

Reconhecimento de Cartas

Álvaro Martins 72447 & André Rodrigues 73152

Resumo - O presente artigo apresenta Reconhecimento de Cartas com a utilização do Template Matching, tal como a sua análise e implementação.

Usando uma abordagem que se pretende pedagógica, o artigo começa por estabelecer o problema Reconhecimento de Cartas com a utilização do Template Matching de um ponto de vista geral e teórico.

De seguida, a apresentação das várias técnicas é abordada detalhadamente relativa a cada passo necessário para o reconhecimento. Finalmente é abordada como é feito o matching, será feita uma apresentação do dataset de match e os resultados obtidos com os diferentes métodos de match e a sua comparação.

O artigo está dividido em apresentação do método de leitura da imagem, reconhecimento da carta, análise da carta, transformações aplicadas na carta, métodos de matching, dataset dos templates, resultados e a sua discussão, terminando com uma breve conclusão.

Com a execução deste trabalho pretendemos, essencialmente, alargar os nossos conhecimentos no que respeita à manipulação de imagens, bem como o seu reconhecimento e uso de algoritmos na sua transformação.

I. Introdução

No âmbito da Unidade Curricular Computação Visual do 4º ano do curso Mestrado Integrado em Engenharia de Computadores e Telemática escolhemos a realização de um projecto em OpenCV no qual desenvolvemos uma aplicação de reconhecimento de cartas utilizando Template Matching. Optamos por este tema pois o conceito reconhecimento e matching é de interesse para ambos os elementos do grupo.

OpenCV[1] (Open Source Computer Vision Library) foi desenvolvida pela Intel em 2000, é uma biblioteca multiplataforma nas linguagens C/C++ e com suporte a Python, Java e Visual Basic. É gratuita ao uso académico e comercial e é utilizado para o desenvolvimento de aplicações na área de Computação Visual.

O Template Matching[2] é uma técnica de processamento de imagem para encontrar pequenas partes de uma imagem que correspondem a uma imagem de modelo(template). Pode ser utilizado no controle de qualidade na fabricação de materiais, como uma forma de detectar bordas em imagens e em inteligência artificial em redes neuronais uma vez que uma característica principal no método de reconhecimento é a rotação e o “scaling” da imagem. Neste caso será utilizado para o reconhecimento de cartas e poderá futuramente ser utilizado num casino por exemplo ou no desenvolvimento de um agente inteligente capaz de jogar um jogo, como o BlackJack, abordado na cadeira de Inteligência Artificial, onde o agente seria capaz de jogar contra humanos em tempo real.

II. Metodologia

Leitura

Foi utilizada uma webcam para a detecção de cartas. No processo de leitura, a webcam captura uma imagem (alguns frames) e posteriormente será feito o match com a biblioteca de todos os matches possíveis.

A resolução da câmara é importante por isso a utilização de uma webcam com melhor resolução fornecerá melhores resultados.

Para o devido funcionamento do projecto é fundamental montar a webcam relativamente perto das cartas e com um ajuste de iluminação favorável uma vez que estes dois fatores podem prejudicar o template matching(Fig.1).



Figura 1 - Leitura da carta numa superfície preta

Determinar os Templates

Na implementação do algoritmo de reconhecimento de uma carta, é necessário anotar em primeiro lugar as características que as distinguem das outras cartas e que as categorizam, para tal é necessário ter em consideração o naipe e o seu valor alfanumérico.

A estratégia de matching utilizado neste projecto foi realizar 2 matches para o devido reconhecimento onde o primeiro match identifica qual é o naipe correspondente e o segundo match identifica o seu valor. Os templates (Fig.2 e Fig.3) serão guardados num dataset com o nome do seu valor ou o nome do naipe, para tal existem 4 templates equivalente a cada naipe, e 13 templates que correspondem ao valor de cada carta, desde o Ás até ao Rei.

