

VISUAL AUTONOMY

Protótipo para Reconhecimento de
Placas de Trânsito

Fernando Poffo – Acadêmico

Prof. Mauro Marcelo Mattos, Dr. – Orientador, FURB

ROTEIRO

- Introdução
- Fundamentação teórica
- Desenvolvimento
- Conclusão

INTRODUÇÃO

- Atenção do motorista.
- Sinalização de trânsito.
- Sistemas de Apoio ao Motorista.
 - Ativos
 - Passivos

INTRODUÇÃO

- Objetivos do Trabalho
 - Desenvolver uma ferramenta que detecte e reconheça placas de trânsito em uma imagem.
 - Informar a placa detectada visualmente e em tempo real.

FUNDAMENTAÇÃO TEÓRICA

- Sistemas de apoio ao motorista



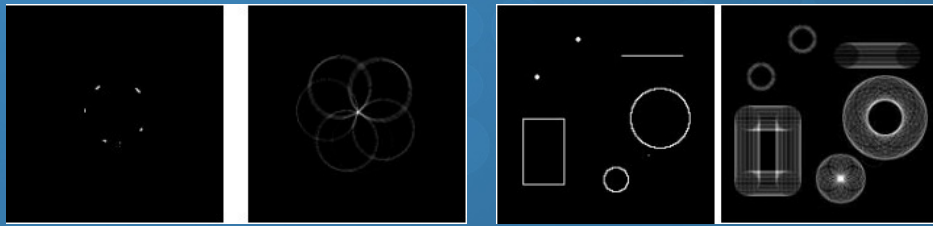
FUNDAMENTAÇÃO TEÓRICA

- Quantização de cores
 - Visão computacional
 - Redução da profundidade da cor



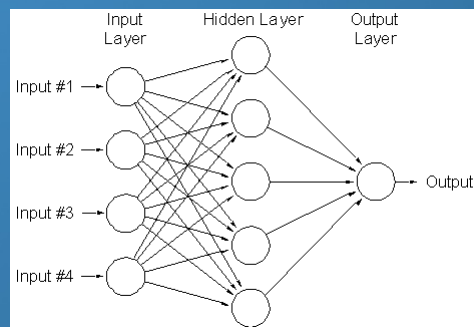
