



Sistemas Sensoriais

2015/2016 – 1º Semestre

Sistema de detecção de sinais de trânsito

Relatório

Grupo formado por:

- João Barata Nº 31559
- Miguel Prego Nº 34008



Índice

1 - Introdução.....	3
2 - Interface.....	4
3 - Algoritmos.....	5
3.1 - Detecção de Vermelhos	5
3.2 - Limites	6
3.3 - Recorte e Redimensionamento.....	7
3.4 - Limpeza e binarização da imagem	8
3.5 - Comparação com a base de dados.....	9
4 - Conclusões.....	10



1 - Introdução

Para este trabalho pretende-se implementar um sistema de detecção de sinais de trânsito utilizando técnicas de processamento de imagem, tal como é feito nos presentes dias e se encontra implementado nos automóveis mais recentes.

A aquisição da imagem contendo o sinal de trânsito a avaliar será feita através de câmaras instaladas no interior do automóvel em que o sistema seja aplicado. Embora sejam disponibilizados alguns exemplos pelos docentes, foi recomendável que os alunos testassem os seus trabalhos com outros exemplos para garantir que a aplicação desenvolvida é suficientemente robusta.

Após o programa correr o código implementado pelo grupo, é possível visualizar as várias etapas de processamento de imagem, através de uma nova janela, bem como o sinal de trânsito mais parecido com aqueles existentes na base de dados.

Os requisitos da aplicação são os seguintes:

- O sistema irá analisar apenas imagens estáticas, em vez de sequências de vídeo como seria num cenário real;
- A identificação dos sinais deverá ser feita por análise de semelhança com uma lista de sinais de trânsito previamente escolhida;
- Para limitar a complexidade dos algoritmos a desenvolver, apenas se pretendem detectar sinais que contenham cor vermelha (perigo e proibição);
- A lista deverá ter um mínimo de 6 sinais de cada tipo (perigo e proibição);
- A dimensão/resolução dos sinais poderá variar;
- Deverá tolerar alguma variação de iluminação entre imagens;
- Existirá apenas um sinal de trânsito em cada imagem;

Para além disto o trabalho poderá apresentar as seguintes opções (facultativas, não implementadas pelo grupo por falta de tempo, devido a horários das diversas avaliações do mestrado):

- Processamento de vídeo (sequências de imagens)
- Permitir vários sinais de trânsito numa imagem;
- Detectar outros sinais de trânsito (obrigação, informativos, etc.);



2 - Interface

A interface do nosso grupo é bastante simplista, aproveitando ao máximo a já existente, feita pelos docentes da cadeira. Além disso o facto de não termos tido tanto tempo como o desejado para a realização deste trabalho, fez com que as funcionalidades extra não fossem programadas, pelo que a interface não disponibiliza muitas opções.

