

---

# UMA ABORDAGEM DE RECONHECIMENTO DE OBJETOS COM USO DA PROJEÇÃO POR HISTOGRAMA VOLTADA PARA ROBÓTICA MÓVEL

André de Andrade Bindilatti

FACEN-UNIMEP  
Piracicaba SP

Luiz Eduardo Galvão Martins

UNIFESP – Instituto de Ciência e Tecnologia  
São José dos Campos SP

---

**Resumo:** Neste artigo é descrita uma abordagem para o reconhecimento automático de formas em imagens digitais, baseada no uso de projeção de segmentos binários de imagem por meio de histogramas. Tal abordagem pode ser vista como uma técnica de *template matching* para o reconhecimento de formas. Um teste comparativo entre a abordagem citada e a técnica de *template matching* pela correlação de imagens é apresentado. A abordagem apresentada está sendo desenvolvida com a finalidade de oferecer capacidade de visão artificial para robôs móveis inteligentes.

**Palavras Chaves:** Visão Computacional, Reconhecimento de Objetos, Processamento de Imagens.

**Abstract:** This article describes an approach for automatic object recognition in digital images, based on the projections of binary segments of images through the use of histograms. Such approach can be thought as a template matching technique for object recognition. A comparative test between that approach and the correlation template matching technique is also presented. The presented approach has been developed to offer artificial vision for intelligent mobile robots.

**Keywords:** Computer Vision, Object Recognition, Image Processing.

## 1 INTRODUÇÃO

A habilidade de distinguir objetos em uma cena é comum para os humanos; o sistema de visão humana é capaz de reconhecer e diferenciar entre diversos objetos rapidamente. Contudo, reproduzir esta mesma habilidade em sistemas de computador, particularmente em sistemas robóticos, tem sido tema de muitas pesquisas. Tal meta ainda é um desafio para as atuais tecnologias e metodologias propostas como abordagem para o problema em questão e, em geral, possuem aplicação restrita a domínios específicos.

A visão computacional consiste em um campo de pesquisa que objetiva o desenvolvimento de técnicas e aplicações computacionais, capazes de reproduzir habilidades tais como as acima mencionadas, em sistemas artificiais. De modo a complementar, Amit (2002) define o objetivo da visão computacional como sendo o desenvolvimento de algoritmos que recebem imagens como entrada, e fornecem como saída, informações descritivas sobre objetos nelas presentes.

O presente trabalho tem como objetivo contribuir para o campo da automação e da robótica inteligente, ao investigar possíveis soluções para o reconhecimento de objetos. A tarefa de reconhecimento de objetos é parte fundamental de sistemas de visão computacional, embora não se trate da única atividade realizada por tais sistemas.

Neste trabalho foram desenvolvidos dois aplicativos de visão computacional para o reconhecimento de objetos, mais especificamente, para a identificação de sinalizações em um ambiente. Como foco, foi abordado o problema de identificar e distinguir ícones de sinalização, como setas de indicação para direita e esquerda.

Uma das aplicações desenvolvidas é baseada na técnica de *template matching* por correlação; a segunda aplicação é baseada em uma abordagem que faz uso da projeção de segmentos binários, como representação das formas para o processo de reconhecimento. Ambas as abordagens podem ser vistas como variações da técnica de *template matching*. Um teste comparativo é realizado com o intuito de demonstrar e avaliar o desempenho das aplicações.

O restante desse artigo está organizado da seguinte forma: a seção 2 descreve os fundamentos da técnica de *template matching*; a seção 3 descreve as técnicas de processamento de imagem empregadas na abordagem discutida nesse artigo; a seção 4 relata como as aplicações descritas foram implementadas; a seção 5 apresenta os resultados obtidos em um teste comparativo entre a técnica de *template matching* por correlação e a abordagem com o uso de projeções; e a seção 6 encerra o artigo apresentando as conclusões e considerações finais.

## 2 TEMPLATE MATCHING

O *template matching* consiste na busca por ocorrências de um determinado objeto ou forma em uma dada imagem, tendo como base um modelo de referência da forma ou objeto a ser reconhecido. A presença do objeto é detectada a partir da correspondência entre esse modelo e as sub-regiões da imagem processada. O referido modelo de referência é chamado de *template*.

Em casos mais simples, uma imagem em escala de cinza do objeto de interesse pode ser utilizada como *template*. No entanto, conforme comentado por Forsyth e Ponce (2003), em

outros casos, o *template* pode ser construído a partir de uma etapa de treinamento sobre exemplares do objeto.