FUNDAMENTAÇÃO TEÓRICA

- Transformada circular de Hough
 - Visão computacional
 - Segmentação



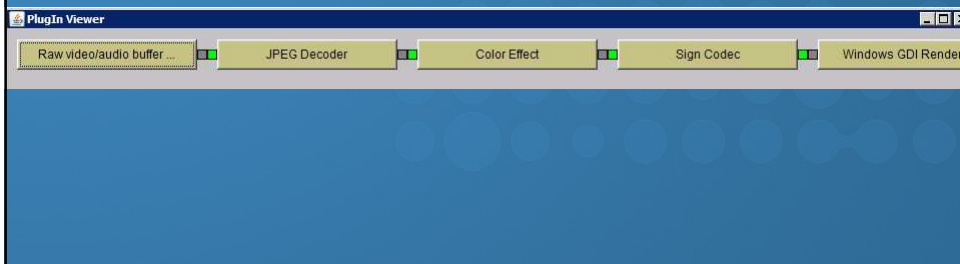
FUNDAMENTAÇÃO TEÓRICA

- Rede neural
 - Multilayer Perceptron
 - Aprendizado Backpropagation



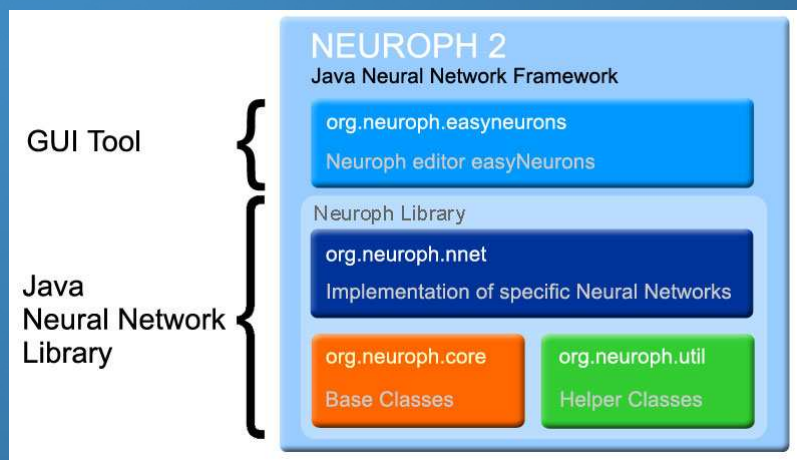
FUNDAMENTAÇÃO TEÓRICA

- JMF
 - Captura
 - Plug-ins
 - Renderização
 - Transmissão



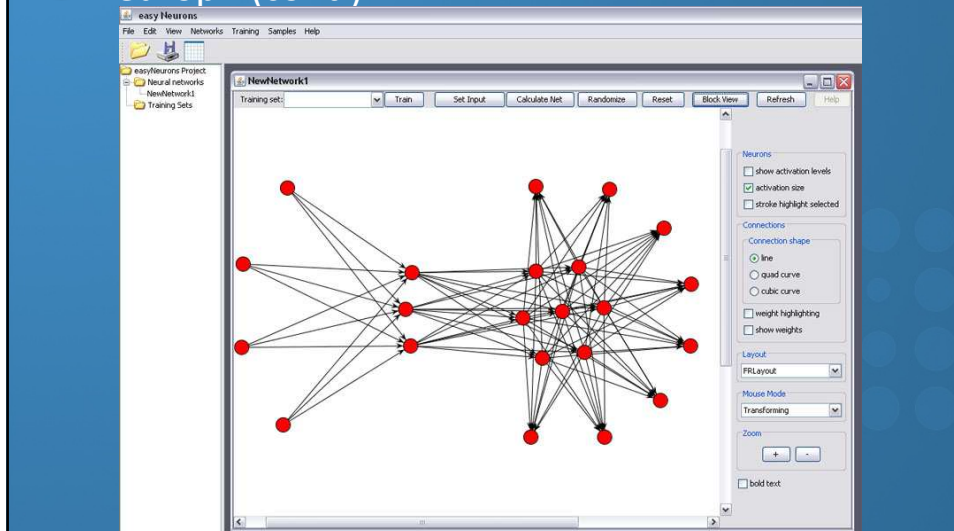
FUNDAMENTAÇÃO TEÓRICA

- Neuroph
 - Redes neurais



FUNDAMENTAÇÃO TEÓRICA

● Neuroph (cont.)



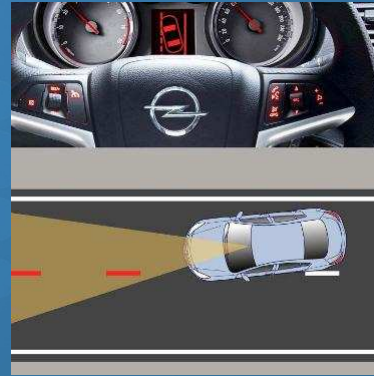
FUNDAMENTAÇÃO TEÓRICA

● Trabalhos Correlatos

- Sistema óptico para identificação de veículos em estradas.
- Opel Eye.
- ARToolkit.

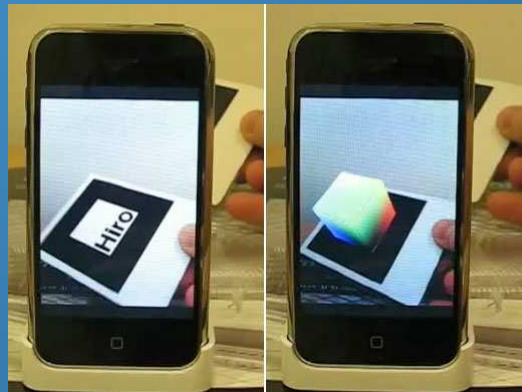
Opel Eye

- Insignia 2008 – GM/Opel



ARToolKit

- Realidade aumentada.



