



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Machine Vision in Casino Game Monitoring

João Paulo Maurício Pimentel

Dissertação para a Obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Júri

Presidente:	Prof. Carlos Silvestre
Orientador:	Prof. Alexandre Bernardino
Vogal:	Prof. Margarida Silveira
Acompanhante Externo*:	Eng. Bernardo Motta

Outubro de 2010

*Trabalho realizado em parceria com a Observit - Tecnologias de Visão por Computador, Lda.

Agradecimentos

Primeiro, queria agradecer ao meu orientador Prof. Alexandre Bernardino, por todo o apoio, disponibilidade e por todas as oportunidades que me trouxe. Agradeço também a toda a equipa da Observit, em especial aos Engenheiros Bernardo Motta e Ricardo Marranita, por me acolherem na formação que tanto me ensinou sobre as ferramentas necessárias para este trabalho, e a toda a equipa do VisLab, sem dúvida uma equipa sempre pronta a ajudar quando é preciso.

Não directamente relacionado com este trabalho, mas com todo o percurso até aqui, tenho que agradecer ao Prof. Dr. Luís Borges de Almeida, por me ter introduzido desde muito cedo à temática do Processamento de Imagem, e se assim não tivesse sido, provavelmente não teria escolhido esta área de especialização. Tenho que agradecer também a todos os meus colegas, com quem pude sempre contar. Ainda que sejam muitos, tenho de agradecer em especial ao Miguel Simões e ao André Meira.

E como não podia deixar de ser, tenho que agradecer a toda a minha família e à minha namorada Inês, por todo o seu apoio, incentivo e presença.

Resumo

Para prevenir perdas em jogo por enganos ou estratégia irregular, ou para aquisição de dados estatísticos, a indústria de jogos de casino está interessada em monitorização automática dos jogos, registando o seu estado para análise posterior.

É proposto um método de detecção e reconhecimento de cartas de jogar tendo em conta oclusão parcial e rotação. Este método está dividido em duas fases: uma cena é analisada para detectar rectângulos por um método original, e cada rectângulo é classificado de acordo com a figura presente no canto da carta. É assumido que as dimensões e escala das cartas são conhecidas e que a imagem a analisar não tem deformações de perspectiva nem ópticas, ainda que uma fase de calibração para ter isto em conta tenha sido desenvolvida e testada.

Três métodos para a fase de reconhecimento das cartas são estudados e comparados, avaliando o seu desempenho em termos de robustez a alterações de luminosidade e contraste e em tempo computacional. Foi mostrado que um método de *template matching* funciona com sucesso e eficientemente, ainda que mostre uma baixa robustez a alterações luminosas, enquanto que um método probabilístico rígido baseado em contornos mostrou robustez a estas alterações com um custo de maior tempo computacional. Um modelo probabilístico rígido apresentou pouco sucesso e um tempo computacional muito longo.

Palavras chave: Reconhecimento de Cartas de Jogar, Detecção de Rectângulos, Reconhecimento de Objectos, Template Matching, Modelo Probabilístico Baseado em Edges.

Abstract

To prevent game losses due to mistakes or irregular strategies, or for statistical data acquisition, the casino game industry is interested in automatic monitoring of the games, recording their state for posterior analysis.

A method for detection and recognition of playing cards taking into account partial occlusion and rotation is proposed. This is carried out in two phases: a scene is analyzed to detect rectangles with an original method, fitting the card size, and then each detected rectangle is classified according to the figure present on the corner of the card. The cards are all assumed of a known size and scale, and the image to be analyzed has no perspective or optical deformations, although a calibration phase to take this into account has been developed and tested.

Three methods for the card recognition stage are studied and compared, evaluating their performance in terms of robustness to brightness and contrast changes and computation time. A template matching method was shown to work successfully and time efficiently, although showing low robustness to lighting changes, whereas an edge based probabilistic rigid model showed robustness to such changes at the expense of a longer computation time. A probabilistic deformable model presented low success and a very long computation time.

Keywords: Playing Card Recognition, Rectangle Detection, Object Recognition, Template Matching, Edge Based Probabilistic Model.

Conteúdo

Resumo	v
Abstract	vii
Conteúdo	ix
Lista de Figuras	xi
1 Introdução	1
1.1 Formulação do problema	1
1.2 Abordagem proposta	3
1.3 Contribuição	4
1.4 Estrutura	4
2 Estado da Arte	7
2.1 Detecção e reconhecimento de formas	7
2.2 Reconhecimento de caracteres	8
2.3 Detecção e reconhecimento de cartas de jogar	8
3 Detecção de Cartas	11
3.1 Transformada de Hough	12
3.2 Implementação da Transformada de Hough para rectângulos	12
3.3 Compensação de erros nas dimensões	14
3.4 Detecção de máximos da transformada de Hough	15
3.5 Detecção dos rectângulos	16
4 Classificação de Cartas com um Modelo Probabilístico Rígido	19
4.1 <i>Edge features</i>	19
4.2 Modelo probabilístico	21
4.3 Classificação	22
4.4 Computação	23

4.5	Treino do modelo	24
5	Classificação com um Modelo Deformável	27
5.1	Modelo deformável	27
5.2	Classificação	28
5.3	Computação	29
5.4	Treino do Modelo	30
6	Classificação por <i>Template Matching</i>	33
6.1	Implementação	33
6.2	Classificação	34
6.3	Treino do Modelo	34
7	Detecção e Reconhecimento em Perspectiva	37
7.1	Modelo da Câmera	37
7.2	Transformação Projectiva	38
7.3	Homografia	40
7.4	Calibração da imagem	41
8	Resultados	43
8.1	Detecção de cartas	43
8.2	Classificação de forma rígida	47
8.3	Modelo probabilístico deformável	48
8.4	Classificação com correcção de perspectiva	49
9	Conclusão	53
	Bibliografia	55

Lista de Figuras

1.1	Tipo de carta e seus componentes usados neste trabalho.	2
1.2	(a) Os 13 valores diferentes. (b) Os 4 naipes: Paus, Espadas, Copas e Ouros	2
1.3	Exemplo de uma imagem típica do problema a resolver.	3
1.4	Exemplos dos três tipos de oclusões parciais. (a) Oclusão vertical. (b) Oclusão horizontal. (c) Oclusão oblíqua.	3
3.1	Pose de uma carta.	11
3.2	Processamento da imagem para posterior detecção de rectângulos. (a) Imagem original. (b) Binarização do módulo do gradiente. (c) Resultado da detecção de contornos exteriores.	13
3.3	Exemplo de um preenchimento do acumulador dado \mathbf{x}	14
3.4	Exemplo de um preenchimento do acumulador dado \mathbf{x} , com compensação de erros . . .	16
3.5	Três tipos de detecções erradas. (a) Múltiplas detecções da mesma carta. (b) Detecção de rectângulo sem estar inteiramente sobre uma carta. (c) Detecção de um rectângulo sem que nenhum dos cantos coincida com um canto de carta.	17
3.6	Dois exemplos de detecções erradas sem solução, quando usada apenas a informação dos contornos.	18
4.1	As seis diferenças de intensidade comparadas na identificação de um <i>edge</i> vertical . . .	20
4.2	(a) Um exemplo de uma imagem de um valor. (b) Os <i>edges</i> detectados na direcção de $\mathbf{v} = (1, 0)$. (c),(d),(e),(f) Os mapas de <i>edges</i> após dilatação 3×3 , para quatro direcções diferentes, correspondentes aos quatro valores do vector \mathbf{v}	20
4.3	Duas detecções de rectângulos. As linhas vermelhas mostram o canto do rectângulo detectado. (a) Rectângulo detectado no local esperado. (b) Rectângulo detectado com um certo desvio face ao esperado.	22
4.4	Mapas de probabilidades para a classe do valor K . Em cima, os mapas para as direcções $\mathbf{v}: (1, 0), (1, 1), (0, 1), (-1, 1)$. Em baixo, os mapas para os sentidos contrários.	24

5.1	Classificação com modelo deformável. (a) Imagem original, sem deformações. (b) Imagem a classificar, resultante de uma distorção afim de (a). (c) As <i>features</i> detectadas na imagem (b), para uma orientação horizontal. (d) Mapa de probabilidades para a mesma orientação, sem deformações. (e) Mapa de probabilidades após serem estimados os desvios de cada parte e pós a operação de <i>patchwork</i> . (f) Os desvios estimados de cada parte e a forma como ficam distribuídas as partes. Os pontos representam os pontos de referência \mathbf{y}_i e as linhas os desvios \mathbf{v}_i . As regiões mais claras representam regiões com maior número de partes sobrepostas.	30
5.2	Treino do modelo POP. (a) Os mapas de <i>edges</i> para uma orientação horizontal nas imagens de treino, na região correspondente a uma <i>parte</i> . (b) A mesma <i>parte</i> treinada, pela média das imagens em (a). (c) O mapa de probabilidades $p_e(\mathbf{x})$ após a operação de <i>patchwork of parts</i> , com a região correspondente à parte treinada evidenciada a vermelho.	31
6.1	Treino dos modelos para <i>template matching</i> . (a) O modelo treinado para o valor “Ás”. (b) Uma imagem de treino para o valor “Ás”. (c)(d) O mesmo para o naipe “Espadas”.	35
7.1	Modelo da câmara <i>pinhole</i> . O objecto no ponto P do mundo físico é projectado em p no plano de imagem pelo raio que passa no <i>pinhole</i> . A matemática é equivalente com plano de imagem ou com o plano de imagem invertido, sendo com este último mais simples. Ambos os planos de imagem estão à mesma distância f do plano do <i>pinhole</i> , que está a uma distância Z do objecto.	38
8.1	Detecção de cartas em oclusão. Os rectângulos a vermelho representam as cartas detectadas e correctamente classificadas. A verde estão representados os rectângulos detectados mas rejeitados na fase de classificação. (a) Disposição que não permite a presença de rectângulos do tamanho das cartas no interior dos contornos. (b) Disposição que permite a presença de vários falsos rectângulos.	44
8.2	Detecção de cartas em oclusão, com apenas uma pequena parte do contorno do 3 de Espadas visível.	45
8.3	Detecção com rejeição por proximidade. (a) Rectângulos detectados com os três métodos de rejeição activos. (b) Rectângulos detectados sem a rejeição por proximidade.	45
8.4	Detecção com rejeição por correspondência de cantos das cartas e dos rectângulos. (a) Rectângulos detectados com os três métodos de rejeição activos. (b) Rectângulos detectados sem a rejeição por correspondência de cantos.	46
8.5	Detecção na presença de oclusões horizontais, havendo quatro cartas presentes.	46

8.6	Variações de iluminação. (a) Variação de luminosidade de -40 . (b) Variação de contraste de -60 . (c) Variação de luminosidade de -20 . (d) Variação de contraste de -30 . (e) Imagem original. (f) Variação de luminosidade de $+20$. (g) Variação de contraste de $+30$. (h) Variação de luminosidade de $+40$. (i) Variação de contraste de $+60$	48
8.7	Taxas de reconhecimento com variação de (a) Luminosidade; (b) Contraste. As linhas a azul representam o <i>template matching</i> . As linhas a vermelho representam o modelo probabilístico rígido.	49
8.8	Correcção de perspectiva pouco acentuada. (a) Imagem original. (b) Imagem corrigida a partir de baixa resolução. (c) Imagem corrigida a partir de alta resolução.	50
8.9	Correcção de perspectiva acentuada. (a) Imagem original. (b) Imagem corrigida a partir de baixa resolução. (c) Imagem corrigida a partir de alta resolução.	50

Lista de abreviaturas

EM - Expectation-Maximization

GHT - Generalized Hough Transform - Transformada de Hough Generalizada

MAP - Máximo *a posteriori*

OCR - Optical Character Recognition - Reconhecimento Óptico de Caracteres

POP - Patchwork of Parts

List of Abbreviations

EM - Expectation-Maximization

GHT - Generalized Hough Transform

MAP - Maximum *a posteriori*

OCR - Optical Character Recognition

POP - Patchwork of Parts

Capítulo 1

Introdução

Com a crescente procura de soluções de vídeo-vigilância e com o conseqüente aumento da presença destas no mundo actual, a tecnologia tem caminhado no sentido de automatizar a monitorização das sequências de vídeo, através de técnicas de processamento de imagem e visão por computador. Esta monitorização é feita tendo muitas vezes como objectivo a segurança, mas também por motivos estatísticos, como contagem de pessoas e de veículos.

Neste sentido, a indústria dos jogos está interessada em monitorização e registo automáticos do estado dos jogos, para prevenção de perdas por enganos, irregularidades e incumprimento de regras, assim como para fins estatísticos.

Este trabalho dedica-se à detecção e classificação de cartas de jogar, sendo necessária a contagem e identificação mesmo com oclusões parciais. São abordadas técnicas de detecção de formas e de reconhecimento de objectos.

1.1 Formulação do problema

Havendo diversos tipos de cartas de jogar comercialmente disponíveis, caracteriza-se o tipo de carta usado neste trabalho e definem-se os seus principais componentes. A figura 1.1 exemplifica uma carta do tipo usado, e os seus componentes relevantes. Definem-se como índice o conjunto dos símbolos presentes no canto superior esquerdo, e no canto inferior direito, rodado de 180°. O índice é composto por um de 13 valores, de 2 a 10 e A (ás), J (valetes), Q (dama) e K (rei); e pelo naipe, havendo quatro naipes diferentes: paus, espadas, copas e ouros. As cartas de paus e espadas possuem o índice de cor preta, e as cartas dos restantes naipes possuem o índice de cor vermelha. Uma carta pode assim ter um de 13 valores, de um de 4 naipes, havendo assim 52 cartas diferentes. Os diferentes valores e naipes estão representados na figura 1.2.

As cartas usadas são do tipo Anglo-Americano, com dois índices “jumbo” (de tamanho grande, quando comparado com outros tamanhos de índice), de dimensões de “poker” (63 mm × 88 mm).

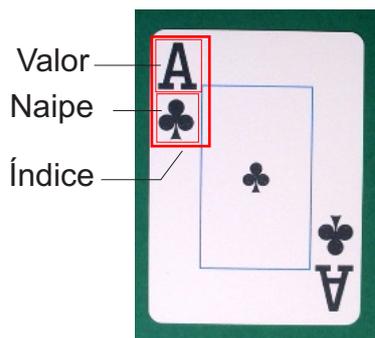


Figura 1.1: Tipo de carta e seus componentes usados neste trabalho.

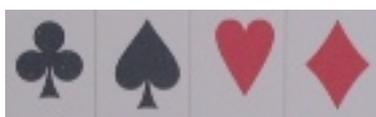
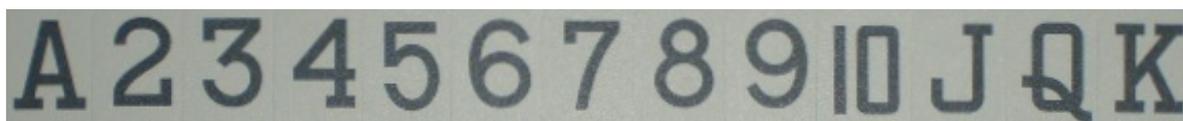


Figura 1.2: (a) Os 13 valores diferentes. (b) Os 4 naipes: Paus, Espadas, Copas e Ouros

O problema de detecção e reconhecimento de cartas de jogar é posto da seguinte forma: Diversas cartas são colocadas numa superfície escura e uniforme, em qualquer rotação, podendo estar parcialmente em cima de outras, mas de forma que todas tenham sempre pelo menos um dos índices da carta visível. Neste caso ocorrem oclusões parciais, que serão classificadas em três tipos. Todas as cartas presentes deverão ser detectadas e reconhecidas. Considera-se também que podem haver repetições de cartas, permitindo assim usar mais cartas do que as 52 diferentes. A figura 1.3 exemplifica uma cena representativa do problema.

Os três tipos de oclusão parcial, exemplificados pela figura 1.4, são os seguintes:

Vertical: Uma carta oclui parte da de trás, estando as duas cartas alinhadas em altura.

Horizontal: De forma semelhante à oclusão vertical, mas estando as cartas alinhadas em largura.

Oblíqua: A carta é ocluída sem qualquer alinhamento dos seus contornos, estando pelo menos um dos cantos de uma carta sobre outra, mas podendo estar mais do que um canto escondido.

Chama-se também de oclusão múltipla à situação em que uma carta é ocluída parcialmente por mais de uma carta.

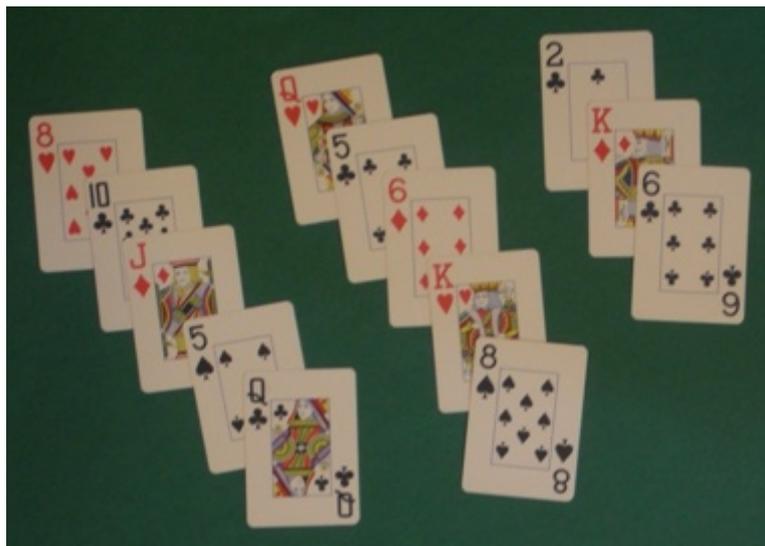


Figura 1.3: Exemplo de uma imagem típica do problema a resolver.