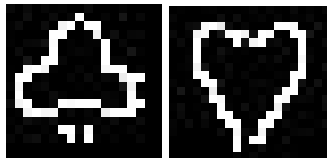


Figura 2 - Template dos naipes Copas e Paus (contornos)

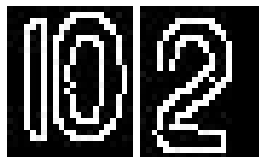


Figura 3 - Template de alguns valores, 10 e 2 (contornos)

Carta

Na leitura da carta, é lida em RGB, convertida para HSV e de seguida é feito um threshold da imagem a preto e branco de forma a remover o ruído existente. Na seleção da carta na imagem é feita a detecção dos contornos (Fig.4) da carta com atenção que os cantos da carta são curvos, para tal foi usada a função `isContourConvex` numa pesquisa em árvore.



Figura 4 - Detecção dos cantos da carta

No caso da carta não se enquadrar num rectângulo perfeito (Fig.5) é feita uma rotação e scale da imagem de modo a enquadrá-la num retângulo quase perfeito. A imagem final apenas contém os contornos da carta. (Fig.6)



Figur 5 - Detecção dos cantos da carta “desenquadrada”

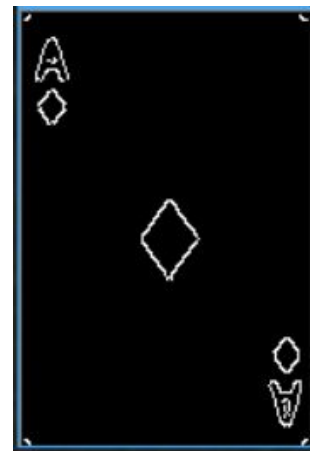


Figura 6 - Imagem final da carta após rotação e crop/corte

Template Matching

Na execução do algoritmo de reconhecimento, foi utilizado o método de match `SQDIFF NORMED` que faz a diferença dos quadrados em cada pixel e a normalização.

Para preparar um match é necessário o dataset dos templates e uma imagem cropped da carta que contém os contornos dos naipes e do seu valor. A cada template é feito um pré-processamento com o mesmo algoritmo aplicado à cropped card, ou seja, cada template apenas possui o contorno de um naipe ou valor, o match em si será a comparação dos contornos da carta com os contornos dos templates. (Fig.7)

Na deteção de um match, após a escolha do método adequado de matching e alguns testes, foi notado que quando o `matchLoc(x,y)` encontra-se no ponto (0,0), o match não foi encontrado.



Figura 7 - Mensagem de deteção do Match.

III. Testes e Resultados

Foram feitos dois tipos de testes, o primeiro teste foi realizado num ambiente ideal, com iluminação mediana, as cartas colocadas sempre na mesma posição e numa superfície plana com fundo preto de modo a facilitar a deteção dos contornos.

No segundo teste, a câmara não estava fixa, podendo assim movimentá-la e auxiliar no match, em algumas situações apenas uma ligeira movimentação da câmara era suficiente para o correcto reconhecimento e adicionalmente testamos com outro baralho de cartas.

Cada coluna da tabela a seguir representada indica qual a carta utilizada no teste, se a sua deteção do Naípe foi fácil(cor verde), mediana(cor amarela) e difícil(cor vermelha).

Resultados 1º Teste

Carta	Deteção do Naípe	Deteção do Valor
Às		
2		
3		
4		
5		
6		
7		
8		
9		
Figuras(J,Q,K)		

Tabela1- Resultados do 1º teste, ambiente ideal

Resultados 2º Teste

Carta	Deteção do Naípe	Deteção do Valor
Às		
2		
3		
4		
5		
6		
7		
8		
9		
Figuras(J,Q,K)		

Tabela1- Resultados do 2º teste, ambiente 'não-ideal'

Discussão dos Resultados

Foi notado que no 1º teste não houve quase nenhuma dificuldade, existe uma pequena dificuldade em detectar o naipe no Ás pois este contém unicamente uma imagem do seu respectivo naipe.

No 2º teste as cartas com menor quantidade de símbolos foram as que tiveram mais dificuldades no seu devido reconhecimento, supomos que uma carta com mais símbolos seja mais fácil uma vez que há mais opções de match correcto enquanto que por exemplo um Ás contém menor número de símbolos do seu naipe do que um 9 por exemplo.