DESENVOLVIMENTO

● Principais Requisitos

- capturar uma imagem contendo um cenário da via à frente do veículo (RF);
- detectar a região da imagem onde cada placa de trânsito aparece (RF);
- reconhecer a placa de trânsito de cada região detectada (RF);

DESENVOLVIMENTO

● Principais Requisitos (cont.)

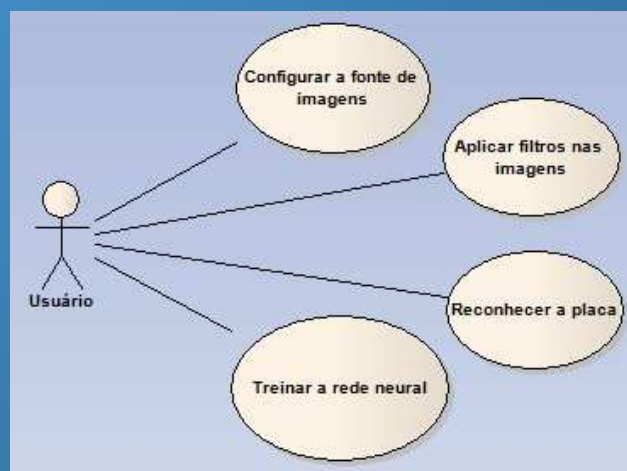
- informar o usuário caso não seja possível reconhecer a placa (RF);
- informar o usuário para cada placa de trânsito reconhecida (RF);
- implementar a ferramenta utilizando a tecnologia Java (RNF).

Especificação

- Técnicas e Ferramentas Utilizadas
 - Enterprise Architect 7.1
 - *Unified Modeling Language (UML)*