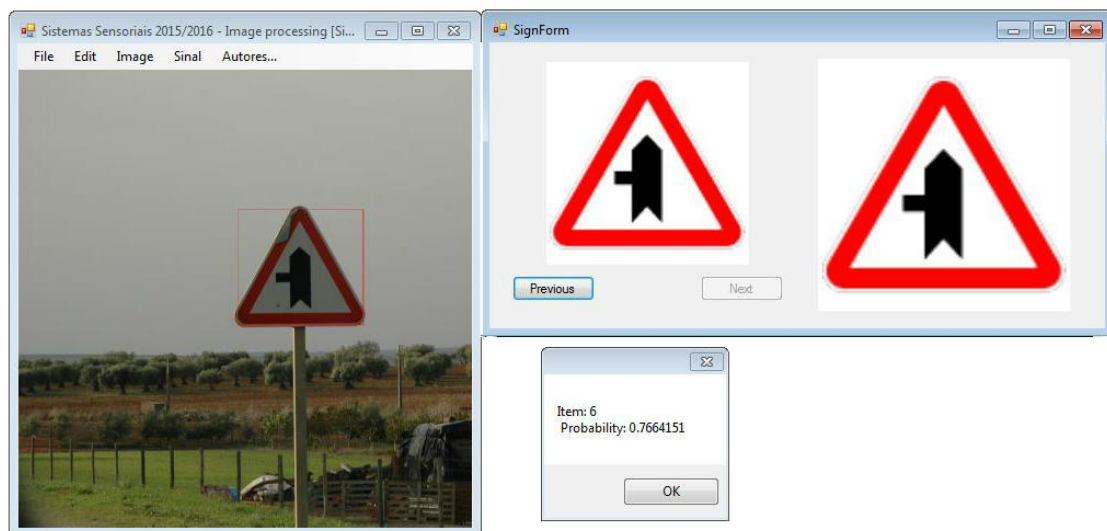


Figura 1 - Interface

Na janela principal do programa é mostrada a imagem em análise, com um rectângulo em volta do sinal detectado, de modo a mostrar ao utilizador a detecção. Ainda nesta janela estão disponibilizados todos os tipos de processamento de imagem realizados pelo grupo ao longo da duração da cadeira, na opção “Image”.

Na janela “SignForm”, do lado direito, o utilizador pode visualizar o resultado proposto pelo programa, após analisar todas as imagens da base de dados enquanto do lado esquerdo são mostradas todos os processos aplicados à imagem, que se encontram numa lista de imagens que é passível de ser percorrida com o clique nos botões “Previous” e “Next”.

É também exibida uma pequena janela que contém a informação sobre o número da imagem reconhecida na base de dados e a probabilidade com que essa imagem foi escolhida.



3 - Algoritmos

Neste capítulo do relatório apresentamos a explicação dos algoritmos utilizados neste trabalho para responder aos requisitos propostos no trabalho, bem como as justificações com imagens retiradas do programa, durante a sua execução.

3.1 - Detecção de Vermelhos

Para esta detecção, é efectuado a diferença do valor da componente vermelha e a média das componentes azul e verde bem como verificada a diferença entre as componentes azul e verde de cada um dos pixéis da imagem. No caso do primeiro cálculo, é verificado se esse valor é maior que 30 (mostrando que a componente vermelha é mais forte que a média das outras duas) e no segundo é confirmado que o valor resultante é menor que 20 (significando que as componentes azul e verde se encontram equilibradas. Cada um dos pixéis detectados é transformado em branco ou preto, tonando assim a imagem original numa imagem binarizada.