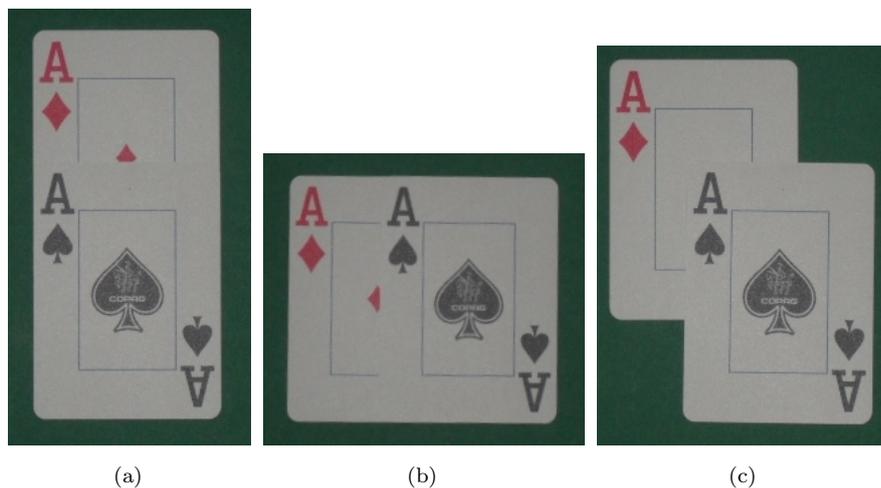


Figura 1.4: Exemplos dos três tipos de oclusões parciais. (a) Oclusão vertical. (b) Oclusão horizontal. (c) Oclusão oblíqua.

1.2 Abordagem proposta

O método proposto divide-se em duas fases principais: a detecção das cartas, e a sua classificação. Esta mesma abordagem é seguida por outros trabalhos semelhantes [1, 2], ainda que os métodos usados sejam diferentes. A detecção é feita com recurso a uma Transformada de Hough para rectângulos, usando a informação dos contornos da imagem das cartas. Desta forma a detecção é possível em casos de oclusão parcial. A posterior classificação das cartas é feita pela classificação dos índices destas. Fazendo uma detecção eficaz dos rectângulos que incluem as várias cartas, a segmentação dos índices é trivial, facilitando assim o processo de classificação. Como será mostrado, esta classificação é baseada no método proposto por Amit e Trouvé [3], um modelo probabilístico deformável, ainda que tenha sido

usada para os resultados finais o método na forma de modelo rígido.

1.3 Contribuição

Havendo vários métodos de classificação de formas, objectos ou caracteres, este trabalho procura usar e analisar vários métodos existentes, com certas variações para aplicação ao problema presente. Estando o problema dividido em duas fases, a detecção de cartas e a sua classificação, foram abordadas diferentes técnicas para cada uma delas.

Na primeira fase, é feita uma detecção de formas geométricas (rectângulos) usando a informação dos contornos da imagem, com uma variação original da Transformada de Hough desenvolvida pelo autor. Na segunda fase, o problema da classificação é tratado por três diferentes métodos: dois deles são métodos probabilísticos, usando *edge features* binárias, sendo um destes um modelo deformável; o outro método usado é o *template matching*.

As principais contribuições deste trabalho são: o desenvolvimento de um método eficaz de detecção e reconhecimento de cartas de jogar, mesmo na existência de oclusões parciais, pela integração de vários métodos de visão por computador e processamento de imagem; um método de detecção de rectângulos em escala fixa, com oclusões; e uma comparação entre vários métodos de classificação de objectos.

1.4 Estrutura

A dissertação está organizada em nove capítulos.

No capítulo 2 é feita uma breve análise de métodos existentes para vários problemas de visão por computador, que podem ser usados em aplicações semelhantes à deste trabalho.

O capítulo 3 descreve o método desenvolvido para detecção de cartas de jogar, mesmo em oclusão parcial, sob condições de rigidez. São explicados os problemas inerentes a esta detecção, e o que foi feito para resolvê-los.

No capítulo 4 é apresentado um dos métodos de classificação usados. Trata-se de um método probabilístico rígido, baseado em *features* de contornos. É explicado o cálculo destas *features*, a formulação do modelo probabilístico, e como é feita a classificação. É também explicado como é feito o treino do modelo. O capítulo 5 explica um outro método de classificação probabilístico. Trata-se de uma extensão do modelo anterior mas deformável, usando as mesmas *features*. No capítulo 6 é apresentado o último método de classificação usado, este por *template matching*. É explicado como este é implementado, e como é feita a classificação e o treino.

O capítulo 7 apresenta o que foi realizado para corrigir as imagens na presença de deformação de perspectiva. É feita uma introdução simples à teoria da geometria de perspectiva, é explicado como esta é usada e como é feita a correcção das imagens.

A demonstração dos resultados obtidos é feita no capítulo 8. São também comparados os diferentes

métodos de classificação, de acordo com a sua robustez à iluminação e deformação, assim como a sua eficiência computacional.

O capítulo 9 conclui a dissertação, apresentando uma descrição simples do que mais poderia ser feito neste trabalho.

Capítulo 2

Estado da Arte

A detecção e reconhecimento de objectos permanece um problema de visão por computador em aberto. A aplicação deste problema a cartas de jogar é um exemplo, e assim esta revisão do estado da arte incidirá tanto sobre métodos de detecção e reconhecimento de cartas de jogar como de objectos em geral.

2.1 Detecção e reconhecimento de formas

Muito trabalho tem sido desenvolvido na detecção e reconhecimento de formas geométricas. A Transformada de Hough Generalizada (GHT) [4] e juntamente com esta vários métodos de votação continuam a ser utilizados em diversas aplicações de visão por computador, para vários tipos de detecção de objectos, tanto formas simples como objectos com textura, através do uso de *features*. Por permitir a detecção de formas geométricas de forma simples apenas com a informação do gradiente e permitindo oclusões parciais, a GHT apresenta-se como um bom método para detecção e reconhecimento de formas simples.

Técnicas de detecção de rectângulos têm sido exploradas, conseguindo detectar rectângulos mesmo em perspectiva [5] e com oclusão [6], tendo como principal objectivo a detecção destes em cenários de construção e arquitectura para *stereo matching* ou reconstrução 3D. Esta detecção poderá ser usada para detectar cartas em imagens que sofram de deformações de perspectiva com oclusão.

Um trabalho recente, a assinatura analítica (ANSIG) [7], propõe uma representação de formas 2D por meio de uma função analítica, robusta a escala e translação. A classificação sugerida permite também invariância a rotação. Este método consegue bons resultados em classificação (taxa de reconhecimento próxima de 100%), de forma simples e rápida, de imagens binárias. No entanto, uma boa segmentação é necessária, pois quaisquer pontos não pertencentes à forma a classificar, ou a omissão de pontos pertencentes, contribuirão para uma representação diferente do objecto. Este método é útil para reconhecimento de formas fixas, como logotipos ou caracteres, sendo um bom método para a

classificação dos índices das cartas.

2.2 Reconhecimento de caracteres

O problema de reconhecimento óptico de caracteres (OCR) é geralmente já considerado um problema resolvido em aplicações de reconhecimento de texto de documentos, onde se possui uma imagem bem definida com os caracteres a reconhecer. Assim existem várias soluções de software de OCR disponíveis, com taxas de erros de reconhecimento de caracteres na ordem de 1-3% [8].

Alguns dos actuais desafios em OCR são o reconhecimento automático de matrículas de veículos e o reconhecimento de caracteres manuscritos. O reconhecimento de matrículas envolve, para além da detecção desta, o reconhecimento dos caracteres nela presentes. Considerando que os sistemas de reconhecimento automático de matrículas analisam imagens captadas normalmente por câmeras de vigilância, de forma que a posição destas face a diferentes carros não é constante, uma imagem de uma matrícula pode apresentar diversos tipos de deformação dos caracteres. Nesta aplicação, a abordagem típica para classificação envolve a segmentação dos caracteres e a sua posterior classificação. O trabalho de Anagnostopoulos et al. [9] aplica esta abordagem ao problema, e apresenta uma revisão de vários métodos relacionados. Nesta revisão é mostrado que a taxa de reconhecimento dos caracteres é tipicamente superior a 95%, mas a taxa de reconhecimento global (incluindo a detecção da matrícula) é normalmente inferior a este valor, mas ainda assim é maioritariamente superior a 90%.

O reconhecimento de dígitos manuscritos é abordado em [3]. Este método, implementado em parte neste trabalho, usa um modelo deformável para se adaptar às várias deformações típicas de caracteres manuscritos. Ainda que o algoritmo não tenha sido feito especificamente para leitura de códigos postais, consegue resultados semelhantes a algoritmos mais antigos dedicados a este problema, com taxas de reconhecimento de 96.5%.

2.3 Detecção e reconhecimento de cartas de jogar

Pouco trabalho científico tem sido feito em reconhecimento de cartas de jogar. O problema é normalmente abordado por duas fases: a detecção da carta e a sua classificação.

Os métodos analisados baseiam-se na extracção de contornos para detecção[1, 2]. No entanto, estes métodos não mostram robustez a oclusões, oclusões essas que são frequentes em cenários de jogo de casino. Em [1] a detecção é feita apenas para uma carta isolada. O rectângulo é detectado estimando a caixa que melhor cobre a carta inteira (*bounding box*), e a partir daí a rotação da carta é estimada. Esta forma de detecção da carta torna impossível a robustez a oclusões parciais. A classificação é feita por *template matching* do valor e do naipe.

Em [2] a detecção é feita através dos contornos da carta. Um contorno com 4 vértices e uma dada área é considerado uma carta. Os contornos mais pequenos e com uma determinada relação

altura/comprimento são considerados “blocos de naipe” - símbolos dos naipes presentes no interior das cartas numeradas e do ás. No caso destas cartas, a classificação é feita por contagem dos “blocos de naipe”, cujo número corresponde ao valor da carta. Esta contagem envolve a classificação dos vários blocos, feita com *template matching* por correlação cruzada (*cross correlation*). O reconhecimento das cartas de figura é feito com recurso a SIFT[10], cujas *features* são calculadas no interior da carta. Caso o número de pontos-chave com correspondência ao modelo, feita por distância euclideana, seja superior a um certo valor, a carta é classificada como pertencente à classe do dado modelo. Esta classificação ignora o índice da carta, usando toda a superfície desta, sendo por isso impossível em situações de oclusão.

Capítulo 3

Detecção de Cartas

A primeira fase do método proposto é a fase de detecção, feita de forma independente da fase de reconhecimento. Esta primeira fase foi implementada por meio de um método original, baseado na Transformada de Hough Generalizada, sendo aplicado a rectângulos cujas dimensões são conhecidas *a priori*. Sob esta condição, o método não será robusto a qualquer deformação não rígida. É feita uma procura de possíveis poses dos rectângulos presentes: localização do centro $\mathbf{c} = (c_x, c_y)$ e orientação ϕ . Como as cartas são rotacionalmente simétricas em ângulos de 180° , considera-se como domínio de orientação $\phi \in [0, \pi[$. Define-se assim a pose de uma carta como $\theta = (\mathbf{c}, \phi) \in \Theta$, o espaço das poses admissíveis. Considera-se também que a orientação da carta é de 0° quando esta carta encontra-se na horizontal (lado mais comprido na horizontal). Uma representação gráfica da pose de uma carta está presente na figura 3.1.

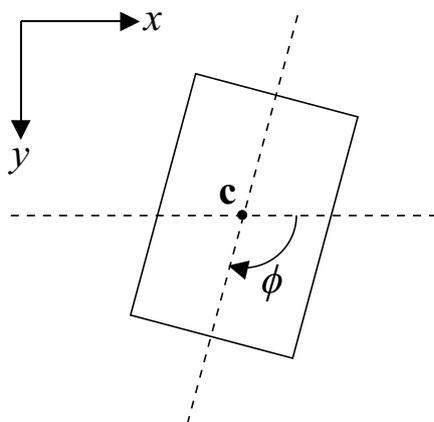


Figura 3.1: Pose de uma carta.

3.1 Transformada de Hough

A transformada de Hough, na sua forma geral, é um método de votação para detecção de objectos[4]. São usados pontos de uma imagem, tipicamente binarizada, pontos estes que votam nas possíveis poses do objecto a detectar tendo em conta a sua posição ou, de forma geral, um modelo do objecto. Considerando que um objecto está presente na figura, cada ponto presente votará nas várias poses que o possam originar, sendo estas representadas por pontos num espaço H de dimensionalidade igual ao número de parâmetros da pose a detectar. Desta forma, haverá um máximo em H correspondente à pose presente, não impossibilitando que também ocorram outros máximos locais.

A detecção das cartas pela transformada de Hough para rectângulos foi feita usando apenas a informação dos contornos e a orientação do gradiente. Considera-se que o rectângulo a detectar não tem os cantos arredondados, ao contrário das cartas usadas. Esta diferença é pequena face ao tamanho das cartas, permitindo assim uma detecção eficaz com um modelo simples: detectam-se rectângulos perfeitos, ao invés de rectângulos com cantos arredondados.

3.2 Implementação da Transformada de Hough para rectângulos

Seja $I(\mathbf{x})$ a imagem de escala de cinza a analisar e H um acumulador indexado pelos elementos do espaço das poses Θ . Considera-se conhecido o tamanho das cartas: $2R$ pixéis de altura e $2r$ pixéis de largura.

Começa-se por calcular o gradiente $G(\mathbf{x})$ da imagem I , usando máscaras de Sobel, vertical e horizontal. Das filtragens com cada máscara obtêm-se o módulo do gradiente, $M(x)$, e $\Phi(\mathbf{x})$, a sua orientação. O vector do gradiente tem o sentido do escuro para o claro. Como as cartas têm fundo branco e são colocadas numa superfície escura, o gradiente tem o sentido de fora para dentro da carta. De seguida $M(\mathbf{x})$ é binarizado, por *thresholding*, usando um valor de *threshold* proporcional ao seu máximo. É feito um processamento a $M(\mathbf{x})$, sendo alterado pelas funções *findContours* e *drawContours* da biblioteca livre OpenCV [11] para se obter apenas os contornos exteriores, removendo os contornos correspondentes aos índices e figuras no interior das cartas e ficando apenas com a metade exterior dos contornos das cartas. Estes contornos pela natureza da filtragem de Sobel, têm originalmente 2 pixéis de largura, ficando com metade da largura após o processamento. As várias fases do processamento de imagem estão demonstradas na figura 3.2. Os valores finais de $M(\mathbf{x})$ são mostrados na figura 3.2c, cujos pontos a branco representam os pontos utilizados na transformada de Hough.

Este processamento impõe uma limitação no método: para se obter os contornos externos, estes têm que formar linhas fechadas, pelo que os contornos devem ser bem detectados. No entanto, com boas condições de iluminação, mais concretamente, havendo bom contraste que permita contornos das cartas bem definidos, os contornos fechados são facilmente obtidos. Não podem também haver contornos fechados que contenham os contornos das cartas no seu interior, a fim de não rejeitar estes últimos

juntamente com os contornos interiores usuais.

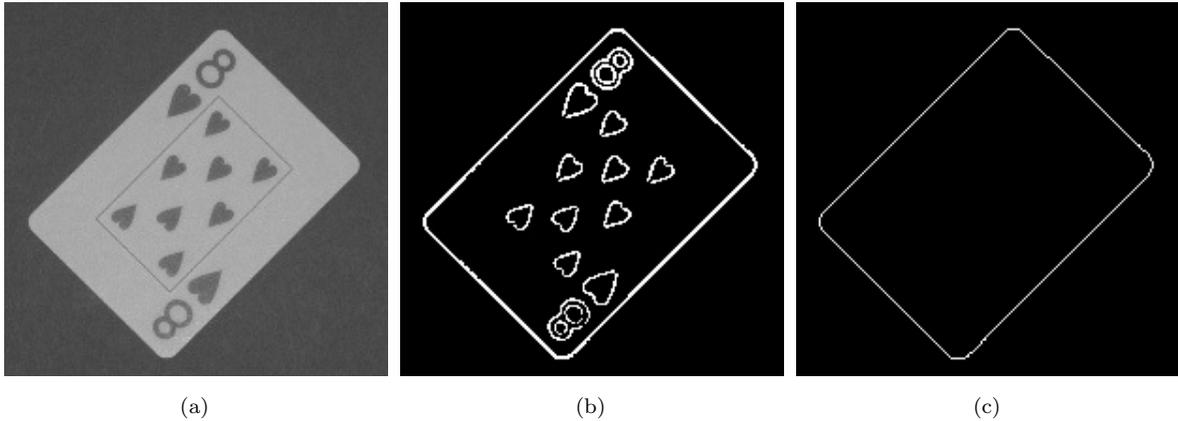


Figura 3.2: Processamento da imagem para posterior detecção de rectângulos. (a) Imagem original. (b) Binarização do módulo do gradiente. (c) Resultado da detecção de contornos exteriores.