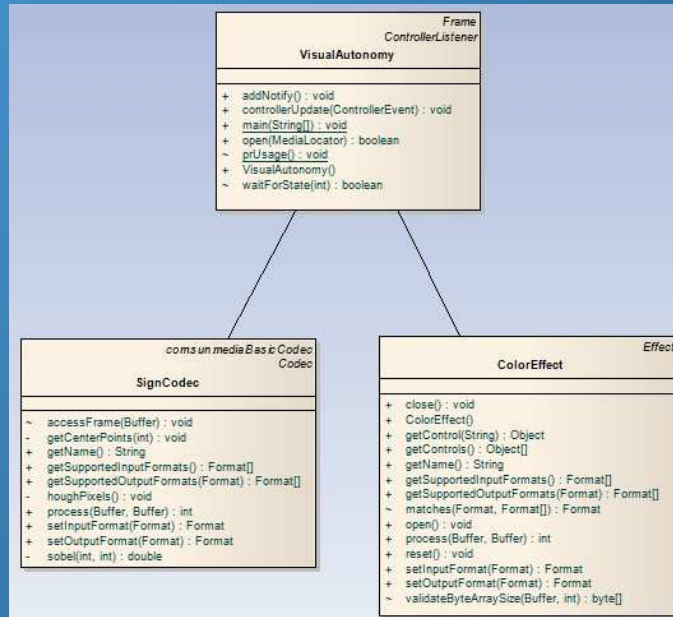
Especificação

- Diagrama de casos de uso



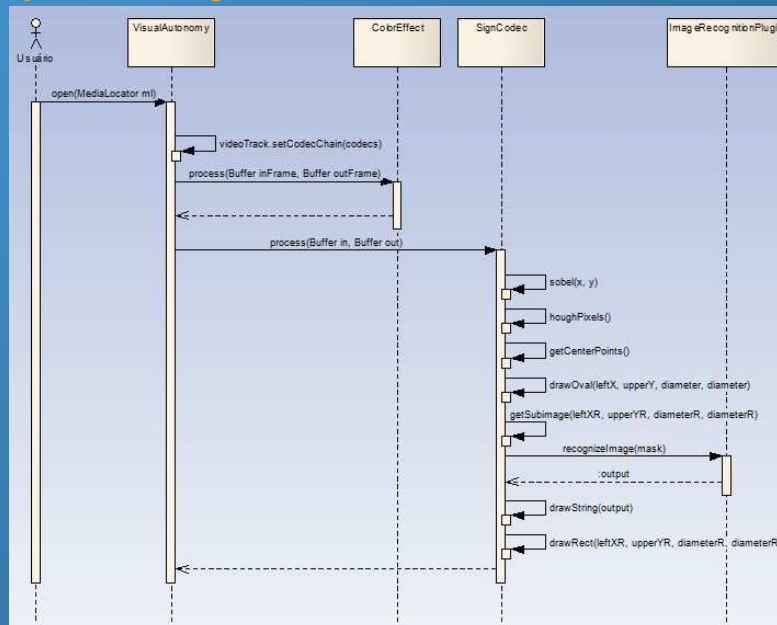
Especificação

Diagrama de classes



Especificação

Diagrama de seqüência



Implementação

● Técnicas e Ferramentas Utilizadas

- Eclipse IDE
- Java
 - JMF
 - Neuroph



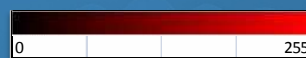
Implementação

● PRÉ AJUSTE

```
byte[] indata = (byte[]) inFrame.getData();  
  
for(int index = 0; index < indata.length; index = index + 3)  
    int blue = indata[(int) (index)];  
    int green = indata[(int) (index+1)];  
    int red = indata[(int) (index+2)];
```



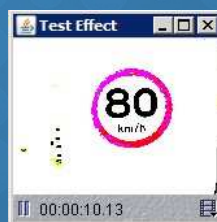
```
if (blue < 0)    blue = blue+255;  
if (green < 0)  green = green+255;  
if (red < 0)    red = red+255;
```



Implementação

● QUANTIZAÇÃO UNIFORME

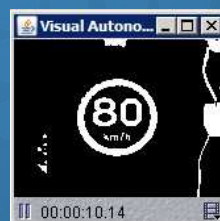
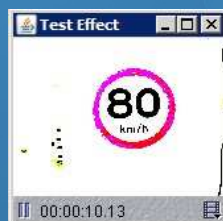
```
int bright = 128;  
if (blue < bright) blue = 0;  
else blue = 255;  
if (green < bright) green = 0;  
else green = 255;  
if (red < bright) red = 0;  
else red = 255;
```



Implementação

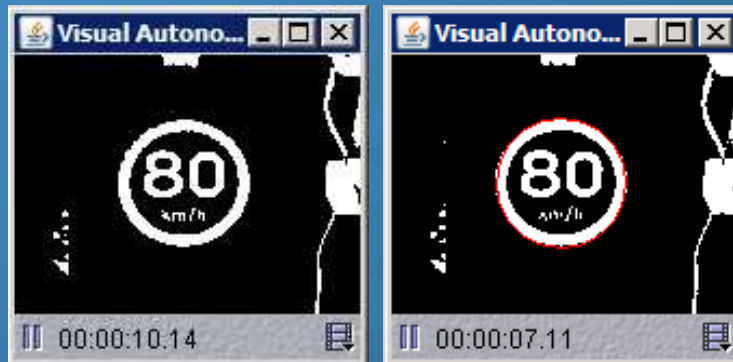
● SELEÇÃO DE CORES

```
if (((green == 0) && (red == 255)) ||  
    ((blue == 0) && (green == 0) && (red == 0))) {  
    blue = 255;  
    green = 255;  
    red = 255;  
}  
else {  
    blue = 0;  
    green = 0;  
    red = 0;  
}
```