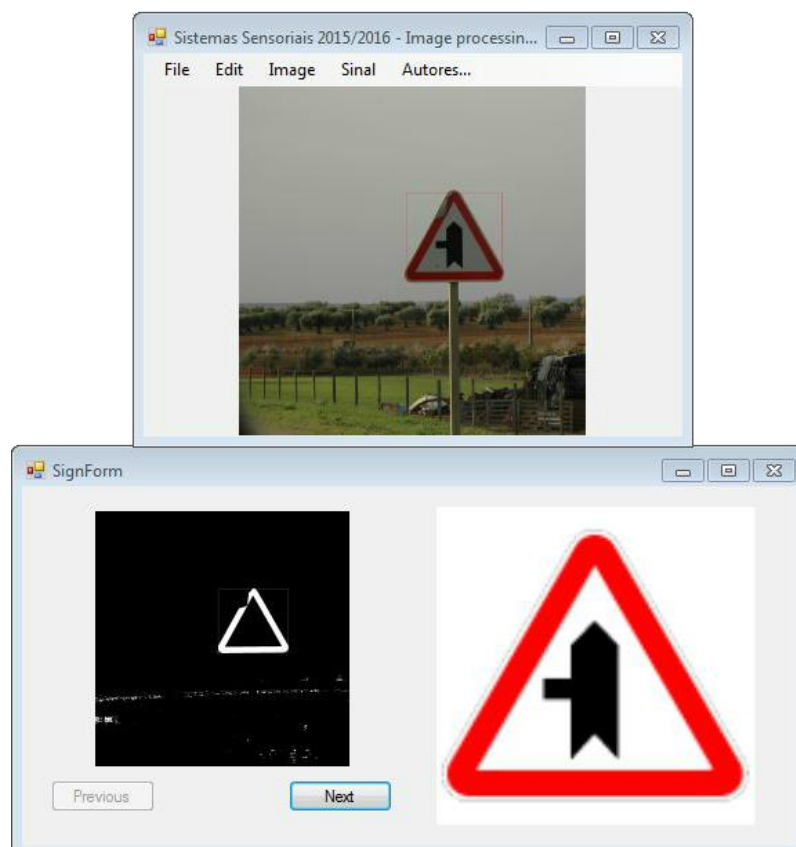


Figura 2 - Detecção de vermelhos



3.2 - Limites

No processo seguinte, são feitas duas projecções, uma para cada eixo, de forma a detectar os limites dos sinais. O facto de não termos escolhido converter a imagem RGB para HSV ou outro qualquer sistema de cores mostrou-se um desafio nesta etapa, visto que na etapa anterior tinham sido detectados alguns pixéis em tons de castanho, (cor muito comum nas paisagem circundantes aos sinais).

O grupo ainda ponderou efectuar uma erosão na imagem obtida no passo 1 mas esta não se mostrou muito eficaz em algumas imagens e portanto a decisão tomada foi a de ignorar todas as zonas em que a contagem de pixéis brancos fosse menor do que 15, eliminando desta forma algum do ruído existente, facilitando a detecção dos limites do sinal. Esta implementação não é a mais acertada, visto que as imagens não têm todas a mesma resolução e seria facilmente corrigida, mudando o valor 15 para uma percentagem do total de pixéis da imagem.

Além disso, existe uma verificação para que seja detectado o sinal na maior zona em que existam contagens superiores a 15. Desta forma, mesmo que existam regiões de ruído acentuado, são ignoradas, supondo que o sinal ocupa mais espaço na imagem do que o ruído existente na mesma. Após ser feita a detecção do sinal, é feito um rectângulo em volta do sinal de modo a mostrar de forma a sobressair mais a sua localização