O preenchimento do acumulador é feito pelos pontos resultantes do processamento anterior. Cada ponto $\mathbf{x} : M(\mathbf{x}) = 1$ irá incrementar os valores de $H(\theta_i)$, em que os $\theta_i \in \Theta$ são as poses que possam ter originado $M(\mathbf{x}) = 1$. Conhecendo a orientação do gradiente neste ponto, as poses possíveis são as das cartas cujo contorno esteja em \mathbf{x} , no lado do sentido de $\Phi(\mathbf{x})$, e com orientação igual $\Phi(\mathbf{x})$ caso se considere que \mathbf{x} pertence ao lado mais curto, ou com orientação $\Phi(\mathbf{x} + \frac{\pi}{2})$ caso pertença ao lado mais comprido. Desta forma, os possíveis centros da carta serão

$$c_R^j = \mathbf{x} + rv + jw, \quad j = -R, \dots, R \quad (3.1)$$

$$c_r^k = \mathbf{x} + Rv + kw, \quad k = -r, \dots, r \quad (3.2)$$

em que v e w são, respectivamente, vectores de norma unitária com sentido e direcção dados por $\Phi(x)$ e $\Phi(x) + \frac{\pi}{2}$. Os centros c_R^j correspondem à hipótese de \mathbf{x} estar presente num dos lados mais compridos da carta, colocando os centros numa linha paralela ao contorno com distancia r . De forma análoga, c_r^k corresponde aos possíveis centros da carta considerando a hipótese de \mathbf{x} estar presente num dos lados mais curtos da carta.

Desta forma, os pontos θ_i a incrementar no acumulador H serão dados pelo conjunto $\Theta_c \subset \Theta$:

$$\Theta_c = \{\theta^j = (c_R^j, \Phi(x) + \frac{\pi}{2}), j = -R, \dots, R\} \cup \{\theta^k = (c_r^k, \Phi(x)), k = -r, \dots, r\}. \quad (3.3)$$

A figura 3.3 mostra este preenchimento para um ponto \mathbf{x} . Realizando este preenchimento para todos os pontos $\mathbf{x} : M(\mathbf{x}) = 1$, haverá necessariamente várias intersecções, e a pose na qual todos os conjuntos se intersectam será a pose correcta, considerando que não há qualquer folga nas dimensões ou nas rotações possíveis, considerando portanto o caso ideal.

A detecção do rectângulo é dada pelos máximos locais do acumulador. O valor de cada máximo do acumulador representa o número de pontos de contorno que pertencem ao rectângulo representado por

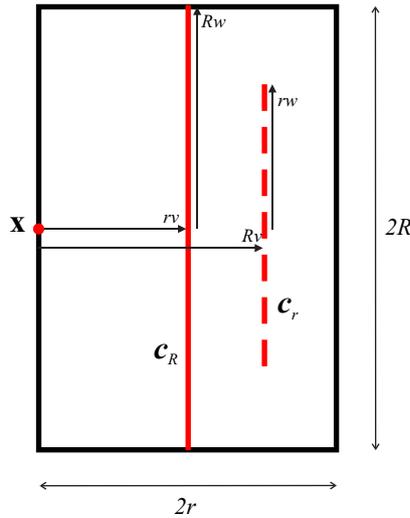


Figura 3.3: Exemplo de um preenchimento do acumulador dado \mathbf{x} . As linhas a vermelho representam os possíveis centros dada a localização de \mathbf{x} e a orientação do gradiente. A linha contínua agrupa os centros na hipótese de \mathbf{x} estar num dos lados compridos. A linha tracejada agrupa os centros na hipótese de \mathbf{x} estar num dos lados curtos.

esse máximo, podendo ser inferior ao número de pontos de um contorno completo. É desta forma que se resolve o problema da oclusão parcial. No entanto, mesmo que um rectângulo seja detectado com apenas alguns pontos de contorno, estes pontos têm de pertencer a pelo menos dois lados adjacentes, i.e. perpendiculares. Caso os únicos pontos de contorno usados na detecção sejam de lados opostos, i.e. paralelos, o problema é equivalente ao problema da oclusão vertical ou horizontal. Como descrito em 1.1, neste caso haverão vários rectângulos possíveis dados estes contornos, com igual probabilidade de serem o correcto. O mesmo acontece caso apenas um dos lados da carta esteja presente na imagem.

3.3 Compensação de erros nas dimensões

Como a imagem $I(\mathbf{x})$ está representada no domínio discreto, tendo portanto as cartas dimensões inteiras, terá que se ter em atenção o facto de que um ponto a uma distância R de um lado do rectângulo, no sentido do gradiente, pode não coincidir com um ponto à mesma distância do lado oposto. Como exemplo tome-se uma carta com comprimento de $2R = 20$ pixéis e orientação de 0° . Neste caso, se um dos lados estiver em $x = 0$, o lado oposto estará em $x = 19$, respeitando o comprimento de 20 pixéis. Desta forma o centro estará em $c_x = 10$ ou $c_x = 9$, por não haver um ponto equidistante dos dois lados, o que levará a que no preenchimento do acumulador não haja uma intersecção das contribuições dos lados opostos e assim não se obtenha um máximo para a pose correcta da carta. Considerando outras formas de medir um comprimento, incluindo ou não alguns ou todos os contornos, o problema persistirá. Este problema também pode ocorrer por ruído, que pode alterar a localização dos contornos face ao

esperado, ou por um erro na medição da carta, mesmo que seja de apenas um pixel.

Para fazer face a este problema, deve ser feito um incremento do acumulador não apenas nos centros indicados acima, mas numa vizinhança destes, permitindo assim pequenos desvios dos contornos face ao esperado. O resultado desta acumulação pode ser visto como uma suavização do acumulador[4]. Por se considerar que na orientação não ocorre este problema pelo preenchimento do acumulador, usa-se o mesmo valor de ϕ nos vários pontos (x, y) adicionalmente preenchidos. Para cada ponto do acumulador preenchido como dito anteriormente, são também incrementados os pontos imediatamente adjacentes ao primeiro, na direcção do gradiente do contorno. Desta forma, dado um ponto \mathbf{x} nas mesmas condições anteriormente referidas, e sendo os vectores v e w definidos de igual forma, as hipóteses para centros são agora dadas por

$$c_{R-1}^j = \mathbf{x} + (r-1)v + jw, \quad (3.4)$$

$$c_{R_0}^j = \mathbf{x} + rv + jw, \quad (3.5)$$

$$c_{R_1}^j = \mathbf{x} + (r+1)v + jw, \quad j = -R, \dots, R \quad (3.6)$$

$$c_{r-1}^k = \mathbf{x} + (R-1)v + kw, \quad (3.7)$$

$$c_{r_0}^k = \mathbf{x} + Rv + kw, \quad (3.8)$$

$$c_{r_1}^k = \mathbf{x} + (R+1)v + kw, \quad k = -r, \dots, r \quad (3.9)$$

e como anteriormente, os pontos θ_i a incrementar no acumulador H serão dados pelo conjunto $\Theta_c \subset \Theta$:

$$\begin{aligned} \Theta_c = & \{ \theta_i^j = (c_{R_i}^j, \Phi(x) + \frac{\pi}{2}), \quad j = -R, \dots, R, \quad i = -1, 0, 1 \} \\ & \cup \{ \theta_i^k = (c_{r_i}^k, \Phi(x)), \quad k = -r, \dots, r, \quad i = -1, 0, 1 \}. \end{aligned} \quad (3.10)$$

A figura 3.4 mostra os conjuntos de pontos considerados como centros possíveis, representados pelos rectângulos, que neste caso se consideram ter uma largura de 3 pixels.

De forma a não penalizar pequenos desvios de um pixel nos contornos, fez-se o preenchimento do acumulador sempre incrementando em 1 unidade (equivalente a uma suavização quadrada ou dilatação morfológica), ao invés de usar um decaimento (p.e. gaussiano), cujos resultados assemelham-se aos resultantes do preenchimento simples.

3.4 Detecção de máximos da transformada de Hough

Após o preenchimento do acumulador, são procurados os máximos deste para recolher os rectângulos com maior probabilidade de existir. Esta procura é feita comparando cada ponto do acumulador com a sua vizinhança de raio = 1 em 3 dimensões, fazendo assim 26 comparações para cada ponto. Um máximo é declarado caso o seu valor seja maior ou igual aos valores da sua vizinhança. Esta procura não é feita percorrendo todo o acumulador, visto que os valores do acumulador são guardados de forma

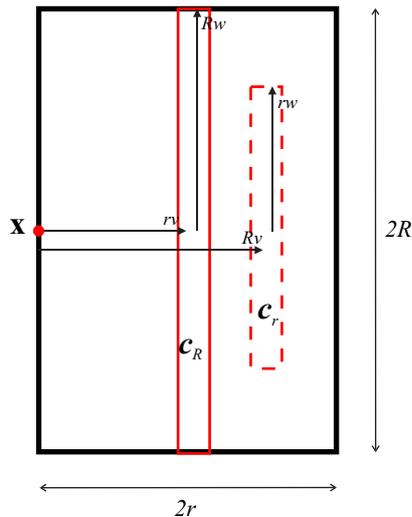


Figura 3.4: Exemplo de um preenchimento do acumulador dado \mathbf{x} , com compensação de erros. Os retângulos a bordo vermelho representam os conjuntos de possíveis centros dada a localização de \mathbf{x} e a orientação do gradiente. O retângulo com bordo contínuo agrupa os centros na hipótese de \mathbf{x} estar num dos lados compridos. O retângulo com bordo tracejado agrupa os centros na hipótese de \mathbf{x} estar num dos lados curtos.

esparsa, onde apenas os valores não nulos estão alocados. Esta forma de armazenamento dos dados foi considerada conveniente, visto que o acumulador para uma imagem com várias cartas consegue ficar com apenas cerca de 2% ocupação de valores não nulos. Considera-se também que um ponto pode ser um máximo apenas se tiver um valor acima de um dado *threshold*, que representa o número mínimo de pontos de contorno admissível para que seja considerada a existência de uma carta.

Desta procura resulta uma lista de possíveis rectângulos, ordenada pelo valor no acumulador.

3.5 Detecção dos rectângulos

Pela natureza da transformada de Hough, uma carta com apenas parte do seu contorno visível pode ser detectada, desde que o *threshold* de detecção de máximos tenha um valor que admita o tamanho do contorno visível e que o contorno visível contenha pelo menos dois lados adjacentes, como explicado em 3.1. Este facto é favorável ao problema em questão, pois permite detecção mesmo com oclusões parciais. Por outro lado, poderão haver rectângulos detectados cujos contornos apenas parcialmente coincidem com os contornos detectados da imagem. Outras detecções incorrectas serão dadas pelo facto de de uma imagem com cartas não resultarem contornos perfeitos e, com o preenchimento do acumulador com compensação de erros, serem detectados vários máximos do acumulador, alguns muito próximos uns dos outros. Assim, deve ser feita uma análise a cada um dos rectângulos detectados, para se avaliar se se trata de um rectângulo possível, usando outros dados para além dos contornos. Esta

análise é feita em três fases:

A primeira fase desta análise consiste em rejeitar rectângulos muito próximos (supressão de não máximos). Analisando a lista de máximos do acumulador por ordem decrescente dos seus valores, um rectângulo é rejeitado se algum rectângulo, cujo centro e orientação estejam a uma distância inferior a um certo valor deste, já tenha sido admitido por todas as fases do processo como um rectângulo válido. Esta forma de supressão de não máximos mostrou-se computacionalmente mais eficiente que uma supressão típica por janela deslizante.

A segunda fase procura eliminar rectângulos cujos contornos não coincidam com os contornos da imagem das cartas. Verifica-se se dentro do rectângulo existem pontos de contornos, dando uma certa tolerância para certos desvios deste face à posição que seria considerada correcta. Caso existam, o rectângulo é rejeitado.

A terceira e última fase verifica se pelo menos um dos cantos do rectângulo corresponde a um canto de uma carta, de forma muito simplificada, a fim de não despender muita computação neste passo. Tal é verificado usando a imagem final dos contornos, como na figura 3.2c, mas com o interior preenchido a branco. Percorrendo todos os cantos do rectângulo a analisar, verifica-se numa vizinhança do canto deste se o número de pixels brancos está próximo de um quarto do número de pixels nesta vizinhança. Caso esteja, o canto é considerado um canto válido e o rectângulo é aceite. Se nenhum dos cantos do rectângulo for aceite, o rectângulo é rejeitado.

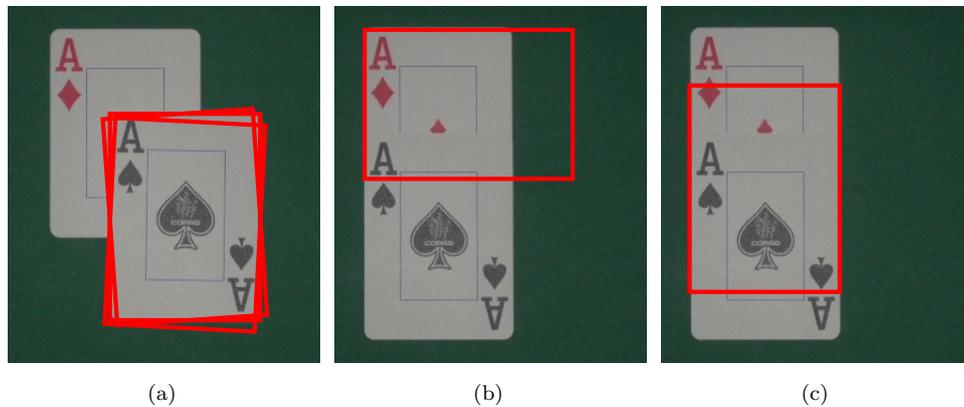


Figura 3.5: Três tipos de detecções erradas. (a) Múltiplas detecções da mesma carta. (b) Detecção de rectângulo sem estar inteiramente sobre uma carta. (c) Detecção de um rectângulo sem que nenhum dos cantos coincida com um canto de carta.

A figura 3.5 exemplifica alguns tipos de detecções incorrectas que se procuram corrigir com estes três testes. A figura 3.5c em especial mostra um problema da detecção de cartas em caso de oclusões verticais ou horizontais usando apenas a informação dos contornos. Caso numa imagem hajam mais do que duas cartas num destes tipos de oclusão, o problema de detecção pela transformada de Hough está mal posto. Nestes casos, os contornos serão iguais para qualquer número de cartas alinhadas, e

apenas se pode garantir a detecção das duas cartas das pontas. A detecção de uma carta entre estas duas será possível, ainda que nada se possa concluir acerca da validade desta apenas com a informação de contornos. Assim, a terceira fase da rejeição de rectângulos pode não permitir mais do que duas cartas em oclusão vertical ou horizontal.

Usando também apenas a informação dos contornos, há certos tipos de detecção incorrecta que não podem ser resolvidos mesmo com a análise anteriormente feita. Tratam-se de rectângulos detectados cujos contornos ou parte destes sejam coincidentes com os contornos detectados na imagem, estando o rectângulo contido no interior dos contornos da imagem e pelo menos um dos cantos corresponda a um canto verdadeiro. Estes tipos de detecção são exemplificados na figura 3.6. Mesmo visualmente, pelos contornos, nada garante o observador que não esteja presente uma carta no rectângulo detectado.

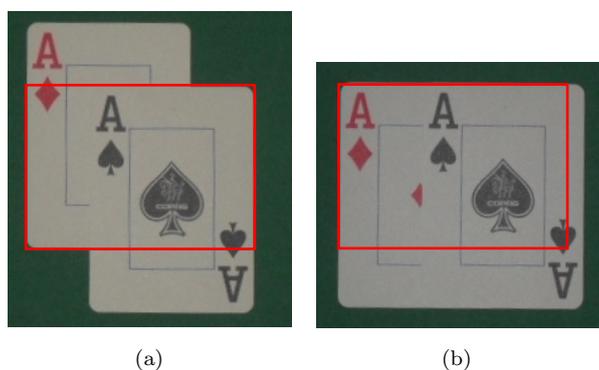


Figura 3.6: Dois exemplos de detecções erradas sem solução, quando usada apenas a informação dos contornos.

De mencionar que todo o processo de detecção é feito de forma conservadora, de forma a que não hajam falsos negativos na detecção dos rectângulos. Quaisquer rectângulos a mais poderão ser rejeitados na fase da classificação, e qualquer carta cujo rectângulo correspondente não seja detectado não será classificada.

Caso um rectângulo seja considerado válido, é feita uma cópia da imagem na vizinhança do canto superior esquerdo e do canto inferior direito, considerando uma carta na rotação de 90° . Obtém-se desta forma uma imagem do índice, e assim uma imagem do valor e do naipe da carta. Usando a informação da rotação do rectângulo calcula-se a rotação da imagem de forma a se obter uma imagem do valor e do naipe na rotação original, como demonstrado na figura 1.2. Esta cópia é feita de forma a ser apanhada parte da imagem fora do rectângulo e com dimensões superiores à largura esperada do índice. Assim permite-se algum espaço para pequenas falhas na localização e rotação do rectângulo face à pose da carta detectada, pois o rectângulo detectado não está frequentemente perfeitamente alinhado com os contornos da carta e, nestes casos, esta segmentação do índice poderia ser feita incorrectamente, afectando o resultado da classificação. A classificação é realizada exclusivamente pela classificação destas imagens dos índices.

Capítulo 4

Classificação de Cartas com um Modelo Probabilístico Rígido

A segunda fase do método proposto é a fase de classificação de cartas. Após a fase de detecção, uma lista de candidatos a cartas é dada como entrada no algoritmo de classificação. Cada um dos elementos da lista contém a localização e rotação do candidato a carta, bem como as imagens dos índices da carta, uma para o valor e outra para o naipe. Assim, considera-se que tanto o valor da carta como o seu naipe estão bem segmentados. A classificação da carta em valor e naipe é feita exclusivamente pela classificação das imagens do índice.

O método usado segue e implementa parte do método proposto por Amit e Trouvé [3], considerando apenas o modelo rígido.

4.1 *Edge features*

O modelo usado será baseado em *edge features* binárias e orientadas, calculadas em cada ponto da imagem. Estas *features* funcionam como detecção de *edges*, sendo tradicionalmente pouco variantes a transformações fotométricas. São definidas de acordo com [12], sendo o seu cálculo descrito em seguida. Como anteriormente, a imagem em escala de cinza a analisar é definida como $I(\mathbf{x})$, sendo indexada por um valor \mathbf{x} em 2D.