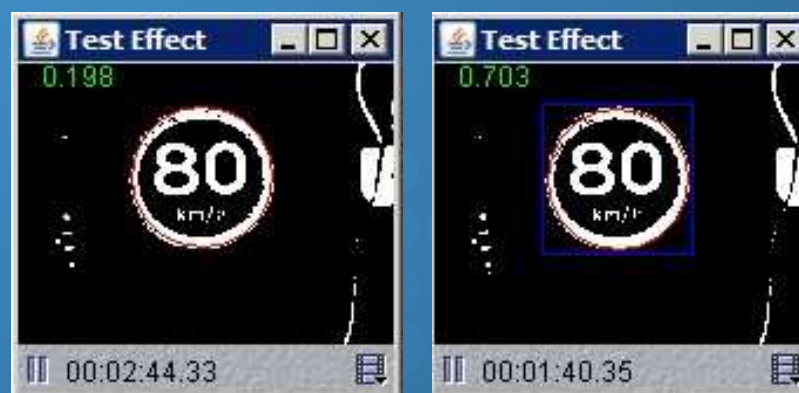
Implementação

- DETECÇÃO DA PLACA
 - Transformada circular de Hough



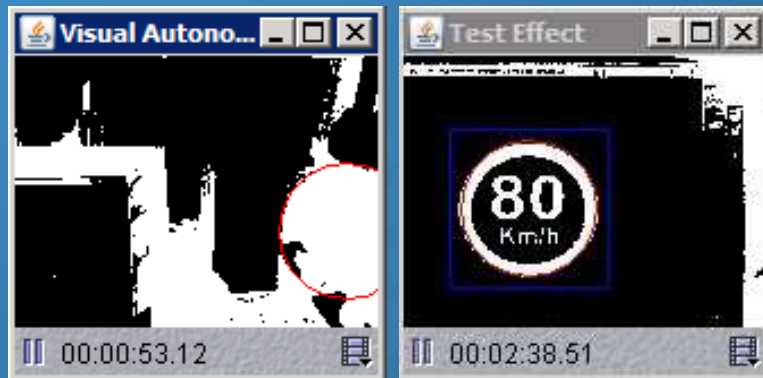
Implementação

- NEUROPH



Implementação

- OPERACIONALIDADE DA IMPLEMENTAÇÃO



Resultados e Discussão

- Controle manual de brilho da imagem.
- Controle manual do diâmetro para detecção da placa.
- Aumento da resolução atrasa o tempo de resposta.
- Dispositivo de captura fornece imagens com muito ruído e função de auto-brilho não pôde ser controlada.
- Limitação ao treinar a rede neural (exemplar único).

Resultados e Discussão (cont.)

- Complexidade do tema.
- Requisitos de tempo de resposta.
- Código segmentação Hough adaptado do trabalho de Schulze.
- Dispositivo de captura, padrão UVC, utilizado: webcam XBOX Live Vision.



CONCLUSÃO

- Resultado de reconhecimento satisfatório, dentro de um ambiente controlado.
- Detecção e reconhecimento ocorrem em tempo real.
- Desenvolvido utilizando apenas Java como linguagem.
- Mercado automobilístico começando a popularizar sistemas com visão computacional.
- Ideal refinar protótipo e explorar mais técnicas.

Extensões

- Controlar o brilho da imagem.
- Utilizar algoritmo que não sofra influência do brilho.
- Explorar outras APIs de manipulação de imagens.
- Desenvolvimento de uma ferramenta de treino de rede neural específica.
- Detectar placa em aproximação (movimento).

Obrigado!



A mente que se abre a uma nova idéia
jamais voltará ao seu tamanho original.
Albert Einstein

Blumenau, 05 de julho de 2010.