Treiber (2010) descreve uma abordagem para o *template matching*, que consiste no uso de uma imagem do objeto de interesse como *template*. Em tal abordagem, o *template* é posicionado sobre cada possível posição da imagem processada e em cada passo, é realizado o cálculo do coeficiente de **correlação cruzada normalizado** (CCN). O coeficiente CCN é utilizado como medida de correlação entre o *template* e as sub-regiões sobre as quais é posicionado na imagem em questão. Sub-regiões cujo valor resultante do cálculo do coeficiente CCN excederem um dado limiar, possuem boa probabilidade de possuir uma ocorrência do *template*. A fórmula para o cálculo do coeficiente CCN é a seguinte:

$$CCN(a, b) = \frac{\sum_{x=0}^W \sum_{y=0}^H (f(x+a, y+b) - \mu_f) \cdot (t(x, y) - \mu_t)}{\sqrt{\sum_{x=0}^W \sum_{y=0}^H (f(x+a, y+b) - \mu_f)^2 \cdot \sum_{x=0}^M \sum_{y=0}^N (t(x, y) - \mu_t)^2}}$$

Nessa fórmula,  $f(x, y)$  corresponde a função que descreve os valores dos pixels da imagem processada, para as dadas coordenadas de pixel  $(x, y)$ ;  $t(x, y)$  a função que descreve os valores dos pixels da imagem *template*;  $(a, b)$  é o deslocamento com relação a origem da imagem para a posição onde o coeficiente CCN é calculado;  $\mu_f$  e  $\mu_t$  correspondem as médias das funções  $f$  e  $t$  na sub-região em questão; e  $W$  e  $H$  são as dimensões do *template*, sendo sua largura e altura respectivamente.

### 3 TÉCNICAS DE PRÉ-PROCESSAMENTO EMPREGADAS

Nesse trabalho uma variação da técnica de *template matching* foi implementada como abordagem para o reconhecimento de formas em imagens digitais. Como alvo para reconhecimento, sinalização de setas indicativas para direita e esquerda foram utilizadas em testes práticos. O processamento realizado sobre as imagens nessa abordagem, pode ser descrito por meio dos seguintes passos:

- (1) **Segmentação da imagem:** essa etapa é necessária para possibilitar que objetos de interesse sejam detectados na imagem e processados individualmente;
- (2) **Deteção dos componentes isolados:** após obter uma imagem segmentada, é realizado um novo processamento para se extrair os diferentes componentes isolados na imagem, isto é, processar a imagem de segmentos de modo a extrair as regiões de pixels conexos;
- (3) **Projeção dos componentes extraídos:** nessa etapa é feita a projeção dos componentes individuais obtidos na etapa anterior;
- (4) **Comparação entre as projeções dos objetos da imagem com a projeção do *template*:** a projeção da imagem *template* é comparada com as projeções dos objetos obtidos no processamento da imagem de entrada. Uma medida de similaridade é utilizada como critério de comparação, por meio de um limiar os objetos são classificados como ocorrências ou não do *template*; se a pontuação da medida de similaridade for superior ao limiar uma ocorrência do objeto é reconhecida, caso contrário, o segmento candidato é rejeitado.

Para a implementação dos passos descritos, técnicas de processamento de imagem são aplicadas. As subseções a seguir descrevem os conceitos a cerca de tais técnicas.

### 3.1 Limiarização de Imagens

A limiarização ou binarização de uma imagem, segundo Pedrini e Schwartz (2008), trata-se de uma das técnicas mais simples de segmentação de objetos e consiste na definição de um ou mais limiares, por meio dos quais é estabelecido um critério para classificar os pixels em uma imagem entre zero ou um.

Em outras palavras, a limiarização consiste na discriminação de pixels de uma imagem, de acordo com seus valores de intensidade, com respeito aos limiares preestabelecidos. Assim, o resultado dessa operação é uma nova imagem na qual os pixels possuem um entre dois valores.

Conceitualmente, os pixels em uma imagem limiarizada são definidos com valores de intensidade zero ou um. Contudo, na prática esta não é uma questão restrita; os valores em uma imagem limiarizada podem ser definidos como 0 e 255 (em imagens de 8-bits) por exemplo, ou quaisquer outros pares de valores. O aspecto relevante é que os pixels serão classificados entre dois valores distintos.

A ideia sobre a qual essa técnica fundamenta-se, parte da premissa de que níveis de cinza pertinentes a um objeto em uma cena, se concentram em um intervalo diferente dos níveis de cinza dos pixels que compõe o fundo da imagem.

Portanto, é possível afirmar que o sucesso desta técnica de segmentação, depende da definição de limiares que possam ser utilizados para distinguir pixels em uma imagem de acordo com duas classes; segundo Russ (2011), em geral essas duas classes representam os pixels que fazem parte do fundo e os pixels que fazem parte dos objetos presentes na imagem.