Seja \mathbf{v} um dos vectores $(1, 0), (1, 1), (0, 1), (-1, 1)$, e seja \mathbf{w} a rotação de 90° de \mathbf{v} . Seja \mathbf{z} um pixel e $\mathbf{y} = \mathbf{z} + \mathbf{v}$, sendo assim um dos oito pixels mais próximos de \mathbf{z} . Sejam também $\mathbf{z}_1 = \mathbf{z} + \mathbf{w}, \mathbf{z}_2 = \mathbf{z} - \mathbf{w}, \mathbf{z}_3 = \mathbf{z} - \mathbf{v}$ e $\mathbf{y}_1 = \mathbf{y} + \mathbf{w}, \mathbf{y}_2 = \mathbf{y} - \mathbf{w}, \mathbf{y}_3 = \mathbf{y} + \mathbf{v}$. Um *edge*, e assim uma *feature*, está presente em \mathbf{z} se

$$|I(\mathbf{z}) - I(\mathbf{y})| > \max_{i=1,2,3} (\max(|I(\mathbf{z}) - I(\mathbf{z}_i)|, |I(\mathbf{y}) - I(\mathbf{y}_i)|)). \quad (4.1)$$

A orientação do *edge* é \mathbf{v} se $I(\mathbf{z}) > I(\mathbf{y})$ e $-\mathbf{v}$ caso contrário. Os oito tipos de *edge* correspondem às quatro direcções de \mathbf{v} , cada uma com dois possíveis sentidos, dados pelo sinal de $I(\mathbf{z}) - I(\mathbf{y})$. As

quatro direcções correspondem a uma direcção horizontal e três incrementos de 45° . A disposição dos vários pontos analisados no cálculo de um *edge* vertical ($\mathbf{v} = (1, 0)$) é mostrada na figura 4.1.

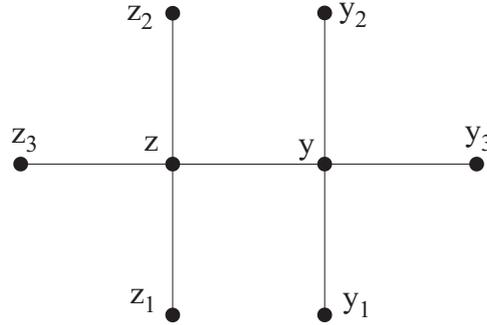


Figura 4.1: As seis diferenças de intensidade comparadas na identificação de um *edge* vertical

Assim, na detecção de uma *feature* são comparadas seis diferenças de intensidade. Estas *features* podem ser consideradas como máximos locais do módulo do gradiente. É também considerada a presença de uma *feature* apenas caso o módulo do gradiente seja superior a um dado valor, i.e., $|I(\mathbf{z}) - I(\mathbf{y})| > t$.

Após realizada a detecção de *edges*, cada um detectado é dilatado na sua vizinhança 3×3 . Esta dilatação procura trazer robustez a pequenas deformações e melhorar a classificação, com qualquer classificador usado. Na figura 4.2 estão presentes os mapas de *edges* para quatro orientações.

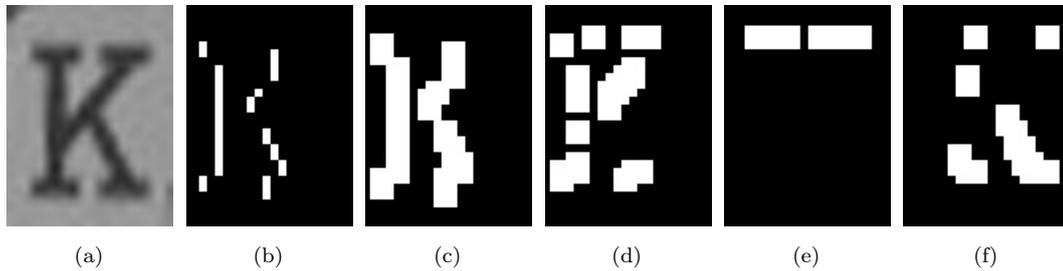


Figura 4.2: (a) Um exemplo de uma imagem de um valor. (b) Os *edges* detectados na direcção de $\mathbf{v} = (1, 0)$. (c),(d),(e),(f) Os mapas de *edges* após dilatação 3×3 , para quatro direcções diferentes, correspondentes aos quatro valores do vector \mathbf{v} .

A imagem a classificar é então representada por um conjunto de 8 mapas de *edges*. A imagem é representada por $X = \{X_e(\mathbf{x}) : e = 1, \dots, E\}$ onde $E = 8$, em que $X_e(\mathbf{x})$ é uma variável binária que indica a existência ou não de uma *feature* da orientação e no pixel \mathbf{x} .

Ainda que tenham sido usadas *features* simples de *edges*, poderiam ser usadas *features* de qualquer tipo, pois como é explicado em 4.2, o modelo não leva em conta o tipo mas apenas a presença ou não das *features* binárias, e a sua localização. Estas *features* foram usadas com sucesso em várias aplicações como reconhecimento de caracteres, de faces [3], de objectos 3D, imagens médicas [12] e reconhecimento de matrículas automóveis [13].

4.2 Modelo probabilístico

Para a classificação de um objecto, neste caso, o valor ou o naipe de uma carta, foi usado um modelo rígido. Ainda que seja assumido que o objecto a classificar está perfeitamente segmentado, esta segmentação não implica uma posição do objecto constante. Desta forma, a pose do objecto é dada por θ , que representa apenas a localização em 2D. Com este modelo rígido, θ será simplesmente um vector $\mathbf{r} = (r_x, r_y)$. No capítulo 5, θ terá em conta tanto a localização \mathbf{r} , como uma deformação, sendo portanto representado de uma forma mais complexa.

Define-se um mapa de probabilidades em 2D dado por $p_e(\mathbf{x})$. Assume-se que os *edges* detectados são condicionalmente independentes e que o objecto, isolado na imagem, está numa localização \mathbf{r} . A probabilidade de uma *feature* de orientação e estar activa no ponto \mathbf{x} é dada por

$$P(X_e(\mathbf{x}) = 1|\theta) = p_e(\mathbf{x}; \theta) \doteq p_e(\mathbf{x} - \mathbf{r}). \quad (4.2)$$

Uma translação do objecto por \mathbf{r} resulta numa translação para \mathbf{r} do mapa de probabilidades. Ainda que dificilmente os vários *edges* sejam independentes entre si, esta assunção permite formular um modelo probabilístico simples e intuitivo.

Para ter em consideração a presença de *edges* onde não seria esperado, portanto, no fundo, define-se um suporte do objecto em cada direcção dos *edges*, $S_e(\mathbf{r})$, e assume-se um modelo do fundo fora de $S_e(\mathbf{r})$. No fundo, os *edges* são também assumidos independentes, mas com probabilidades marginais constantes, iguais a p_{bgd} . O suporte $S_e(\mathbf{r})$ é definido como o conjunto de pontos \mathbf{x} onde a probabilidade marginal é maior que um dado valor de *threshold* p_{min} :

$$S_e(\mathbf{r}) = \{\mathbf{x} : p_e(\mathbf{x}; \mathbf{r}) \geq p_{\text{min}}\}. \quad (4.3)$$

Este conjunto permite modelar as áreas pertencentes ao objecto e as áreas que correspondem ao fundo, permitindo a definição de uma distribuição usando estas duas regiões. Assim, dada a localização \mathbf{r} do objecto, a função de verosimilhança dos dados das *features* X é dada por

$$L(\mathbf{r}) = P(X|\mathbf{r}) = \prod_e \prod_{\mathbf{x} \in S_e(\mathbf{r})} [p_e(\mathbf{x}; \mathbf{r})]^{X_e(\mathbf{x})} [1 - p_e(\mathbf{x}; \mathbf{r})]^{1 - X_e(\mathbf{x})} \prod_e \prod_{\mathbf{x} \notin S_e(\mathbf{r})} [p_{\text{bgd}}]^{X_e(\mathbf{x})} [1 - p_{\text{bgd}}]^{1 - X_e(\mathbf{x})}. \quad (4.4)$$

Este modelo corresponde a uma distribuição Binomial, modelando as presenças de *features* por uma distribuição de Bernoulli. Os termos da função de verosimilhança são maiores quando o mapa de probabilidades $p_e(\mathbf{x})$ sugere uma probabilidade elevada de existir uma *feature* $X_e(\mathbf{x})$ (em \mathbf{x}) e quando o mapa sugere uma probabilidade baixa e a *feature* não existe. Os termos são menores, e portanto penalizantes, quando o mapa de probabilidades sugere a existência de *features* e estas não existem e quando o mapa sugere a não existência de *features* e estas estão presentes.

4.3 Classificação

Como mencionado anteriormente, a segmentação do objecto é assumida correcta, mas a localização efectiva do objecto não é esperada que seja correcta ($\mathbf{r} = (0,0)$). Considerando que o objecto está presente (pela prévia detecção de rectângulos), e que apenas não se sabe a sua classe e localização, não é considerado nenhum conhecimento *a priori* da localização. Esta localização pode variar muito, pois depende da exactidão do rectângulo detectado face à posição correcta da carta. O rectângulo detectado é (idealmente) único para cada carta presente, e a sua exactidão depende de factores como ruído da imagem ou variações fotométricas, não tendo sido modelado. Por esta razão a classificação é feita pela máxima verosimilhança, ao invés de uma classificação bayesiana de máximo *a posteriori* (MAP). Em geral, para um caso como este, um classificador de MAP penalizaria a classificação tanto mais quanto maior fosse o desvio \mathbf{r} face ao esperado. Esta classificação poderia, sendo apresentada uma imagem a classificar, escolher como correcta uma classe errada numa localização, ao invés da classe correcta numa localização mais distante. A figura 4.3 demonstra dois exemplos de detecções, uma mais exacta que a outra. As linhas vermelhas mostram o canto do rectângulo detectado. Em ambos os casos, o índice deve ser classificado sem se penalizar por um maior desvio na figura 4.3a.

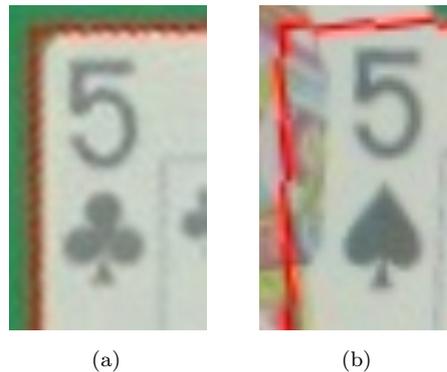


Figura 4.3: Duas detecções de rectângulos. As linhas vermelhas mostram o canto do rectângulo detectado. (a) Rectângulo detectado no local esperado. (b) Rectângulo detectado com um certo desvio face ao esperado.

Seja $P_c(X|\mathbf{r}) \doteq L(\mathbf{r}, c)$ a verosimilhança dos dados X corresponderem à classe $c \in C$ (o espaço das diferentes classes), dado um deslocamento \mathbf{r} . A classificação é dada pela maximização

$$\hat{Y} = \arg \max_c \max_{\mathbf{r}} P_c(X|\mathbf{r}) = \arg \max_c \max_{\mathbf{r}} L(\mathbf{r}, c). \quad (4.5)$$

Portanto, a classe do objecto gerador das *features* X é dada pela classe com maior verosimilhança, considerando todos os valores de \mathbf{r} .

4.4 Computação

A classificação é feita pela maximização da função de verosimilhança $L(\mathbf{r}, c)$. Esta maximização é equivalente à maximização do logaritmo de $L(\mathbf{r}, c)$. É assim feita a maximização do logaritmo da verosimilhança (*log-likelihood*), que é dada por

$$\begin{aligned} \log L(\mathbf{r}, c) = l(\mathbf{r}, c) = & \sum_e \sum_{\mathbf{x} \in S_e(\mathbf{r})} X_e(\mathbf{x}) \log(p_e(\mathbf{x}; \mathbf{r})) + (1 - X_e(\mathbf{x})) \log(1 - p_e(\mathbf{x}; \mathbf{r})) \\ & + \sum_{\mathbf{x} \notin S_e(\mathbf{r})} X_e(\mathbf{x}) \log(p_{\text{bgd}}) + (1 - X_e(\mathbf{x})) \log(1 - p_{\text{bgd}}). \end{aligned} \quad (4.6)$$

A classificação é feita da mesma forma, apenas com o logaritmo da verosimilhança:

$$\hat{Y} = \arg \max_c \max_{\mathbf{r}} \log P_c(X|\mathbf{r}) = \arg \max_c \max_{\mathbf{r}} l(\mathbf{r}, c). \quad (4.7)$$

Esta maximização é feita calculando $l(\mathbf{r}, c)$ para cada classe c , e para cada valor de \mathbf{r} . Os valores de \mathbf{r} admitidos são todos os valores possíveis dentro de uma vizinhança limitada, cuja dimensão dependerá naturalmente da resolução da imagem.

Foi tomado um valor de *threshold* l_{\min} para a classificação. Quando um objecto analisado tem um valor máximo de $l(\mathbf{r}, c)$ em todos c e \mathbf{r} inferior a l_{\min} , o rectângulo ao qual pertence é considerado um rectângulo incorrectamente detectado, resolvendo-se assim o problema dos falsos positivos da fase de detecção de cartas, como foi demonstrado na figura 3.6.

Dado que a maximização da verosimilhança envolve a computação desta para vários valores de \mathbf{r} e para cada classe c , foi permitida uma detecção precoce. Foi para isso definido um outro valor de *threshold*, $l_{\text{pre}} > l_{\min}$. Iterando pelos valores de \mathbf{r} em cada classe c , caso $l(\mathbf{r}, c)$ seja igual ou maior que l_{pre} , considera-se o objecto como pertencente à classe c , sem se calcular $l(\mathbf{r}, c)$ para outros valores de \mathbf{r} ou c . Desta forma consegue-se uma classificação mais rápida, considerando que se encontrou uma classe com uma verosimilhança suficientemente alta para se considerar correcta. O facto de l_{pre} ser superior a l_{\min} faz com que seja procurada uma verosimilhança melhor, ainda que um dado $l(\mathbf{r}, c)$ seja suficientemente alto para não ser considerada uma falsa detecção.

Considerando N_C o número de classes e $N_{\mathbf{r}}$ o número de valores para o vector \mathbf{r} , e tendo uma imagem representada por 8 mapas de *edges*, o método de classificação pode ser descrito pelo seguinte algoritmo:

1. Para cada $c = 1, \dots, N_C$,
 Para cada $\mathbf{r}_i, i = 1, \dots, N_{\mathbf{r}}$, calcular $l(\mathbf{r}_i, c)$. Caso $l(\mathbf{r}_i, c) > l_{\text{pre}}$ classificar a imagem como pertencente à classe c e terminar a classificação. Caso contrário, continuar a percorrer os valores de c e i e prosseguir com o algoritmo.
2. Para cada $c = 1, \dots, N_C$, calcular o máximo em \mathbf{r}_i de $l(\mathbf{r}_i, c)$ e assinalar este valor $l(c)$ como a log-verosimilhança da classe c .

3. Encontrar o valor c cuja log-verosimilhança é máxima, denominando-o por c^* . Caso $l(c^*) > l_{\min}$, classificar a imagem como pertencente à classe c^* . Caso contrário, considerar a imagem uma falsa detecção.

4.5 Treino do modelo

O cálculo da verosimilhança implica o conhecimento do mapa de probabilidades $p_e(\mathbf{x})$. Para esse efeito, foi feito um treino para cada classe. Dado que a segmentação feita após a detecção de rectângulos tem como resultado o objecto a identificar isolado, sempre numa rotação muito próxima da correcta (i.e. a usual, como ilustrado na figura 1.2) e é assumido que não há qualquer tipo de deformação não rígida, o treino do modelo foi feito de forma bastante simples.

Para cada classe foram reunidas $N = 15$ imagens de exemplo, todas com o objecto na mesma localização (\mathbf{r} igual). Para cada uma são detectados os *edges* nos 8 sentidos, é feita a dilatação 3×3 , e em seguida, os mapas de cada orientação são somados, sendo depois divididos ponto a ponto por N . Desta forma os mapas de probabilidade $p_e(\mathbf{x})$ são simplesmente dados pela média das imagens de treino. A figura 4.4 demonstra os mapas de probabilidades para a classe do valor K (Rei) em quatro direcções. Em cima são mostrados os mapas calculados nos sentidos $(1, 0)$, $(1, 1)$, $(0, 1)$, $(-1, 1)$. Em baixo é mostrado o mesmo para os sentidos contrários aos de cima.

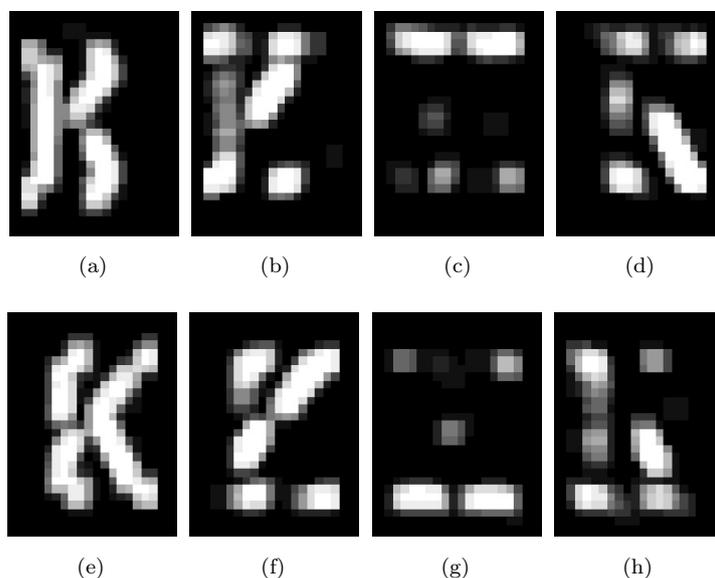


Figura 4.4: Mapas de probabilidades para a classe do valor K. Em cima, os mapas para as direcções \mathbf{v} : $(1, 0)$, $(1, 1)$, $(0, 1)$, $(-1, 1)$. Em baixo, os mapas para os sentidos contrários.

