam em 4 pontos (os 2 pontos de acumulação nas bordas laterais correspondem à reta horizontal).</p>





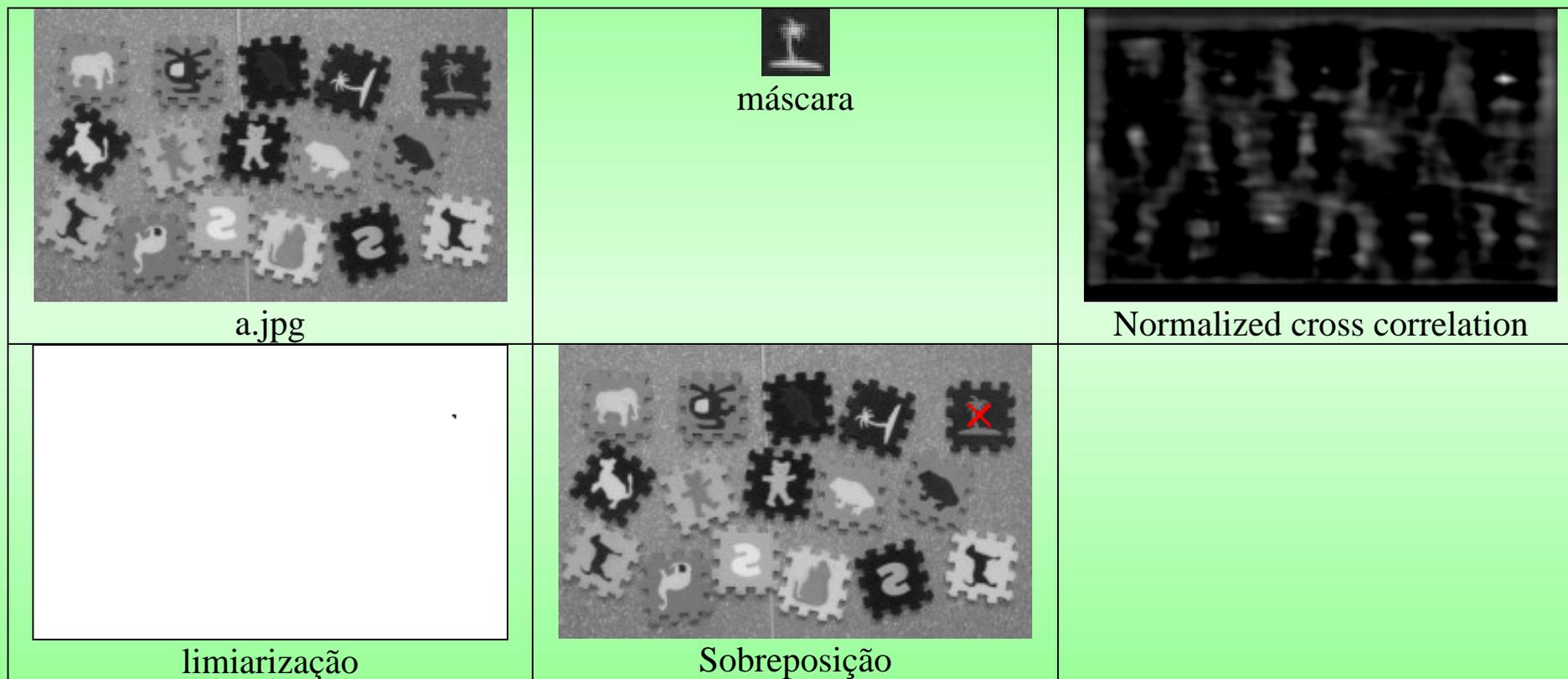
3.1 Detecção de placa de limite de velocidade