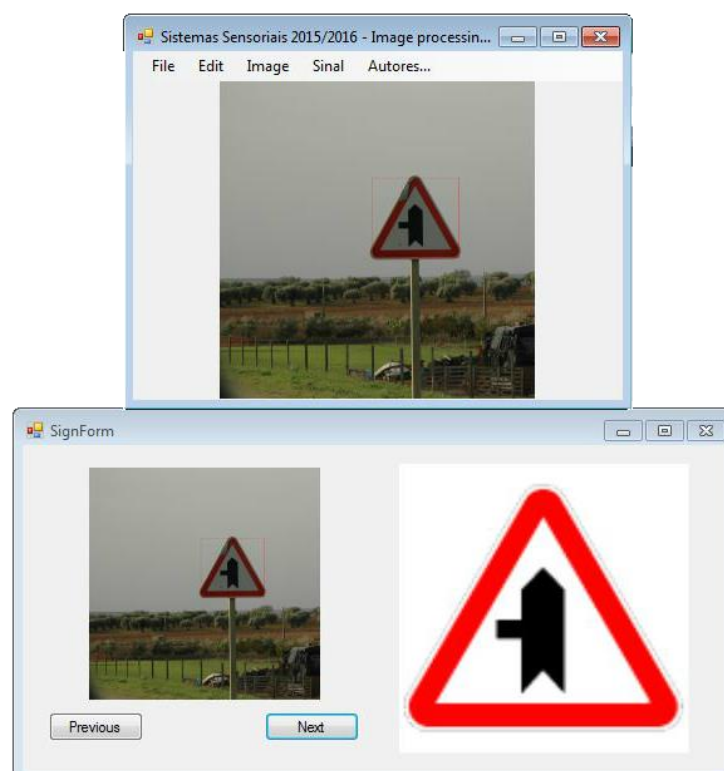


Figura 3 - Selecção do sinal



3.3 - Recorte e Redimensionamento

Após já conhecer bem os limites do sinal a analisar, a imagem é copiada a partir da original (não binarizada) para uma nova imagem com as dimensões de 111 X 111 pixéis, de modo a igualar o tamanho das imagens existentes na nossa base de dados. A cópia da imagem é feita através de um método semelhante ao da rotação inversa, onde são percorridas as posições da imagem de destino em vez das posições da imagem de origem. Embora no caso da translação não exista o mesmo problema que na rotação, optámos por fazer este tipo de translação pois os ciclos que percorrem a imagem são mais simples de implementar.

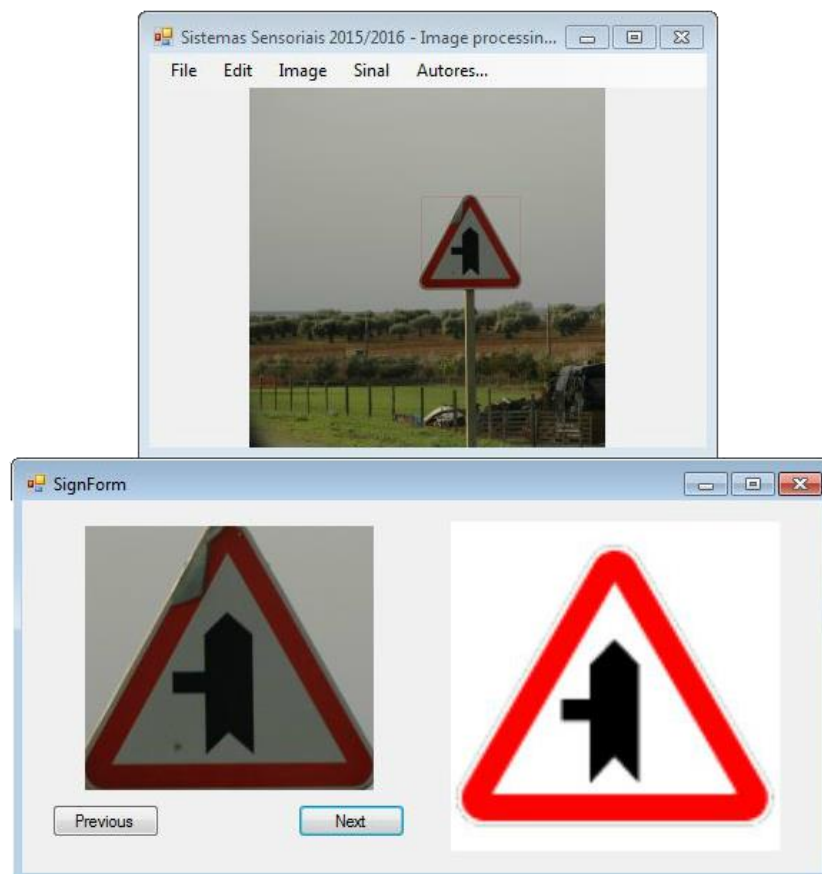


Figura 4 - Redimensionamento da imagem



3.4 - Limpeza e binarização da imagem

Nesta etapa é feita uma limpeza mais detalhada da imagem de forma a facilitar a comparação com as imagens da base de dados e ao mesmo tempo o reconhecimento do sinal correcto torna-se mais eficaz. Primeiro, a imagem é percorrida de cima para baixo, da esquerda para a direita até encontrar um pixel dentro do nível de vermelho considerado válido (explicado no ponto 2.1). Todos os pixéis pelos quais o ciclo passou são pintados de branco. Depois, a imagem resultante é analisada de cima para baixo, da direita para a esquerda até encontrar novamente um pixel vermelho e tal como anteriormente, os pixéis percorridos ficam brancos. Estes dois ciclos fazem com que o exterior da imagem fique totalmente branco, aproximando-a com as existentes na base de dados.