Como o treino não foi exaustivo, ao não ser usado como imagens de treino todos os exemplos possíveis dos objectos (e eventualmente não poderia ser), o modelo poderá não trazer resultados desejáveis. Estes resultados indesejáveis resultam de o modelo ser altamente penalizador para não correspondências únicas

entre o mapa de probabilidades e as *features* detectadas. Observando a expressão da verossimilhança $L(\mathbf{r})$, há termos que sozinhos podem trazer o valor da verossimilhança a 0. Isto acontece quando $p_e(\mathbf{x}; \mathbf{r}) = 1$ e $X_e(\mathbf{x}) = 0$. Estes valores fazem $[1 - p_e(\mathbf{x}; \mathbf{r})]^{1 - X_e(\mathbf{x})} = 0$, anulando $L(\mathbf{r})$ com apenas um ponto sem correspondência entre o mapa de probabilidades e as *features* detectadas. Esta anulação ocorre ainda que o resto da imagem tenha uma boa correspondência. O caso de $p_e(\mathbf{x}; \mathbf{r}) = 0$ e $X_e(\mathbf{x}) = 1$ não traz este problema pois um ponto \mathbf{x} tal que $p_e(\mathbf{x}; \mathbf{r}) = 0$ não pertencerá ao suporte, pelo que não dará nenhum termo nulo.

A fim de relaxar o modelo face a estes casos, é feita uma saturação ao mapa de probabilidades proveniente do treino, com um valor p_{\max} próximo mas nunca igual a 1. Desta forma $p_e(\mathbf{x}; \mathbf{r})$ nunca tomará o valor de 1, penalizando não correspondências entre o mapa e a imagem, mas sem anular a verossimilhança totalmente.

Ainda que o conjunto de treino seja reduzido, dada a pouca variabilidade das imagens a classificar, o modelo utilizado traz bons resultados.

Capítulo 5

Classificação com um Modelo Deformável

Para a fase de classificação das cartas, foi também implementado um método baseado num modelo deformável. Dado que o problema da classificação está colocado de forma bastante rígida, não sendo esperadas deformações não rígidas, não existe grande motivação para a implementação de métodos deformáveis. Por essa razão este método não foi analisado de forma exaustiva.

O método implementado é bastante semelhante ao método rígido descrito no capítulo 4, com o cálculo da verosimilhança feito de igual forma, mudando apenas os mapas de probabilidades usados no cálculo desta. Este modelo segue e baseia-se fortemente no trabalho realizado por Amit e Trouvé [3].

Assim como a classificação com o modelo rígido, este modelo deformável usa as mesmas *features* binárias orientadas, tal como descrito em 4.1. É também definido um mapa de probabilidades $p_e(\mathbf{x})$ de presença das *features*, como descrito em 4.2.

5.1 Modelo deformável

Para ter em conta deformações dos objectos, a pose θ de um objecto já não pode ser definida por apenas um vector $\mathbf{r} = (r_x, r_y)$ de localização do objecto, mas também por um conjunto de n deslocamentos \mathbf{v} que descrevem a deformação. São considerados n pontos de referência $(\mathbf{y}_i)_{i=1,\dots,n}$, e a pose é definida por $\theta = (\mathbf{r}, \mathbf{v}) = (\mathbf{r}, \mathbf{v}_1, \dots, \mathbf{v}_n)$. Cada \mathbf{y}_i é mapeado para $\mathbf{z}_i = \mathbf{r} + \mathbf{y}_i + \mathbf{v}_i$, em que o vector $\mathbf{r} + \mathbf{v}_i$ representa um deslocamento rígido do modelo, sendo assim representada a deformação do objecto pelo deslocamento de cada ponto de referência \mathbf{y}_i . Se todos os \mathbf{v}_i forem iguais, a pose pode ser descrita por apenas um deslocamento rígido \mathbf{r} . Para que se possam considerar diferentes desvios \mathbf{v}_i , os diferentes modelos deslocados são combinados através de uma média, realizada por *partes*.

Define-se uma *parte* Q_i associada ao ponto de referência \mathbf{y}_i como um submapa de p_e numa vizinhança

quadrada W em torno de \mathbf{y}_i :

$$Q_i \doteq p_e(\mathbf{y}_i + s), \quad s \in W, \quad e = 1, \dots, E. \quad (5.1)$$

O mapa $p_e(\mathbf{x}; \theta)$ já não pode ser dado por um simples deslocamento de $p_e(\mathbf{x})$ como foi em 4.2. As *partes* são combinadas através de uma operação chamada de *patchwork of parts* (POP). Seja $N(\mathbf{x}) = \{i : \mathbf{x} \in \mathbf{z}_i + W\}$ o conjunto das janelas W deslocadas que cobrem o ponto \mathbf{x} . A operação POP é usada para calcular o mapa $p_e(\mathbf{x}; \theta)$, da seguinte forma:

$$p_e(\mathbf{x}; \theta) = P(X_e(\mathbf{x}) = 1 | \theta) = \begin{cases} \frac{1}{|N(\mathbf{x})|} \sum_{i \in N(\mathbf{x})} p_e(\mathbf{x} - \mathbf{z}_i + \mathbf{y}_i) & \text{se } N(\mathbf{x}) \neq \emptyset \\ 0 & \text{se } N(\mathbf{x}) = \emptyset \end{cases} \quad (5.2)$$

O mapa $p_e(\mathbf{x}; \theta)$ é portanto dado pela combinação das diferentes *partes*, sendo feita a média aritmética nas regiões pertencentes a mais de uma *parte*.

Dado que o objecto está presente na imagem com a pose θ , é novamente assumida a independência condicional das *features* X , e é definido um suporte do objecto $S(\theta)$ e um modelo de fundo uniforme, como em 4.2:

$$S_e(\theta) = \{\mathbf{x} : p_e(\mathbf{x}; \theta) \geq p_{\min}\}. \quad (5.3)$$

Tendo os mapas de probabilidades $p_e(\mathbf{x}; \theta)$ e os suportes $S_e(\theta)$ definidos, a verosimilhança dos dados das *features* X , dada a pose θ , é dada por

$$L(\theta) = P(X | \theta) = \prod_e \prod_{\mathbf{x} \in S_e(\theta)} [p_e(\mathbf{x}; \theta)]^{X_e(\mathbf{x})} [1 - p_e(\mathbf{x}; \theta)]^{1 - X_e(\mathbf{x})} \prod_e \prod_{\mathbf{x} \notin S_e(\theta)} [p_{\text{bgd}}]^{X_e(\mathbf{x})} [1 - p_{\text{bgd}}]^{1 - X_e(\mathbf{x})}. \quad (5.4)$$

5.2 Classificação

A classificação é feita de forma análoga à classificação com o modelo rígido. Foi usado o classificador de máxima verosimilhança, ainda que neste caso haja motivação para a introdução de um *prior* que penalize os desvios \mathbf{v}_i , o que resultaria num classificador MAP. Permitindo apenas pequenos desvios a penalização por um *prior* torna-se pouco influente. A classificação por máxima verosimilhança é mais simples, não sendo necessária uma estimação das distribuições de \mathbf{v}_i , que poderia necessitar de um treino exaustivo a fim de ser considerada uma gama de deformações potencialmente extensa.

Para este caso deformável, a classificação não será baseada numa maximização em \mathbf{r} , mas também nos desvios \mathbf{v}_i . Considerando $P_c(X | \theta) \doteq L(\theta, c)$ a verosimilhança dos dados X corresponderem à classe $c \in C$, dada uma pose θ , a classificação é dada pela maximização

$$\hat{Y} = \arg \max_c \max_{\mathbf{r}} \max_{\mathbf{v}} P_c(X | \theta = (\mathbf{r}, \mathbf{v})) = \arg \max_c \max_{\mathbf{r}} \max_{\mathbf{v}} L(\theta, c). \quad (5.5)$$

Assim a classe do objecto gerador das *features* X é dada pela classe com maior verosimilhança, considerando todos os valores de \mathbf{r} e dos vectores \mathbf{v}_i

5.3 Computação

A maximização da verosimilhança envolve um espaço de valores de \mathbf{v} demasiado grande para ser realizada por força bruta. Para que esta maximização seja feita de forma eficiente, foi usado um de dois métodos propostos em [3], chamado de *Independent Maximization* - Maximização Independente. Esta maximização em \mathbf{v} é realizada para cada valor de \mathbf{r} .

Inicializam-se $\mathbf{v}_i^{(0)} = (0, 0)$, $i = 1, \dots, n$. Escolhe-se uma pequena vizinhança quadrada V em torno da origem. Na iteração t são percorridos os diferentes pontos de referência \mathbf{y}_i . Para cada vector $v \in \mathbf{v}_i^{(t-1)} + V$ é colocada a *parte* Q_i em $\mathbf{y}_i + v$ e é calculada a verosimilhança desta, ignorando todas as outras. Ao vector $\mathbf{v}_i^{(t)}$ é dado o valor de v cuja verosimilhança é máxima. Depois de todos os $\mathbf{v}_i^{(t)}$ serem actualizados, podem ser feitas mais iterações. Por outras palavras, o desvio associado a cada ponto de referência é aquele que maximiza a verosimilhança da *parte* a si associada.

As maximizações da verosimilhança são feitas através do logaritmo desta, tal como em 4.4. A função de log-verosimilhança é assim dada por

$$\begin{aligned} \log L(\theta, c) = l(\theta, c) = & \sum_e \sum_{\mathbf{x} \in S_e(\theta)} X_e(\mathbf{x}) \log(p_e(\mathbf{x}; \theta)) + (1 - X_e(\mathbf{x})) \log(1 - p_e(\mathbf{x}; \theta)) \\ & + \sum_{\mathbf{x} \notin S_e(\theta)} X_e(\mathbf{x}) \log(p_{\text{bgd}}) + (1 - X_e(\mathbf{x})) \log(1 - p_{\text{bgd}}). \end{aligned} \quad (5.6)$$

A classificação é feita pela maximização da log-verosimilhança

$$\hat{Y} = \arg \max_c \max_{\mathbf{r}} \max_{\mathbf{v}} l(\theta, c) \quad (5.7)$$

usando o método da Maximização Independente para a maximização em \mathbf{v} . A classe c escolhida como maximizante da verosimilhança é considerada a classe do objecto caso o logaritmo da sua verosimilhança $l(\theta, c)$ seja superior a um dado *threshold* l_{min} . Caso contrário, é considerada uma falsa detecção. Para tornar a classificação mais rápida, foi novamente implementado um método de detecção precoce, como já tinha sido feito em 4.4.

A figura 5.1 demonstra o método para classificação. A imagem a classificar (figura 5.1b) resulta de uma deformação afim, do tipo *skew* e com uma pequena alteração de escala, de uma imagem de um “Ás”. Na figura 5.1d está presente o mapa de probabilidades para uma orientação horizontal, sem deformações. Na figura 5.1e está presente o mapa de probabilidades a usar para a classificação, após o cálculo dos desvios que melhor aproximam o mapa original das *features* detectadas (figura 5.1c). A figura 5.1f mostra os vários desvios calculados e a forma como as *partes* ficaram sobrepostas. Ainda que hajam partes cujo desvio não aparenta ser o esperado, no geral os desvios calculados são consistentes com a deformação observada, com algumas *partes* desviadas para a direita para compensar o *skew* e para baixo para compensar a alteração de escala.

Dado o método para o cálculo da deformação, pode-se ver que o nível de deformação permitido é dependente da dimensão da vizinhança V para os desvios. Caso se permita apenas desvios pequenos,

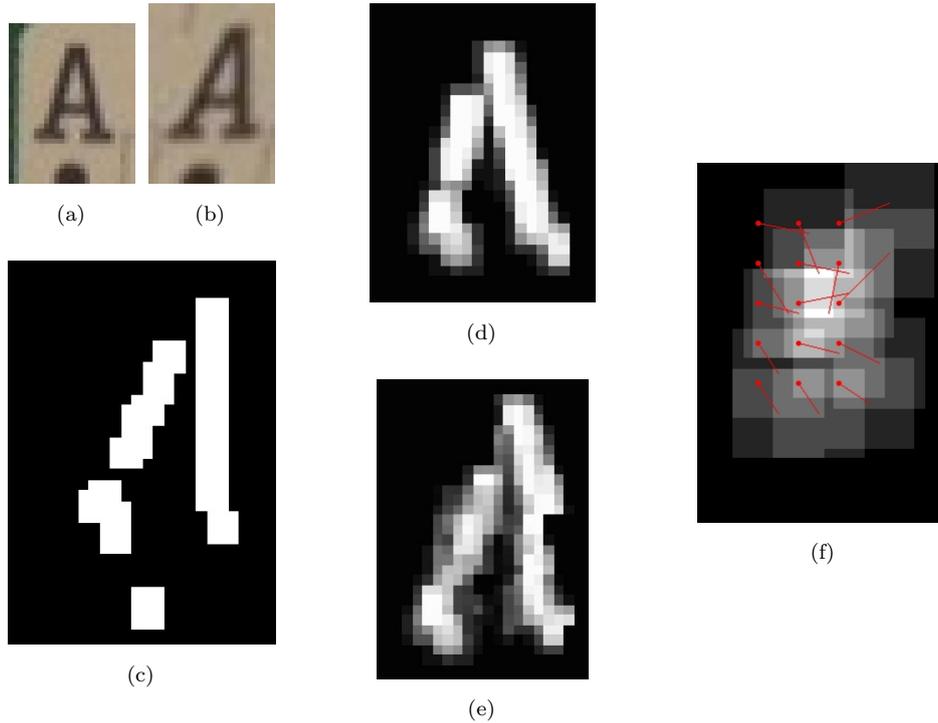


Figura 5.1: Classificação com modelo deformável. (a) Imagem original, sem deformações. (b) Imagem a classificar, resultante de uma distorção afim de (a). (c) As *features* detectadas na imagem (b), para uma orientação horizontal. (d) Mapa de probabilidades para a mesma orientação, sem deformações. (e) Mapa de probabilidades após serem estimados os desvios de cada parte e pós a operação de *patchwork*. (f) Os desvios estimados de cada parte e a forma como ficam distribuídas as partes. Os pontos representam os pontos de referência \mathbf{y}_i e as linhas os desvios \mathbf{v}_i . As regiões mais claras representam regiões com maior número de partes sobrepostas.

uma pequena deformação pode ser bem estimada, mas uma deformação mais severa pode-se tornar impossível de estimar, e portanto de se classificar convenientemente. Permitindo desvios maiores, maiores deformações podem ser admitidas, mas também pode permitir deformar um modelo de uma classe de tal forma que este se aproxime bem de uma imagem pertencente a outra classe. Nestes casos, uma classificação usando um *prior* penalizador dos desvios torna-se especialmente importante.

5.4 Treino do Modelo

O treino do modelo deformável é bastante semelhante ao do modelo rígido, cujo mapa de probabilidades do modelo $p_e(\mathbf{x})$ é simplesmente a média dos mapas de *features* de cada imagem de treino. Para este modelo deformável, sendo baseado em *partes*, é feito o mesmo tipo de treino para cada parte, de forma independente das outras.

Definem-se os pontos de referência \mathbf{y}_i , $i = 1, \dots, n$ numa grelha regular, aos quais estarão associadas cada *parte* Q_i . Cada *parte* consiste num mapa de probabilidades, sendo inicializado a zeros. Esta lista contém todas as *partes*, ainda que possam estar associadas a diferentes orientações e das *features*. Para cada imagem de treino j são detectadas as *edge features* e de seguida é feita uma dilatação 3×3 , sendo guardados os resultados nos mapas $p_e^{(j)}(\mathbf{x})$. Para cada *parte* Q_i associada à orientação e , é copiada a região de $p_e^{(j)}(\mathbf{x})$ a si correspondente (a região $\mathbf{y}_i + W$) e somada a Q_i . Depois de percorrer todas as *partes*, divide-se cada uma pelo número de imagens de treino, obtendo-se assim a média em cada uma. Todas as partes que tiverem uma variância inferior a um dado valor (um mapa de probabilidades muito homogéneo) são rejeitadas. Assim rejeitam-se as *partes* sem informação suficientemente relevante, como por exemplo *partes* que apenas incluem regiões de fundo.

A figura 5.2 mostra o treino de uma *parte*. À esquerda estão presentes os mapas de *edges* das imagens de treino, na região da *parte* a ser treinada, e o mapa final desta, calculado pela média das imagens de treino. À direita está o mapa $p_e(\mathbf{x})$, após a operação de *patchwork of parts*, com a região da *parte* treinada evidenciada a vermelho.