Jain, Kasturi e Schunk (1995) definem o processo de limiarização sobre perspectiva de três casos: (1) com o uso de um limiar fixo  $L$ ; (2) com o uso de um intervalo fechado  $[L_1, L_2]$ ; e (3) com o uso de um esquema mais geral, no qual diferentes intervalos de limiares são definidos. Neste último caso, um conjunto  $Z$  define os valores de intensidade cujos objetos de interesse supostamente irão possuir.

Em símbolos, as definições de limiarização de Jain, Kasturi e Schunk (1995), podem ser enunciadas nas seguintes formas:

- Quando um único limiar for utilizado:

$$B(x, y) = \begin{cases} 1 & \text{se } f(x, y) \leq L \\ 0 & \text{caso contrário} \end{cases}$$

- Quando os valores de intensidade se situam em um intervalo:

$$B(x, y) = \begin{cases} 1 & \text{se } L_1 \leq f(x, y) \leq L_2 \\ 0 & \text{caso contrário} \end{cases}$$

- Quando os valores de intensidade dos objetos de interesse são representados por um conjunto de intervalos de limiares:

$$B(x, y) = \begin{cases} 1 & \text{se } f(x, y) \in Z \\ 0 & \text{caso contrário} \end{cases}$$

Embora a limiarização seja uma técnica efetiva para algumas situações, em geral, essa técnica requer cenários onde a iluminação é um fator controlado e conhecimento de domínio possa ser integrado à solução, isto é, fatores relacionados às distribuições de níveis de cinza dos objetos e fundo de uma imagem necessitam ser previamente conhecidos, e aproximadamente fixos. Histogramas de níveis de cinza podem ser utilizados para o estudo de tais fatores.

É um fato conhecido que a limiarização global (com limiares fixos para todos os pixels da imagem processada) é insuficiente

para obter certos resultados quando a iluminação não é uniforme em uma imagem (CHAN; LAM; ZHU, 1998; BRADSKI; KAEHLER, 2008).

Para superar este problema, existem diversas técnicas que se baseiam no uso de limiares que podem variar em função dos valores locais dos níveis de cinza, ou ainda capazes de estabelecer automaticamente, com base na imagem processada, diferentes limiares capazes de proporcionar melhores resultados no processo de limiarização.

Usualmente, técnicas de limiarização desse gênero são referidas por limiarização adaptativa. A destacar, Chan, Lam e Zhu (1998) propõem um método de limiarização adaptativa baseado na teoria variacional (princípio amplamente conhecido em Física).

### 3.2 Extração de Segmentos

O processo de segmentação de uma imagem consiste no particionamento de tal imagem, em sub-regiões ou segmentos. Tais segmentos normalmente representam objetos de interesse presentes na imagem. Assim sendo, após a etapa de segmentação, é necessário realizar a extração das sub-regiões encontradas.

Com o uso de uma função disponibilizada pela biblioteca de código *OpenCV*, denominada *cvFindContours*, os segmentos presentes na imagem limiarizada são extraídos. A função *cvFindContours* retorna uma sequência com os contornos presentes na imagem; os contornos são representados por estruturas de dados fornecidas pela própria biblioteca e, em particular, uma propriedade importante dessas estruturas é sua região retangular. A região retangular que envolve cada contorno representa uma sub-região da imagem, na qual estão presentes vizinhanças conexas de pixels.

### 3.3 Histogramas

Histogramas são importantes ferramentas estatísticas para a representação e sumarização de dados. Em seu sentido mais comum, representam distribuições de frequências a respeito de medições ou dados a cerca de algum estudo ou contexto arbitrário. Os histogramas possuem grande relevância, pois fornecem uma estrutura poderosa que viabiliza ou facilita a realização de diversas atividades ou técnicas de análise e processamento.

Conforme Burger e Burge (2009), em processamento de imagens, histogramas são utilizados em diversas técnicas de realce e tratamento de qualidades visuais; podem ser utilizados na detecção de problemas de exposição e iluminação não uniforme no processo de aquisição de imagens; ou até mesmo como ferramentas forenses, quando utilizadas para se determinar que tipos de atividades de processamento foram aplicados previamente em uma determinada imagem.

Chang e Krumm (1999) utilizam uma variação do histograma de cores – o histograma de co-ocorrência de cores – para o reconhecimento de objetos em imagens. Tal histograma se difere do histograma de cores convencional, por manter uma contagem da ocorrência de determinados pares de cores, espaçados por certas distâncias no espaço da imagem. Em sua abordagem, Chang e Krumm (1999) fazem uso de histogramas da co-ocorrência de cores do objeto de interesse, computados de diferentes perspectivas, e então o objeto é localizado nas imagens de entrada a partir da comparação com os histogramas do modelo.