Após esta primeira limpeza do exterior, é feita uma binarização semi-manual, visto que o método otsu se mostrou pouco eficaz durante os testes efectuados. Deste modo, a imagem é percorrida uma última vez, de cima para baixo, da esquerda para a direita, em que todos os pixéis que sejam vermelhos ou pretos são transformados em preto e os restantes ficam em branco (neste caso, os do interior do sinal). No final de todas estas passagens, é obtida uma imagem binarizada, com quase nenhum ruído.

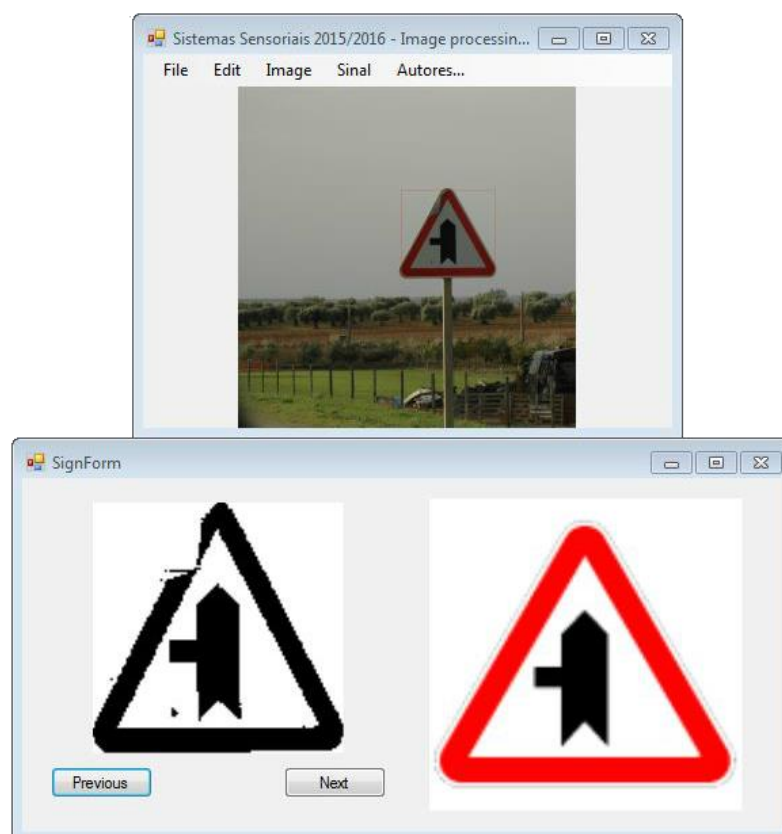


Figura 5 - Imagem binarizada



3.5 - Comparação com a base de dados

Para finalizar a análise da imagem resultante, é efectuada uma comparação com todas as imagens da base de dados, de modo a encontrar a mais parecida. Visto que a finalidade de um programa deste tipo é detectar um sinal de trânsito no menor tempo possível, o grupo optou por fazer as comparações em tempo real, através da utilização de *threads* concorrentes. Cada uma destas *threads* está ligada a uma imagem da base de dados e a comparação entre imagens é feita de forma concorrente.

Quando a comparação entre a imagem em análise e as imagens da base de dados é concluída, a imagem com mais pixéis em comum é seleccionada de entre as possíveis. O resultado final é mostrado na janela do programa, bem como é mostrada a probabilidade e o número da imagem encontrada, numa nova janela.