4 Template matching

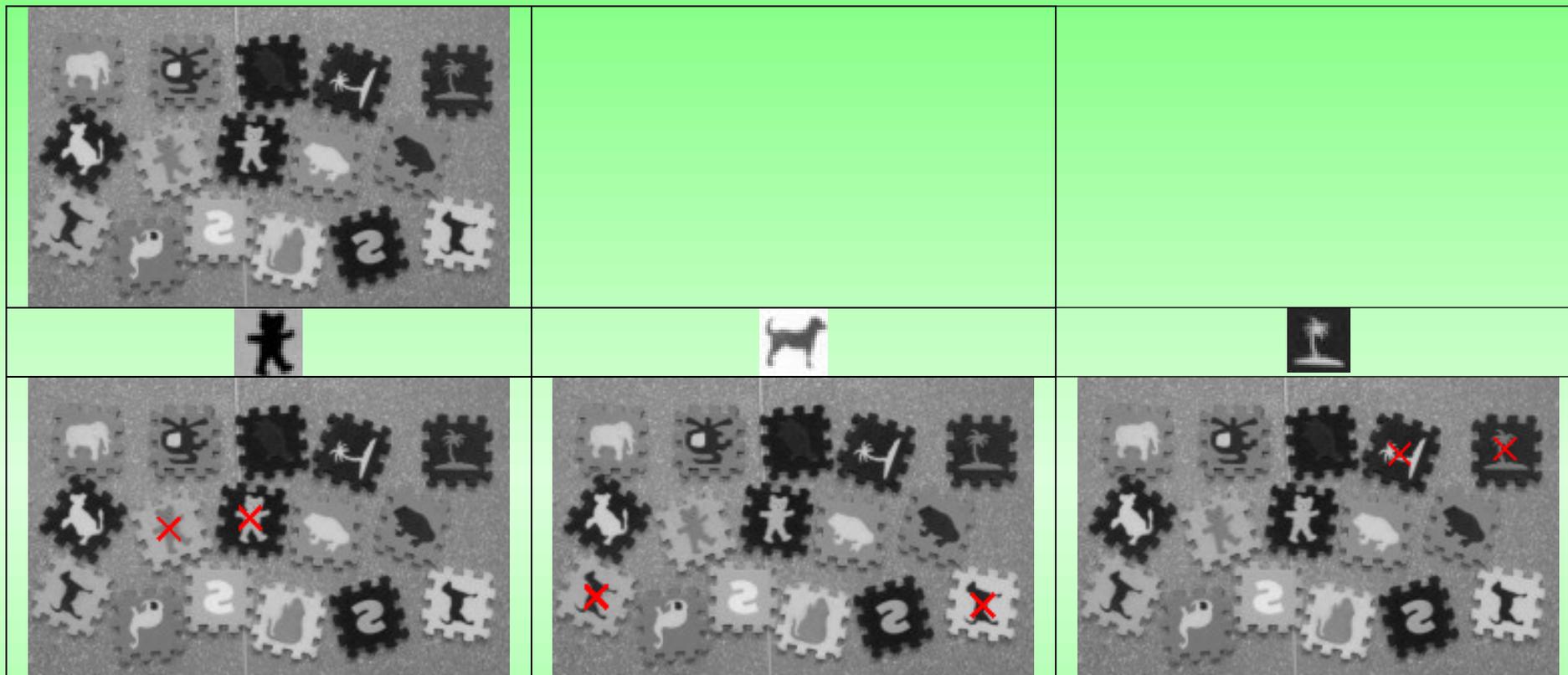
Template matching procura uma máscara (template) dentro de uma imagem.

Normalized cross correlation é um template matching. Retorna valores de -1 a 1, conforme o grau de casamento.



Normalized cross correlation pode ser calculada rapidamente usando FFT e “integral image”.

Normalized cross correlation pode se tornar invariante à rotação repetindo o processo para várias rotações.

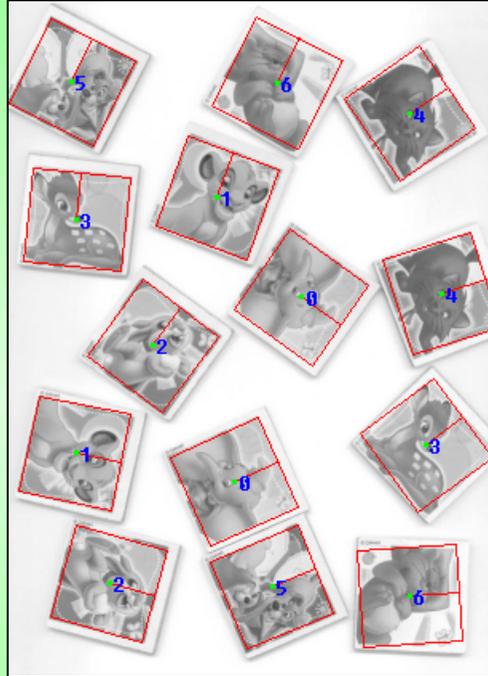


Existem várias técnicas que procuram fazer rapidamente:

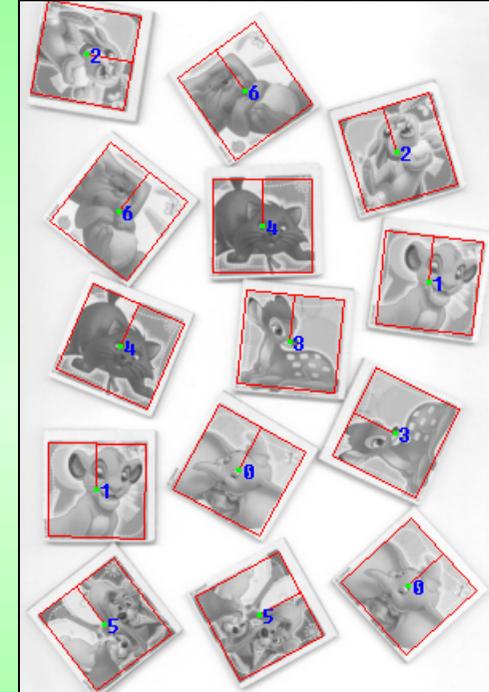
- Template matching invariante à rotação.
- Template matching invariante à rotação e à escala.
- Template matching invariante à rotação, escala e perspectiva.