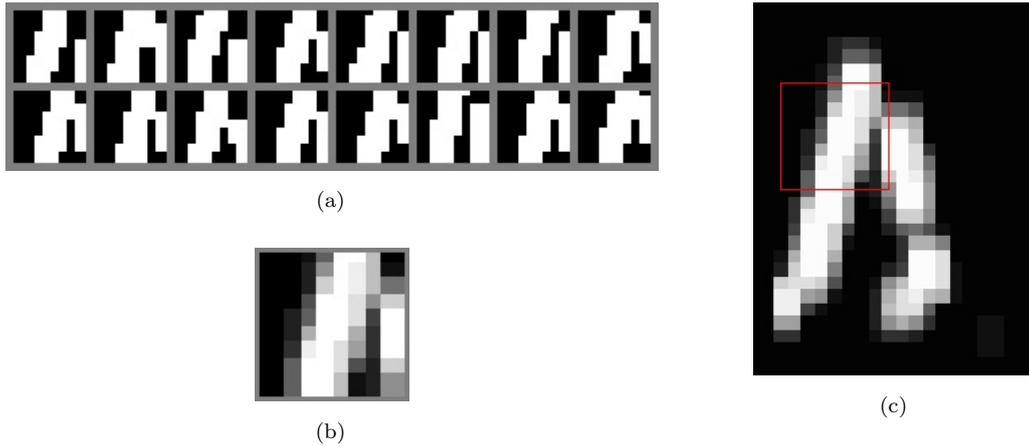


Figura 5.2: Treino do modelo POP. (a) Os mapas de *edges* para uma orientação horizontal nas imagens de treino, na região correspondente a uma *parte*. (b) A mesma *parte* treinada, pela média das imagens em (a). (c) O mapa de probabilidades $p_e(\mathbf{x})$ após a operação de *patchwork of parts*, com a região correspondente à parte treinada evidenciada a vermelho.

Em [3] é sugerido um treino que para além do mapa de probabilidades de cada *parte*, procura também estimar os desvios \mathbf{v}_i . Esse treino é realizado por um método EM (Expectation-Maximization), mas não foi implementado neste trabalho.

Capítulo 6

Classificação por *Template Matching*

Os métodos de *template matching* consistem numa comparação de uma imagem de modelo com uma outra imagem a analisar, usando tipicamente uma métrica como a diferença quadrática ou correlação cruzada (*cross-correlation*). Esta comparação é feita deslizando o *template* (modelo) pela imagem a analisar, e a cada posição atribuir um valor, dado pela métrica usada. Este método corresponde a uma procura de um modelo numa imagem, e tipicamente tem como resultado um mapa 2D, com dimensão igual à dimensão dos deslocamentos que o modelo faz sobre a imagem, com os valores calculados na procura. A posição do mapa resultante com valor mínimo, no caso da métrica de diferença quadrática, ou com valor máximo, para a métrica de correlação cruzada, corresponde à posição onde o modelo melhor se ajusta (*matching*).

Considerando tanto o modelo como a imagem com o objecto a classificar de pequenas dimensões, o *template matching* envolve a soma de algumas dezenas de multiplicações, num número pequeno de posições, sendo assim realizado de forma bastante rápida.

Ainda que este método seja sensível a deformações ou rotações do objecto face ao modelo, visto que a comparação é feita de forma rígida, o facto da detecção prévia das cartas admitir uma escala fixa e sem deformações, realizando também uma segmentação dos valores e naipes das cartas eficaz e na rotação correcta faz do *template matching* uma escolha natural para a classificação das cartas, pela sua rapidez e simplicidade.

6.1 Implementação

Para a classificação por *template matching* foi usada uma métrica de correlação cruzada normalizada. Seja $T_c(\mathbf{x})$ um modelo e $I(\mathbf{x})$ a imagem com o objecto a classificar, com valores em 2D indexados por \mathbf{x} . Neste trabalho $I(\mathbf{x})$ representa a imagem do valor ou do naipe da carta, e o modelo $T_c(\mathbf{x})$ representa

uma imagem de modelo da classe c , que pode ser um dos valores ou um dos naipes. Como explicado em 4.3, o objecto em $I(\mathbf{x})$ pode não estar na posição certa, sendo por isso necessária a procura da posição onde o modelo melhor se ajusta. Seja também $R_c(\mathbf{x})$ o mapa de correlação para a classe c , cujo máximo corresponderá à posição em que o modelo melhor se ajusta, e o seu valor corresponde ao valor de ajuste da classe c .

A correlação cruzada normalizada entre o modelo da classe c e a imagem a analisar é dada por

$$R_c(\mathbf{x}) = \frac{\sum_{\mathbf{x}'} T_c(\mathbf{x}') \cdot I(\mathbf{x} + \mathbf{x}')}{\sqrt{\sum_{\mathbf{x}'} T_c(\mathbf{x}')^2 \cdot I(\mathbf{x} + \mathbf{x}')^2}}. \quad (6.1)$$

O numerador da expressão corresponde à correlação cruzada usual, enquanto que o denominador é apenas um termo de normalização, a fim do resultado tomar valores apenas entre 0 e 1.

6.2 Classificação

A classificação da carta é feita através da classificação da imagem do valor e do naipe, como mencionado anteriormente. Cada imagem a classificar é comparada com o modelo de cada classe de forma sequencial, sendo escolhida a classe com maior valor de correlação. Considerando que um valor ou naipe está presente na imagem a classificar, a classificação é dada pela maximização

$$\hat{Y} = \arg \max_c \max_{\mathbf{x}} R_c(\mathbf{x}). \quad (6.2)$$

No entanto, como não é garantida a presença de um valor ou naipe na imagem por ser possível uma falsa detecção da carta, uma carta é apenas classificada se a classe que melhor se ajusta à imagem do valor tiver um valor de correlação superior a um valor de *threshold* R_{\min} . Caso isto não se verifique, a carta analisada é considerada uma falsa detecção.

6.3 Treino do Modelo

Para que a classificação seja feita com sucesso, o modelo deve ser treinado de modo a que pequenas individualidades das imagens de treino não se reflectam no modelo (e.g., ruído). Assim, uma forma simples e eficaz de treinar o modelo é pela média dos valores de intensidade das várias imagens de treino. A figura 6.1 mostra um exemplo de uma imagem de treino e uma imagem do modelo treinado, para um valor de carta e para um naipe. Calculando a média das várias imagens de treino, o ruído de cada uma das imagens é diminuído, obtendo-se uma imagem de modelo consideravelmente uniforme nas regiões onde assim seria esperado.

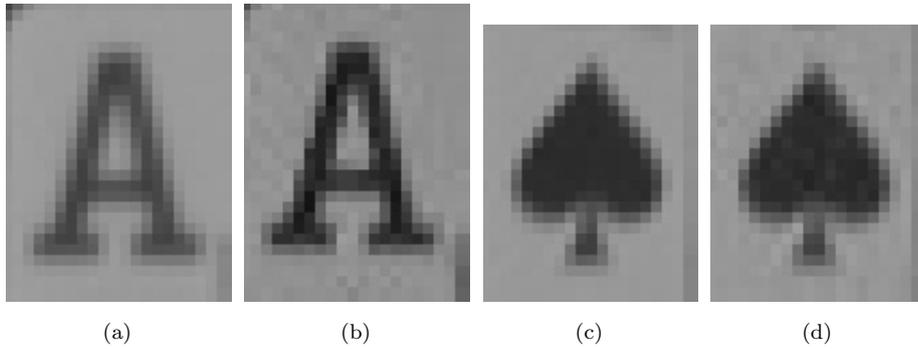


Figura 6.1: Treino dos modelos para *template matching*. (a) O modelo treinado para o valor “Ás”. (b) Uma imagem de treino para o valor “Ás”. (c)(d) O mesmo para o naipe “Espadas”.

Capítulo 7

Detecção e Reconhecimento em Perspectiva

Para demonstrar que a eficácia do método proposto não é totalmente dependente da posição da câmera face à mesa, este foi testado em condições de posição diferentes. Para não haver deformações de perspectiva numa imagem de cartas, a câmera deve estar colocada de forma perfeitamente perpendicular face ao plano onde as cartas são colocadas. A câmera deve também estar colocada a uma distância potencialmente grande do plano e com uma lente cuja distância focal seja também grande, a fim de evitar deformações ópticas.

Para este efeito, foram capturadas imagens de forma oblíqua ao plano das cartas, a uma distância curta (cerca de 1 metro). As imagens para teste foram adquiridas pelas câmeras dos olhos do robô *iCub*[14], com uma correcção de deformações ópticas realizada previamente. Foram também capturadas imagens por uma câmera digital convencional. Uma imagem adquirida desta forma apresenta assim uma deformação de perspectiva, que será corrigida para se proceder à detecção das cartas e de seguida à sua classificação. A base da geometria projectiva, usada nesta correcção, será apresentada de seguida.

7.1 Modelo da Câmera

O modelo usado para a forma como as imagens são adquiridas por uma câmera é o modelo da câmera *pinhole*. Uma câmera *pinhole* consiste numa parede com um furo (*pinhole*) que apenas permite a passagem dos raios de luz que por ele passem. Os raios são então projectados do outro lado da parede (o plano da câmera), e apenas um raio de luz entra por cada ponto no exterior. Trata-se de um modelo simples, que abstrai os efeitos de deformação inerentes ao uso de lentes numa câmera usual e torna a imagem projectada sempre focada, com um tamanho apenas dado pelo parâmetro da distância focal f , que neste modelo é a distância entre o furo e o plano de imagem. Seja (X, Y, Z) as coordenadas de um ponto P no mundo físico e (x, y) as coordenadas do ponto p projectado no plano da imagem. Pela

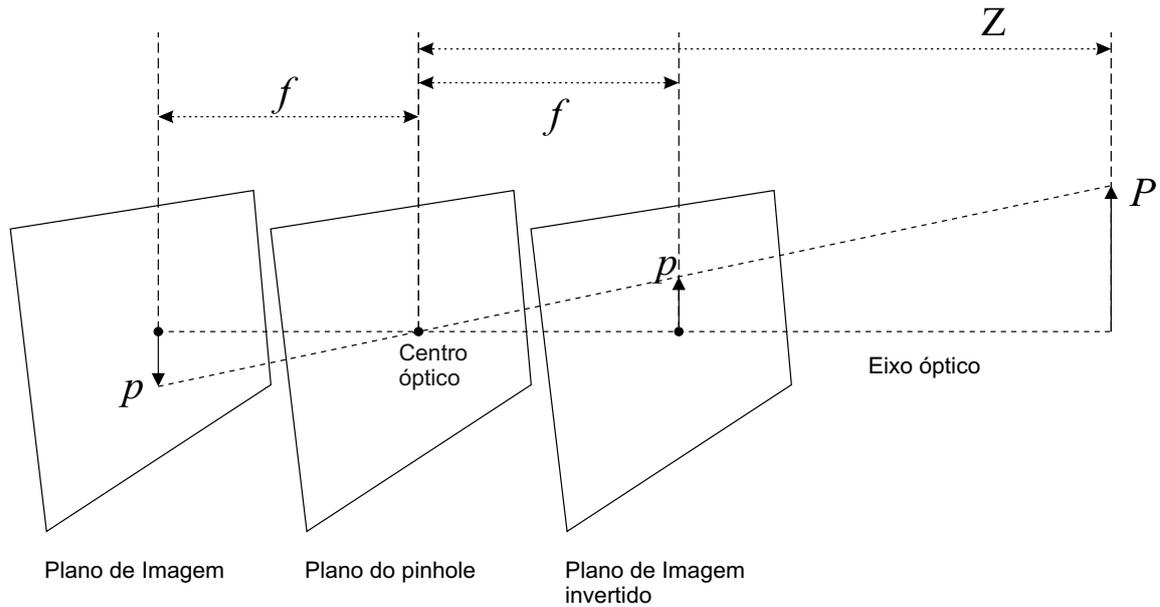


Figura 7.1: Modelo da câmera *pinhole*. O objecto no ponto P do mundo físico é projectado em p no plano de imagem pelo raio que passa no *pinhole*. A matemática é equivalente com plano de imagem ou com o plano de imagem invertido, sendo com este último mais simples. Ambos os planos de imagem estão à mesma distância f do plano do *pinhole*, que está a uma distância Z do objecto.

figura 7.1 pode-se verificar que, por exemplo para a coordenada x de p , $-x/f = X/Z$, e portanto

$$-x = f \frac{X}{Z}, \quad -y = f \frac{Y}{Z}. \quad (7.1)$$

Também pela figura 7.1 é evidente que desta forma as imagens adquiridas ficam invertidas. Para simplificar o modelo e com total equivalência, pode-se inverter o plano da imagem, colocando-o entre o *pinhole* e o objecto. Neste plano invertido, o ponto P é projectado em p de acordo com

$$x = f_x \frac{X}{Z} + c_x, \quad y = f_y \frac{Y}{Z} + c_y. \quad (7.2)$$

Desta vez usaram-se diferentes distâncias focais para x e y , o que permite modelar um sensor de imagem cujos elementos são rectangulares, em vez de quadrados. Foram também adicionados os termos c_x e c_y , que funcionam como valores de translação face ao centro da imagem, que pode não corresponder ao centro óptico (ponto de intersecção com o eixo óptico).

7.2 Transformação Projectiva

O mapeamento de um ponto P no mundo físico com coordenadas (X, Y, Z) para o ponto p no plano de imagem com coordenadas (x, y) é chamada de transformação projectiva. As equações desta transformação tornam-se mais simples se os pontos P, p forem representados por coordenadas

homogéneas, respectivamente definidas por

$$Q = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad q = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (7.3)$$

Um ponto em coordenadas homogéneas é obtido através da multiplicação por um factor de escala, em que o factor de escala é a coordenada adicional. Assim, dois pontos em coordenadas homogéneas que sejam proporcionais entre si representam o mesmo ponto no espaço físico ou no plano de imagem, e o seu valor real pode ser obtido dividindo pelo valor da coordenada adicional. Usando coordenadas homogéneas para os pontos da imagem, a projecção dos pontos do mundo físico no plano de imagem pode ser descrito de forma matricial:

$$sq = \mathbf{M} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (7.4)$$

em que s é o factor de escala de q , que neste caso é igual a Z . A matriz \mathbf{M} é chamada de matriz dos parâmetros intrínsecos da câmara. Estes parâmetros são fixos para uma dada câmara, pois apenas depende da construção desta.

Até este ponto foi considerado que o referencial do mundo físico, em 3D, está localizado no centro óptico da câmara. Dado que é difícil medir coordenadas no mundo físico em ordem a este referencial, é considerado um outro referencial arbitrário que será necessariamente relacionado com o anterior por uma transformação rígida (rotação e translação): Seja P_0 um ponto no referencial escolhido. Neste referencial, as coordenadas de um ponto P definido no referencial da câmara serão dadas por

$$P = \mathbf{R}P_0 + T \quad (7.5)$$

em que \mathbf{R} é uma matriz 3×3 de rotação e T um vector 3D de translação. Os parâmetros de \mathbf{R} e T são chamados de parâmetros extrínsecos da câmara, pois dependem da localização e orientação desta. Esta transformação rígida pode ser escrita usando coordenadas homogéneas da seguinte forma:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \mathbf{R} & T \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \quad (7.6)$$

Tendo explicitado a fórmula para mapear um ponto 3D de um referencial arbitrário para o referencial da câmara e a fórmula para projectar no plano de imagem um ponto 3D no referencial da câmara, estas podem ser combinadas numa só. Assim, usando coordenadas homogéneas, a projecção de um ponto num referencial arbitrário do mundo físico no plano de imagem é dada por

$$sq = \mathbf{M} \begin{bmatrix} \mathbf{R} & T \end{bmatrix} Q, \quad (7.7)$$

em que o factor de escala s explicita o facto de que esta projecção é calculada à parte de um factor de escala. As coordenadas reais (x, y) da projecção são dadas pelas duas primeiras coordenadas do vector q , que deverá ter a terceira coordenada de valor igual a 1.

7.3 Homografia

Em visão por computador, chama-se de homografia planar uma transformação projectiva que mapeia um plano para outro. Neste trabalho procura-se mapear o plano onde estão localizadas as cartas para o plano da imagem.

Usando a fórmula matricial da projecção, temos $sq = \mathbf{M} \begin{bmatrix} \mathbf{R} & T \end{bmatrix} Q$. No entanto, visto que não é preciso calcular a projecção de pontos em todo o espaço físico, mas apenas os pertencentes a um plano, podemos definir um referencial tal que este corresponda ao plano dado por $Z = 0$, sem qualquer perda de generalidade. Usando esta simplificação, e dividindo a matriz de rotação \mathbf{R} em três colunas 3×1 ($\mathbf{R} = [R_1 \ R_2 \ R_3]$), a terceira coluna não será necessária, por ser anulada com $Z = 0$:

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} R_1 & R_2 & R_3 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} R_1 & R_2 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \quad (7.8)$$

Como $\mathbf{M}[R_1 \ R_2 \ T]$ é um produto de duas matrizes 3×3 , temos a homografia descrita por uma matriz $\mathbf{H} = \mathbf{M}[R_1 \ R_2 \ T]$ também 3×3 . Esta matriz de homografia descreve completamente a projecção do plano físico no plano da imagem, sendo esta projecção dada por

$$sq = \mathbf{H}Q', \quad Q' = \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \quad (7.9)$$

A homografia é dada à parte de um factor de escala, portanto mesmo que a matriz \mathbf{H} seja de dimensões 3×3 é totalmente descrita por apenas oito parâmetros. Dado que cada par de pontos (q, Q') define duas equações (em x e em y), a matriz de homografia pode ser estimada usando pelo menos quatro pares de pontos, estabelecendo uma correspondência entre a localização (conhecida) dos pontos no plano físico com a localização desejada no plano de imagem. Caso sejam usados mais pontos, o problema não tem uma solução exacta, pelo que é estimada a melhor matriz de acordo com algum critério, tipicamente de mínimos quadrados. O cálculo da matriz de homografia foi feito através da função *findHomography* da biblioteca OpenCV. Esta função estima a matriz de homografia através de um método não descrito na documentação, que minimiza a soma dos quadrados das distâncias euclidianas entre os pontos dados e os pontos estimados pela projecção.