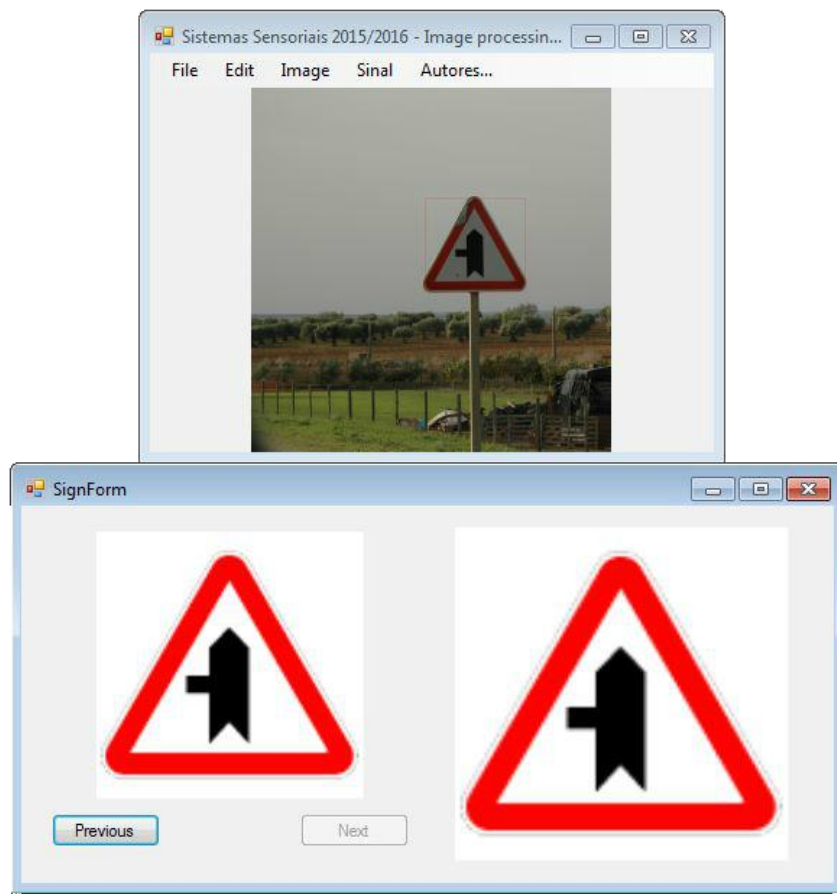


Figura 6 - Resultado da detecção de sinal de trânsito



4 - Conclusões

Concluimos o projecto com sucesso, satisfazendo todas as expectativas, isto é, conseguimos implementar o reconhecimento automático de sinais de trânsito existentes nas vias de circulação, com base em processamento de imagem. Todas as exigências fundamentais foram cumpridas.

Tiramos conclusões muito positivas deste projecto final, desenvolvemos competências importantes a nível de processamento de imagem e adquirimos novos conhecimentos a nível teórico e prático. A nível prático gostaríamos de destacar a resolução de problemas relacionados com processamento de imagem e as competências adquiridas e desenvolvidas ao trabalhar directamente com a linguagem C# bem como a sua utilização directa na biblioteca “OpenCV”.

Os trabalhos realizados durante o decorrer do semestre nas aulas práticas mostraram-se muito úteis na resolução do problema proposto, pela possibilidade de aplicação directa nos algoritmos desenvolvidos e pelo raciocínio necessário fazer para encontrar soluções ainda não implementadas.

Tivemos algumas dificuldades ao longo do projecto, nomeadamente na definição de limite nas regiões de interesse. Devido às diferentes resoluções que as imagens poderiam ter, tornou-se mais difícil determinar valores que seriam óptimos para todos os casos, pelo que temos total consciência que o código desenvolvido não é totalmente eficaz e deixa espaço para melhorias futuras, embora tenhamos gasto bastante tempo em testes exaustivos, em vários exemplos de imagens, de modo a aumentar a eficácia. De qualquer modo, sentimos que o produto final ficou bastante satisfatório, embora não esteja pronto para ser comercializado mas, com algum tempo e dedicação algumas falhas poderiam ser corrigidas.

Em suma, resta-nos fazer uma avaliação global do projecto. Sentimo-nos bastante satisfeitos com o resultado final apresentado. Desde o início existiu um grande interesse por este trabalho e no problema apresentado no mesmo. Fazemos um balanço bastante positivo de toda a experiência e gostaríamos de melhorar o produto final apresentado.