Símbolo “McDonald’s”



“Jogo de memória” 1



“Jogo de memória” 2

O reconhecimento fica mais fácil se utilizar cores.

4.1 Detecção e leitura da placa de carro

Uma aplicação de “template matching” é na leitura computadorizada das placas de automóveis. Basta treinar 10 dígitos e 26 letras.

Para acelerar, restringir busca somente em regiões com cores semelhantes à placa de automóvel.



4.2 Reconhecimento do modelo de carro

Provavelmente, é possível reconhecer o modelo de carro (Ex: Gol, Palio, Mille, Fiesta, Corsa, etc) procurando algumas regiões que caracterizam cada modelo.

4.3 Contagem de número de eixos em caminhões

Provavelmente, é possível contar o número de eixos em caminhões a partir de sua fotografia lateral. Basta detectar e contar o número de pneus.

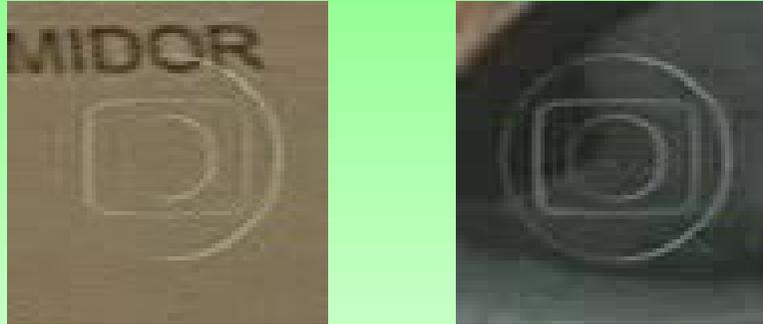
4.4 Detecção de face

OpenCV vem com um programa exemplo que faz detecção de faces.



4.5 Outras aplicações

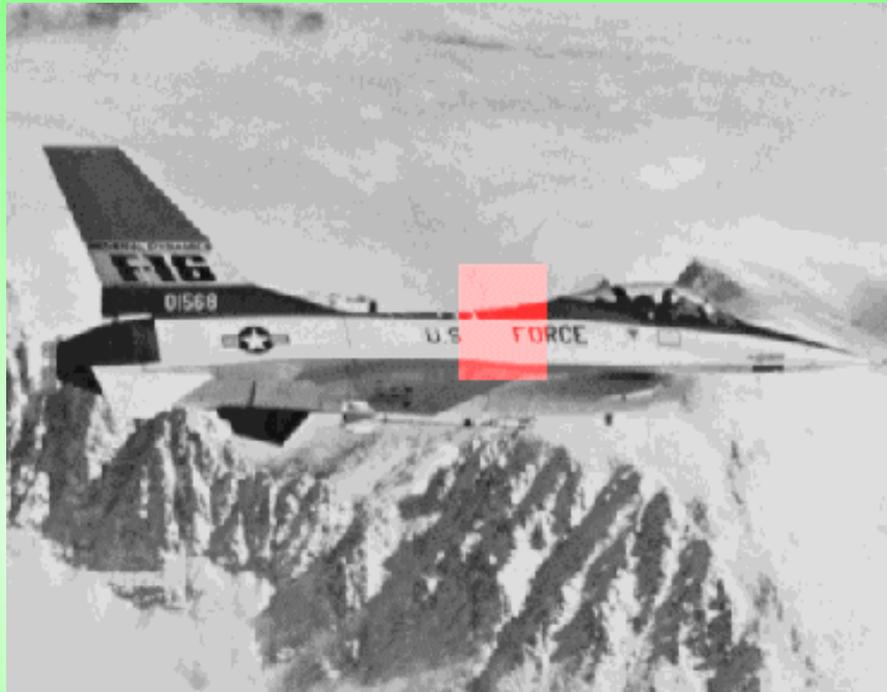
- Reconhecimento do canal sintonizado detectando o logotipo da emissora.



Frames com logotipo

- Reconhecimento do comercial veiculado.

5 Marca d'água de autenticação



Dados invisíveis inseridos na imagem detectam alterações na imagem (note que a palavra "AIR" foi removida).

Pesquisa: Autenticação de imagens binárias (documentos).

Estou desenvolvendo um projeto para autenticar vídeos.

6 Hardware especial

OpenCV traz rotinas para capturar e processar vídeo vindo de webcam.

Há orientandos meus trabalhando com DSP (Blackfin) e FPGA (Altera) para processar imagens e vídeos. Mas não é a minha especialidade.