7.4 Calibração da imagem

Neste trabalho a homografia foi estimada usando apenas quatro pares pontos. Estes pontos foram escolhidos de forma automática. É adquirida uma imagem de uma folha de papel de tamanho conhecido, de cor contrastante ao fundo, e são detectados os quatro cantos mais significativos, que deverão corresponder aos cantos da folha.

Os cantos são detectados pelo seguinte método: para cada pixel da imagem I é considerada uma pequena vizinhança quadrada S em torno deste. É calculada a matriz de covariação de derivadas na vizinhança:

$$C = \begin{bmatrix} \sum_S \left(\frac{dI}{dx}\right)^2 & \sum_S \left(\frac{dI}{dx} \frac{dI}{dy}\right)^2 \\ \sum_S \left(\frac{dI}{dx} \frac{dI}{dy}\right)^2 & \sum_S \left(\frac{dI}{dy}\right)^2 \end{bmatrix} \quad (7.10)$$

em que as derivadas são calculadas pelo operador de Sobel. São então calculados os valores próprios de C e para cada pixel é guardado o mínimo valor próprio, em que quanto menor, mais significativo (i.e. mais acentuado) será o canto nesse pixel. Os quatro cantos serão os quatro mais significativos.

Tendo a folha de papel dimensões conhecidas, a distância entre cada ponto é também conhecida, pelo que a correspondência com o plano da imagem é feita considerando que as localizações destes pontos na imagem devem cumprir a relação entre as distâncias no mundo físico. Estas localizações são escolhidas de forma a que as cartas na imagem, após correcção de perspectiva, tenham um tamanho compatível com o tamanho dos modelos usados na classificação das cartas. É calculada a matriz de homografia, e depois disto as imagens a analisar são corrigidas pela homografia calculada. Após a correcção, a imagem pode ser analisada como foi descrito nos capítulos anteriores, visto que as cartas já não apresentarão deformações e estarão no tamanho correcto.

Capítulo 8

Resultados

O método proposto está dividido em duas partes principais: a detecção e o reconhecimento. A fase de reconhecimento foi implementada de várias formas diferentes, pelo que serão mostrados os resultados de cada uma, assim como da detecção. Serão comparados os diferentes métodos de reconhecimento, avaliando a sua robustez a alterações de luminosidade e contraste, assim como o tempo computacional. Serão também mostrados os resultados do método completo tanto nas imagens originalmente sem perspectiva como nas corrigidas por homografia, pela integração da detecção com o reconhecimento, pois o último está dependente do sucesso do primeiro.

Todas as imagens foram redimensionadas de forma a que as cartas tenham as dimensões de acordo com as dimensões dos modelos de reconhecimento treinados. As imagens dos modelos dos valores das cartas têm as dimensões de 21×28 pixels, e os modelos dos naipes têm as dimensões de 20×26 .

A classificação das cartas nos exemplos apresentados é mostrada através de um rectângulo branco colocado no centro da carta, e dentro deste está escrito o valor reconhecido (2,...10,A,J,Q,K) e o naipe, pelo código correspondente às iniciais em inglês: C - Paus (Clubs), S - Espadas (Spades), H - Copas (Hearts), D - Ouros (Diamonds).

8.1 Detecção de cartas

A detecção de cartas, no geral, funciona com sucesso. Quando as dimensões das cartas são modeladas correctamente não resultam falsos positivos na detecção. Uma diferença de 2 pixels na largura ou altura pode afectar a detecção, tornando-a mais difícil em casos de cartas em posições menos favoráveis (p.e. maior oclusão), mas as falhas de detecção por este tipo de variação de dimensões é rara (cerca de 1% de erros de detecção), podendo-se afirmar que o método permite variações mínimas em torno da dimensão correcta.

Dado que os falsos positivos poderão ser rejeitados na fase de classificação, a sua presença não é inconveniente. No entanto, uma grande quantidade de falsos positivos levará a um maior tempo de

classificação global. A quantidade de falsos positivos depende grandemente da disposição das cartas na imagem. A figura 8.1 exemplifica duas disposições diferentes, com as cartas correctamente classificadas a vermelho e as cartas detectadas mas rejeitadas na classificação a verde. Numa disposição que possibilite a presença de vários rectângulos contidos nos contornos exteriores, como na figura 8.1b, o número de falsos positivos pode ser elevado, enquanto que numa disposição que impossibilite a presença de tais rectângulos, como na figura 8.1a, o número de falsos positivos será menor ou mesmo nulo.

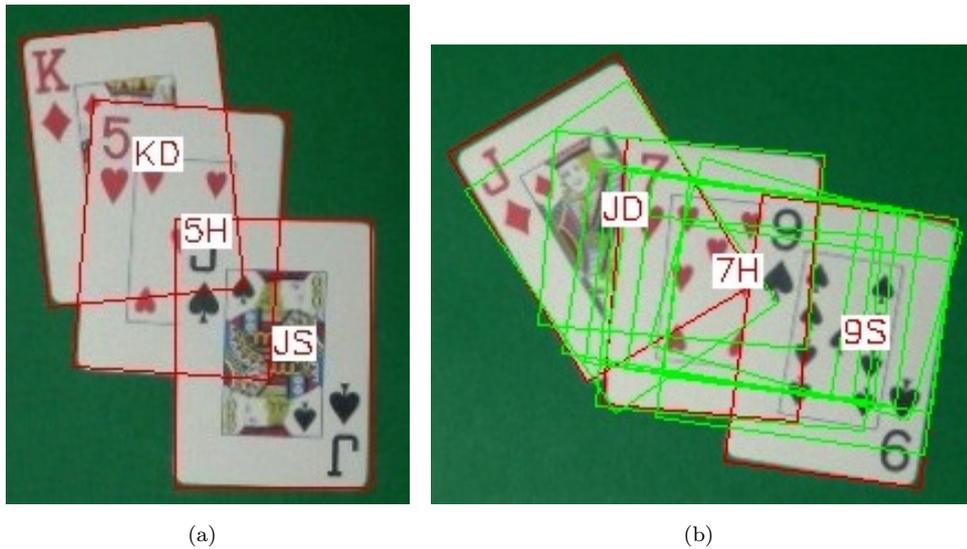


Figura 8.1: Detecção de cartas em oclusão. Os rectângulos a vermelho representam as cartas detectadas e correctamente classificadas. A verde estão representados os rectângulos detectados mas rejeitados na fase de classificação. (a) Disposição que não permite a presença de rectângulos do tamanho das cartas no interior dos contornos. (b) Disposição que permite a presença de vários falsos rectângulos.

A figura 8.2 mostra uma detecção com oclusão, em que a carta central (3 de Espadas) apenas tem um canto visível, para além do canto partilhado pelas outras duas cartas. O método de detecção consegue assim detectar com sucesso mesmo quando apenas uma pequena parte do contorno de uma carta está visível, ainda que possam ocorrer falsos positivos, como foi o caso na figura 8.2.

Conforme explicado em 3.5, foram usados três métodos para reduzir o número de falsos positivos: rejeição de rectângulos muito próximos de um mais forte (i.e. com mais votos), a rejeição de rectângulos cujo interior contenha contornos de cartas e a rejeição de rectângulos que não tenham pelo menos um canto coincidente com um canto de carta. A figura 3.5 mostra os casos que se procuram rejeitar com estes três passos.

O resultado da rejeição por proximidade está mostrado na figura 8.3, sendo comparado o resultado da detecção com a utilização dos três métodos de rejeição activos, com o resultado sem o primeiro método activo. Sem esta rejeição seria considerado um elevado número de rectângulos em posições equivalentes, o que levaria a que uma mesma carta fosse reconhecida diversas vezes.

O segundo passo de rejeição de falsos positivos mostrou-se de elevada importância. O facto de

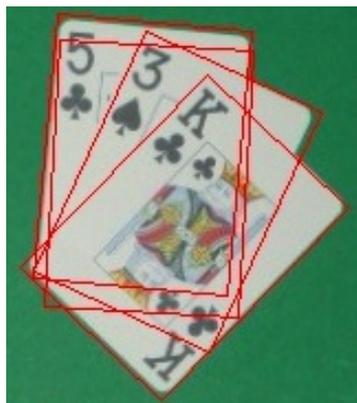


Figura 8.2: Detecção de cartas em oclusão, com apenas uma pequena parte do contorno do 3 de Espadas visível.

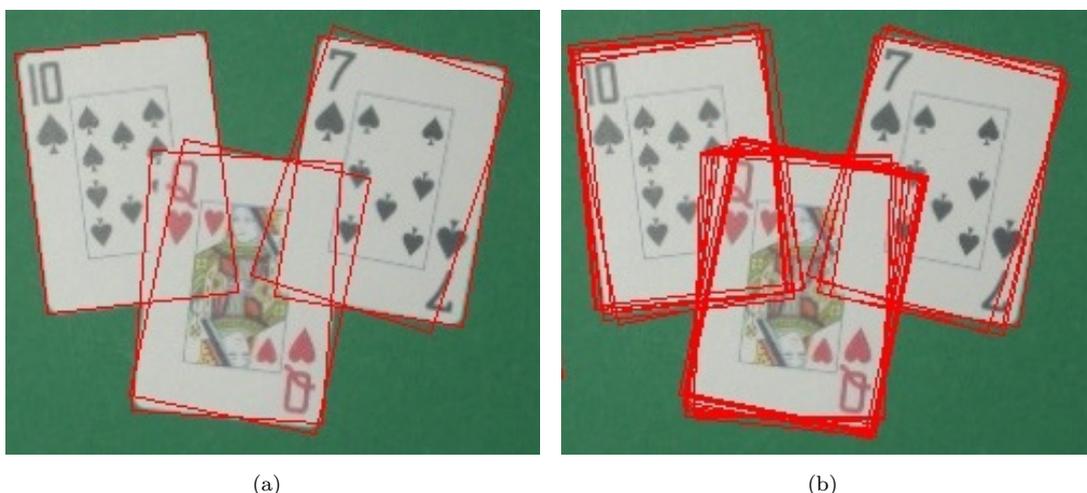


Figura 8.3: Detecção com rejeição por proximidade. (a) Rectângulos detectados com os três métodos de rejeição activos. (b) Rectângulos detectados sem a rejeição por proximidade.

ser usado um baixo *threshold* para a detecção de máximos locais do acumulador da transformada de Hough, a fim de se poder detectar cartas com apenas uma pequena parte do contorno visível, origina um elevado número de falsos positivos. Sem este segundo passo de rejeição, o número de rectângulos detectados é extremamente elevado, fazendo com que o resultado visível tenha pouca informação, visto que o desenho dos rectângulos detectados preenche largamente a imagem das cartas.

Um exemplo do resultado do terceiro passo de rejeição de falsos positivos pode ser observado na figura 8.4. Ainda que esta rejeição seja pouco influente, sendo até nula em muitos casos, permite remover alguns rectângulos, em especial quando há uma disposição potenciadora de ocorrência de falsos positivos. A sua implementação simples, como descrita em 3.5, é responsável pelo facto de o método não ser muito influente.

Um exemplo típico do problema relacionado com as oclusões horizontais está presente na figura

8.5. Neste exemplo, onde estão presentes quatro cartas, há um número elevado de falsos positivos detectados, que terão de ser filtrados na fase de classificação. Este problema resulta do facto de, numa oclusão horizontal, criar-se uma área dentro dos contornos das cartas presentes que permite a existência de vários rectângulos em várias orientações, não sendo possível calcular quantas cartas podem estar presentes apenas com a informação dos contornos. Num destes casos, o facto de ser detectado um elevado número de rectângulos pode permitir a classificação das cartas do interior (as que estão entre as cartas das pontas), ainda que a classificação correcta destas não possa ser garantida.

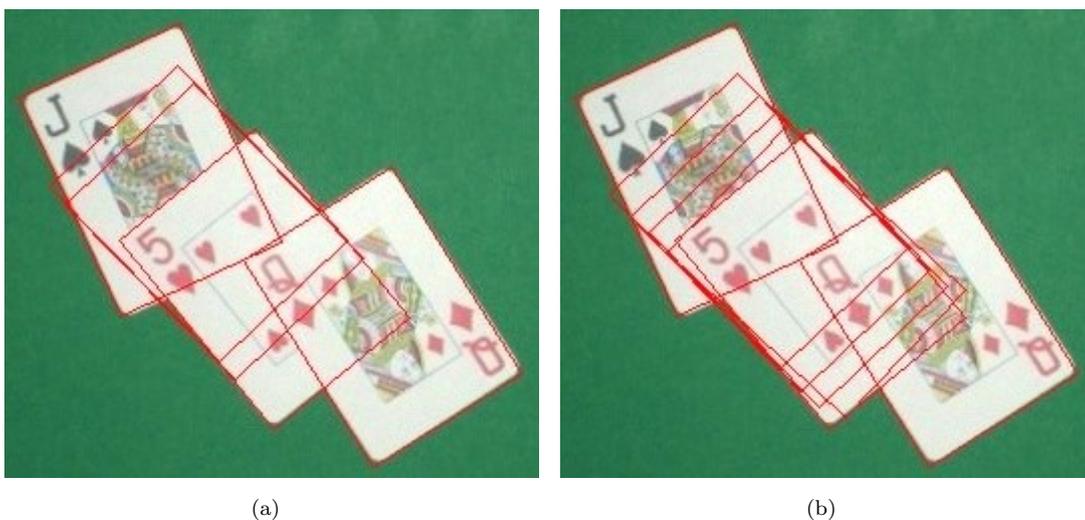


Figura 8.4: Detecção com rejeição por correspondência de cantos das cartas e dos rectângulos. (a) Rectângulos detectados com os três métodos de rejeição activos. (b) Rectângulos detectados sem a rejeição por correspondência de cantos.

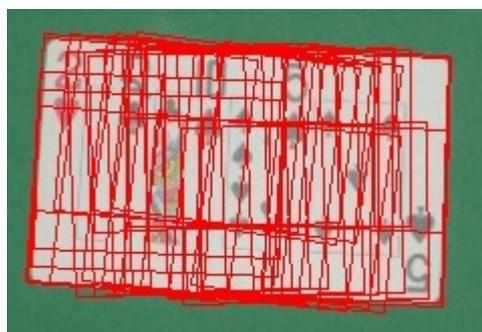


Figura 8.5: Detecção na presença de oclusões horizontais, havendo quatro cartas presentes.

8.2 Classificação de forma rígida

São agora avaliados os métodos de classificação rígidos. São estes o modelo probabilístico rígido, por verosimilhança, e o método de *template matching*. Por serem análogos, no sentido que estão feitos para o mesmo tipo de problema de classificação (objectos rígidos em orientação conhecida), são analisados os resultados de ambos simultaneamente.

Dadas condições de iluminação ideais (iguais às condições presentes nas imagens de treino), ambos os métodos conseguem ter uma taxa de sucesso de 100%. As diferenças de classificação entre os dois surgem quando as condições de iluminação são alteradas. É de esperar que o método probabilístico, sendo baseado em *edge features*, apresente uma robustez considerável a alterações de iluminação. A pouca robustez a estas alterações por parte dos métodos de *template matching* é típica.

As condições de iluminação podem ser descritas de forma simples em termos de luminosidade e contraste. A luminosidade representa o valor médio da intensidade de uma imagem (escala de cinzento). O contraste está relacionado com a diferença entre o máximo valor de intensidade e o mínimo valor. Pode também ser visto como a variância dos valores de intensidade. Uma imagem de baixo contraste apresenta uma baixa variância, tendo os vários pixels um valor de intensidade semelhante.

Foi analisada a robustez dos dois métodos a alterações de luminosidade e de contraste. Foi usado um conjunto de imagens de teste com condições de iluminação iguais às imagens usadas no treino dos dois métodos. A esse conjunto foram sujeitas alterações de luminosidade e de contraste de forma quantificada, procedendo-se à contagem de erros de classificação para cada valor de variação destas grandezas. Estas alterações foram feitas de forma abstracta, numa escala de -100 a 100, em que -100 representa a máxima alteração negativa (o mínimo contraste ou mínima luminosidade) e 100 representa a máxima alteração positiva (o máximo contraste ou máxima luminosidade). A figura 8.6 mostra exemplos de algumas alterações de luminosidade e contraste usadas. No centro está presente uma imagem nas condições de iluminação ideais. Do lado esquerdo estão alterações negativas, e do lado direito alterações positivas. A fila de cima mostra as alterações em luminosidade em incrementos de 20 face à imagem central, e a fila de baixo mostra as alterações de contraste em incrementos de 30.

Na figura 8.7 estão presentes os gráficos da taxa de reconhecimento dos dois métodos com a variação de luminosidade e de contraste. A linha a vermelho representa a taxa de reconhecimento do método probabilístico rígido, e a azul a do método de *template matching*. A taxa de reconhecimento tem em conta a detecção, pelo que se uma carta não for detectada, contará como um erro de reconhecimento. Como esperado, o método probabilístico mostra robustez a variações de luminosidade, tendo sempre uma taxa de reconhecimento de 100% numa larga gama de variação, enquanto que o método de *template matching* vê a sua taxa de reconhecimento descer rapidamente fora de uma pequena gama em torno da iluminação ideal. Nas condições de extremos de luminosidade (baixa ou alta) e de contraste mais baixo, a detecção falha gravemente, devido à diminuição dos gradientes nos contornos das cartas nestas condições. Assim, grande parte dos erros nestas condições são devido à não detecção.