Nas cartas que tivemos dificuldade, um ajuste da câmara e aproximação foi o suficiente para reconhecer a carta, em nenhum dos casos anotados a carta não foi reconhecida.

Se alterarmos de câmara para uma com melhor resolução melhores resultados são obtidos, a luminosidade influencia muito o resultado final, a utilização do outro baralho implica uma ligeira mudança no dataset dos templates, no entanto, uma grande parte das cartas do novo baralho são reconhecidas.

IV. CONCLUSÃO

Neste trabalho foi abordado o tema Reconhecimento de Cartas com o uso de Template Matching, referimos inicialmente a plataforma utilizada e uma breve definição de template matching e a sua utilização, foi feita a devida descrição da metodologia aplicada no desenvolvimento do projecto, leitura da carta, transformações necessárias para que a carta ficasse sempre na mesma posição após o crop pelos contornos, na distinção das cartas abordamos as características valor e naipe, com estas características foi construído o dataset de templates, transformações na imagem cropped e no template de modo a obter só os contornos dos símbolos presentes na carta (valor e naipe) e de finalmente o match dos contornos.

O matching após os testes, consideramos que foi favorável, o grupo estava a espera de piores resultados, no entanto como era de esperar, num ambiente perfeito o reconhecimento é quase perfeito e num ambiente não-ideal é encontrado algumas dificuldades no reconhecimento das cartas influenciado pelo fator luminosidade e posição da câmara.

Cumprimos todos os objetivos que nos foram propostos neste projeto em que o principal era o reconhecimento de cartas.

Para um aumento da performance do sistema o grupo aconselha a utilização de uma câmara com melhor resolução de imagem e considerar fazer testes num ambiente com pouca iluminação e com um ângulo de 90º

com a superfície de modo a que a câmara fique de frente para a carta e facilite no crop e resize da imagem.

Este trabalho foi muito importante para o nosso conhecimento, o aprofundamento deste tema fez-nos aprender e compreender o método utilizado no reconhecimento de templates assim como melhoramos a nossa técnica e utilização do OpenCV e C++.

V. REFERÊNCIAS

- [1] "OpenCV". Em "Wikipedia": a enciclopédia livre. Disponível em: <<https://pt.wikipedia.org/wiki/OpenCV>> Acesso em Janeiro 2017
- [2] "Template Matching". Em "Wikipedia": a enciclopédia livre. Disponível em: <https://en.wikipedia.org/wiki/Template_matching> Acesso em Janeiro 2017
- [3] "Theory". Em "Wikipedia": a enciclopédia livre. Disponível em: <http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html> Acesso em Janeiro 2017
- [4] "Code". Em "Wikipedia": a enciclopédia livre. Disponível em: <http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html> Acesso em Janeiro 2017
- [5] "Rotate Image". Em: "John Fremlin's blog". Disponível em: <<http://john.freml.in/opencv-rotation>> Acesso em: 5 Janeiro 2017
- [6] "How to crop an image on OpenCV". Em: "OpenCV Doc". Disponível em: <http://docs.opencv.org/trunk/d4/dc6/tutorial_py_template_matching.html> Acesso em: 5 Janeiro 2017
- [7] "C++ Tutorial". Em: "Cprogramming.". Disponível em: <<http://www.cprogramming.com/tutorial/c++-tutorial.html>> Acesso em: 5 Janeiro 2017
- [8] "Read files in sequence from a directory". Em: "Stackoverflow".

Disponível em:

<<http://stackoverflow.com/questions/26535662/how-to-read-files-in-sequence-from-a-directory-in-opencv> >

Acesso em: 5 Janeiro 2017

[9] “Canny edge detection”. Em: “OpenCV Doc”.

Disponível em:

<http://docs.opencv.org/trunk/da/d22/tutorial_py_canny.html >

Acesso em: 5 Janeiro 2017

[10] “Thresholding”. Em “Wikipedia”: a enciclopédia livre. Disponível em:

<[https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing)) >

Acesso em Janeiro 2017