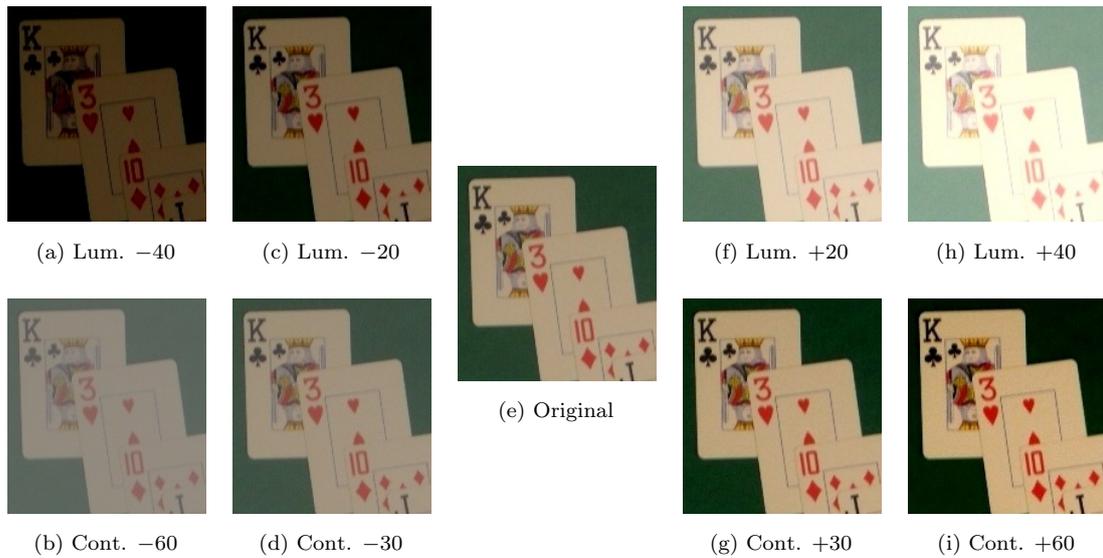


Figura 8.6: Variações de iluminação. (a) Variação de luminosidade de -40 . (b) Variação de contraste de -60 . (c) Variação de luminosidade de -20 . (d) Variação de contraste de -30 . (e) Imagem original. (f) Variação de luminosidade de $+20$. (g) Variação de contraste de $+30$. (h) Variação de luminosidade de $+40$. (i) Variação de contraste de $+60$.

Grande parte dos erros de classificação no reconhecimento por *template matching* ocorrem na classificação dos naipes. O facto de os naipes vermelhos terem um valor de intensidade mais baixo numa imagem de escala de cinzentos quando comparados com os naipes pretos, facilita a classificação de uma imagem de um naipe preto como sendo de um naipe vermelho, quando a imagem a classificar sofreu um aumento de luminosidade face à iluminação ideal.

Apesar da maior robustez do método deformável em relação ao *template matching*, os métodos apresentam uma grande diferença em termos de tempo computacional. A classificação por *template matching* consegue ser até 200 vezes mais rápida que com o modelo probabilístico, demorando cerca de 0.02 segundos a analisar uma imagem com 13 cartas.

8.3 Modelo probabilístico deformável

O modelo deformável explicitado no capítulo 5 não trouxe resultados satisfatórios. Dada a elevada complexidade do método, a classificação é extremamente demorada, podendo demorar mais de 200 segundos para analisar uma imagem com 13 cartas.

Ainda que este método consiga classificar correctamente imagens com deformações, onde os outros métodos rígidos não conseguem, os seus resultados nas condições rígidas consideradas ficam longe do desejado. Não foi possível encontrar um valor de *threshold* que separe os falsos negativos dos falsos

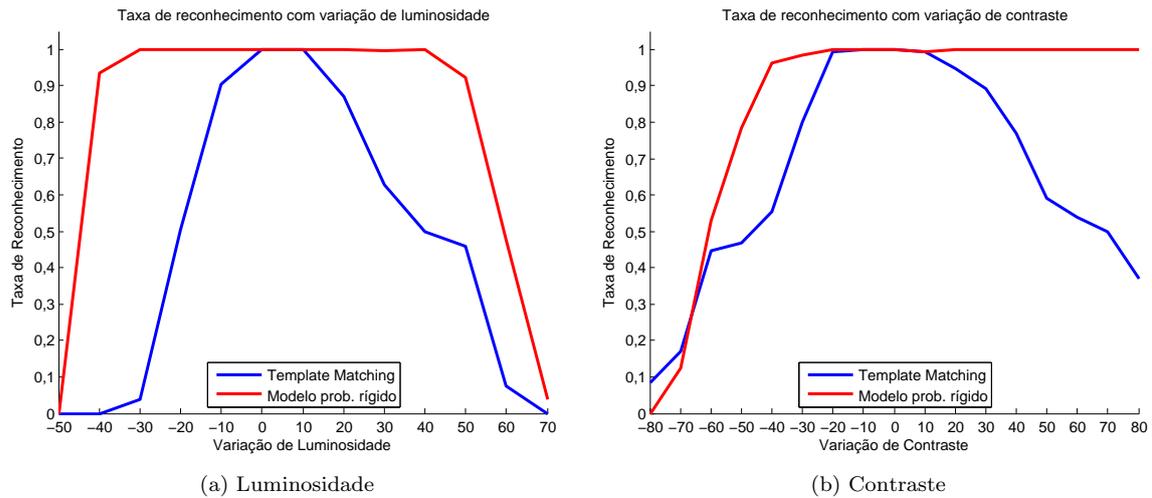


Figura 8.7: Taxas de reconhecimento com variação de (a) Luminosidade; (b) Contraste. As linhas a azul representam o *template matching*. As linhas a vermelho representam o modelo probabilístico rígido.

positivos. Assim, não é possível ter todas as cartas presentes correctamente classificadas e ao mesmo tempo filtrar as falsas detecções. Este resultado pode dever-se ao facto de uma deformação estimada pode transformar um objecto de uma classe de forma a ficar mais semelhante com outra. Como nesta aplicação não há deformações, o método não foi mais aprofundado neste sentido, visto que métodos mais simples funcionavam melhor e mais rapidamente.

8.4 Classificação com correcção de perspectiva

O método proposto foi testado em imagens previamente corrigidas por homografia. As diferentes resoluções das imagens originais podem afectar o resultado da correcção. A correcção pode ser afectada no mapeamento dos pontos para a imagem de destino, que com menor resolução torna necessária maior interpolação, e também na detecção dos cantos para calibração, pois numa resolução mais baixa os cantos podem não ser detectados no mesmo local que seriam numa maior resolução, dado que numa resolução mais baixa há uma maior incerteza acerca da localização exacta dos cantos. Se a qualidade da correcção for afectada, o resultado da classificação pode também ser afectado. Este resultado é tanto mais afectado quanto maior for o grau de perspectiva. A figura 8.8 mostra um exemplo deste problema, com um grau de perspectiva pouco acentuado. As diferenças entre as correcções por homografia a partir de duas resoluções diferentes é notória, principalmente nos contornos ou nos caracteres dos valores. A correcção a partir de baixa resolução tem pior definição. Nesta figura, a alta resolução é de 640×480 , e a baixa resolução é de 320×240 , havendo portanto uma diminuição de 50%.

A figura 8.9 mostra outro exemplo semelhante, mas com um grau de perspectiva mais acentuado.

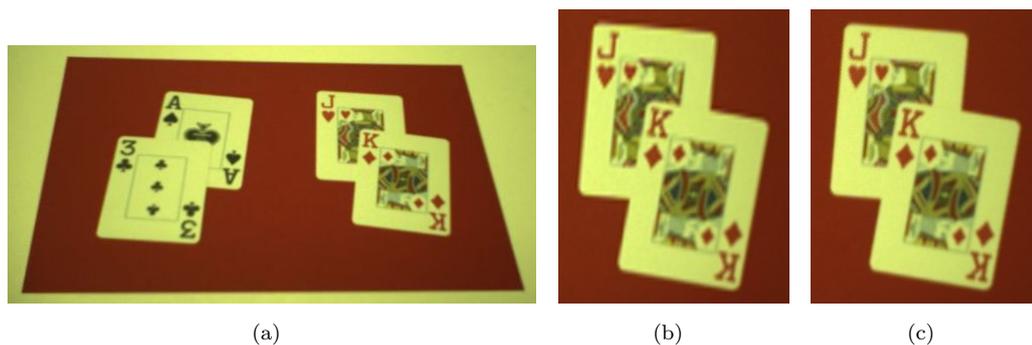


Figura 8.8: Correção de perspectiva pouco acentuada. (a) Imagem original. (b) Imagem corrigida a partir de baixa resolução. (c) Imagem corrigida a partir de alta resolução.

Neste exemplo, as cartas na imagem original mostram-se mais deformadas do que na figura 8.8a. A imagem corrigida a partir de baixa resolução mostra-se com uma definição muito inferior à corrigida por maior resolução. Neste caso, a alta resolução é de 1280×960 , e a baixa resolução é de 800×600 , havendo portanto uma diminuição de 37,5%.

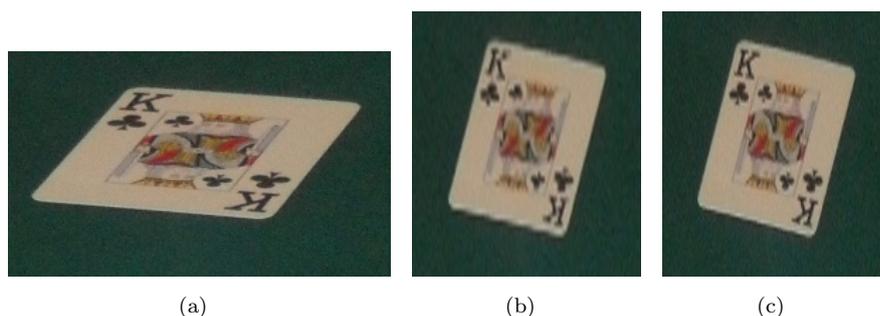


Figura 8.9: Correção de perspectiva acentuada. (a) Imagem original. (b) Imagem corrigida a partir de baixa resolução. (c) Imagem corrigida a partir de alta resolução.

Tendo várias imagens de cartas nos dois graus de perspectiva acima mostrados, foram testados os vários métodos de classificação de cartas usados. Nas imagens corrigidas a partir da maior resolução, ambos os métodos rígidos funcionam com sucesso, tendo uma taxa de reconhecimento de 100%. No entanto, com as imagens corrigidas a partir de menor resolução, o método probabilístico rígido obteve uma taxa de reconhecimento de 98.2% para o caso de pouca perspectiva da figura 8.8, e obteve resultados pobres no caso de perspectiva acentuada da figura 8.9, com uma taxa de reconhecimento de 79.5% e com uma diminuição dos valores de verosimilhança das cartas correctamente classificadas, em relação ao caso de alta resolução. Estes resultados podem dever-se ao facto da correção a partir de baixa resolução afectar a definição da imagem e assim a detecção das *edge features*. O método de *template matching* obteve resultados perfeitos com as imagens corrigidas a partir de qualquer resolução. Isto pode dever-se ao facto de o *template matching* usar todos os valores de intensidade da imagem, e da

menor definição devida à baixa resolução não resultar numa penalização muito forte.

O método probabilístico deformável também não funciona com sucesso neste caso. Ainda que haja alguma deformação dos caracteres, pode haver perda de *features*, como acontece com o modelo rígido.

Assim, usando a classificação por *template matching*, desde que com condições de iluminação controladas, o método funciona com imagens em perspectiva previamente corrigidas, anulando a restrição inicialmente imposta pela condição de rigidez nas imagens das cartas. Caso o baixo tempo computacional não seja uma condição importante, por exemplo numa aplicação que não necessite de ser em tempo-real, e havendo várias alterações de iluminação possíveis e inesperadas, o método probabilístico rígido mostra-se uma boa opção para que o método global seja implementado com sucesso.

Capítulo 9

Conclusão

Foi proposto um método para detecção e reconhecimento de cartas de jogar. Foi desenvolvido um método original baseado na transformada de Hough para a detecção das cartas como rectângulos. Sendo um método de votação, a detecção funciona em casos de oclusão parcial. Para a classificação, feita de forma independente da detecção, foram comparados três métodos de reconhecimento de objectos: *template matching* e dois métodos probabilísticos, um rígido e outro deformável. Com condições de iluminação controladas, o melhor método para esta aplicação é o *template matching*, pela eficácia e eficiência computacional. O método probabilístico rígido funciona bem em várias condições de iluminação, mas é mais lento e apresenta problemas quando há pouca definição numa imagem corrigida por homografia. O método probabilístico deformável mostra-se demasiado lento e com pouca eficácia para esta aplicação rígida, não sendo um bom método a considerar.

Como é natural de qualquer trabalho com tempo limitado, vários outros caminhos poderiam ter sido explorados ou aprofundados. O método proposto simula a visão humana no reconhecimento de cartas apenas para um caso simples e controlado. Na verdade, o ser humano consegue identificar as cartas mesmo sem observar os seus contornos, procurando e encontrando um símbolo de valor da carta e um símbolo de naipe por baixo. Consegue também identificar as cartas mesmo sem a imagem do canto da carta, visto que o interior da carta contém toda a informação necessária acerca desta. Assim, alguns dos caminhos que poderiam ter sido explorados são apresentados.

Detecção de rectângulos. A detecção realizada assume que as cartas não possuem qualquer deformação e têm uma escala fixa e conhecida. Uma detecção mais robusta poderia encontrar cartas mesmo sob perspectiva. A própria detecção das cartas e consequentemente a detecção de perspectiva poderia ser usada para calibrar a imagem, a fim de obter as imagens dos cantos sem deformação e na escala correcta. No entanto, ainda que o método se torne mais robusto, a classificação continuaria a depender da presença dos contornos.

Detecção dos valores das cartas. Uma forma de integrar a detecção e classificação das cartas num

só passo é procurar na imagem valores de cartas (caracteres e símbolo de naipe em baixo). Poderia ser usado um algoritmo de detecção de *features* robustas como SIFT[10] ou SURF[18], treinado para os valores das cartas. Assim, a detecção não depende da visibilidade dos contornos, e seria robusto a deformações como por perspectiva, até certo ponto. A classificação estaria inerente à detecção, identificando cada região candidata (*blob*) como pertencendo a uma carta, classificando-a, ou rejeitando-a.

Classificação a partir da imagem interior. A imagem no interior da carta contém toda a informação necessária para a classificar. Nas cartas de valor (Ás, 2,...,10) o valor é dado pelo número de “símbolos de naipe” no seu interior . Estes símbolos têm todos o mesmo tamanho, excepto no caso do Ás de Espadas, que possui um símbolo de naipe maior, normalmente indicativo da marca das cartas. A classificação pode ser feita pela contagem dos símbolos, caso estejam todos visíveis, ou pela disposição destes, que varia conforme o valor e pode ser usada em casos de oclusão parcial (com limitações). As cartas de figura (Valete, Dama e Rei) possuem uma imagem no interior. Ainda que não seja fácil por parte de um observador humano identificar uma carta de figura apenas por esta imagem, todas estas cartas têm uma imagem interior diferente das outras. Esta imagem pode ser detectada e classificada por um descritor de SIFT ou SURF, mesmo havendo alguma oclusão.

Integração de todos os métodos. O ser humano usa todos estes processos para identificar cartas de jogar. Um método robusto poderia realizar o mesmo, partindo do método mais simples até ao mais complexo à medida que fosse necessário.

Bibliografia

- [1] C. Zheng and R. Green, “Playing card recognition using rotational invariant template matching,” in *Proc. of Image and Vision Computing New Zealand*, pp. 276--281, 2007.
- [2] K. Zutis and J. Hoey, “Who’s Counting? Real-Time Blackjack Monitoring for Card Counting Detection,” in *Computer Vision Systems: 7th International Conference on Computer Vision Systems, ICVS 2009, Liège, Belgium, October 13-15*, p. 354, 2009.
- [3] Y. Amit and A. Trouvé, “Pop: Patchwork of parts models for object recognition,” *International Journal of Computer Vision*, vol. 75, no. 2, pp. 267--282, 2007.
- [4] D. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern recognition*, vol. 13, no. 2, pp. 111--122, 1981.
- [5] F. Han and S. Zhu, “Bottom-up/top-down image parsing with attribute grammar,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 1, pp. 59--73, 2009.
- [6] B. Micušik, H. Wildenauer, and J. Košecká, “Detection and matching of rectilinear structures,” in *CVPR*, vol. 1, p. 3, 2008.
- [7] J. Rodrigues, P. Aguiar, and J. Xavier, “ANSIG -- An Analytic Signature for Permutation Invariant 2D Shape Representation,” in *IEEE CVPR, USA*, 2008.
- [8] R. Smith, “An Overview of the Tesseract OCR Engine,” in *Proceedings of the Ninth International Conference on Document Analysis and Recognition-Volume 02*, pp. 629--633, IEEE Computer Society, 2007.
- [9] C. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, “A license plate-recognition algorithm for intelligent transportation system applications,” *IEEE Transactions on Intelligent transportation systems*, vol. 7, no. 3, pp. 377--392, 2006.
- [10] D. Lowe, “Object recognition from local scale-invariant features,” in *iccv*, p. 1150, Published by the IEEE Computer Society, 1999.
- [11] I. Inc., “The OpenCV Open Source Computer Vision Library.” Intel Inc., [Online]. Available: <http://opencvlibrary.sourceforge.net>.

- [12] Y. Amit, *2D Object Detection and Recognition: Models, Algorithms, and Networks*. The MIT Press, 2002.
- [13] Y. Amit, D. Geman, and X. Fan, “A Coarse-to-Fine Strategy for Multiclass Shape Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, p. 12, 2004.
- [14] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, “The iCub humanoid robot: an open platform for research in embodied cognition,” in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, pp. 50--56, ACM, 2008.
- [15] J. S. Marques, *Notes for Image Processing and Vision*. Instituto Superior Técnico.
- [16] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O’Reilly Media, 2008.
- [17] R. Szeliski, “Computer Vision: Algorithms and Applications,” *Microsoft Research*, 2010.
- [18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346--359, 2008.