Não obstante, de acordo com Bradski e Kaehler (2008), histogramas também podem ser utilizados para representação da distribuição de cores de um objeto; podem ser utilizados como descritores sobre a forma de um objeto, por meio do histograma da orientação dos gradientes de suas arestas; e ainda, como a representação da função de distribuição de probabilidade, com respeito à hipótese sobre a localização de um objeto em uma imagem.

Formalmente, um histograma pode ser visto como uma função discreta  $h(i) = n_i$ , sendo  $i$  um inteiro em um intervalo fechado  $[0, K - 1]$ ;  $K$  é o número de intervalos ou classes sobre os quais os dados são distribuídos; e o valor  $n_i$  corresponde a frequência ou a contagem de itens pertinentes a coleção de dados subjacentes, associada ao  $i$ -ésimo intervalo ou  $i$ -ésima classe.

O significado atribuído às frequências e intervalos de um histograma depende do contexto de sua aplicação. Por exemplo, no caso do histograma dos níveis de cinza de uma imagem monocromática, os possíveis níveis de cinza podem ser divididos em  $K$  intervalos de igual tamanho; e assim sendo, cada  $n_i$  representa a contagem de pixels presentes na imagem, com um valor de intensidade de cinza dentro dos limites do  $i$ -ésimo intervalo.

Uma operação importante é a comparação de histogramas, uma vez que histogramas podem ser utilizados como forma de sumarizar ou organizar informações sobre características ou atributos de segmentos ou objetos em imagens. Existem vários métodos para a comparação de histogramas. Bradski e Kaehler (2008) citam alguns deles: correlação; chi-quadrado; interseção; e Bhattacharyya.

### 3.4 Projeções de Imagens

Conforme descrevem Pedrini e Schwartz (2008), as projeções horizontais e verticais de uma imagem binária consistem, respectivamente, na soma dos pixels em cada linha da imagem (projeção horizontal); e na soma dos pixels em cada coluna (projeção vertical). De acordo com esta definição, é possível observar que histogramas podem ser adotados como estruturas convenientes para o cálculo de tais projeções. Também é importante salientar que uma mesma projeção pode resultar de diferentes objetos ou imagens.

Em dos Santos *et al.* (2009) projeções verticais e horizontais são utilizadas em um método para a extração de texto de páginas de documentos, para fins de aplicações em sistemas de reconhecimento de documentos. No referido trabalho, páginas de documentos são submetidas a um processo de limiarização, e então um conjunto de operações, envolvendo o processamento dos histogramas das projeções verticais e horizontais, são utilizadas em um algoritmo para a extração automática de segmentos de linhas de texto.

## 4 IMPLEMENTAÇÃO DAS APLICAÇÕES

Duas aplicações para o reconhecimento de formas e objetos foram feitas; uma aplicação baseada no *template matching* por meio do cálculo de correlação; e uma segunda aplicação baseada na abordagem de histogramas descrita anteriormente. O intuito ao implementar uma abordagem tradicional, tal como o *template matching* baseado na correlação, foi o de proporcionar uma base para comparações. As aplicações foram implementadas na linguagem de programação C++, com a utilização da biblioteca de código *OpenCV*.

A *OpenCV* trata-se de uma biblioteca de código aberto, destinada a aplicações de visão computacional. É escrita em C e C++ e oferece suporte multiplataforma, ou seja, é compatível com várias plataformas de desenvolvimento, tais como Linux, Microsoft Windows e Mac OS X (BRADSKI; KAEHLER, 2008). Essa biblioteca de código fornece a implementação otimizada de muitos algoritmos e rotinas de uso comum em aplicações de visão computacional.

O desenvolvimento das aplicações foi realizado no ambiente Windows, utilizando-se uma *webcam* como dispositivo para a aquisição de imagens. Para a realização de testes, uma folha impressa contendo a seta a ser detectada foi exposta frente à câmera e como resultado, as aplicações exibiam as imagens capturadas, e desenhavam um quadro ao redor da área contendo o objeto quando reconhecido.

Para avaliação do comportamento das aplicações em um ambiente próximo ao de um robô móvel inteligente, foi utilizado uma giga de testes baseada em um dos módulos pertinentes a uma plataforma robótica, chamada RoboDeck, desenvolvido pela empresa Xbot.

A referida giga de testes consistiu em um módulo embarcado de alto desempenho, equipado com uma *webcam* para a captura de imagens. Como plataforma de desenvolvimento, a giga de testes contou com uma distribuição Linux, o Debian.

Uma vez que o código utilizado na implementação das aplicações mencionadas estava em conformidade com o padrão ANSI, e tendo em mente que a *OpenCV* trata-se de uma biblioteca multiplataforma, as aplicações puderam ser facilmente compiladas em ambos os ambientes (Windows e Linux).

## 4.1 Correlation Template Matching

A abordagem nessa aplicação é basicamente tal como a técnica descrita na seção 2. Um *looping* realiza a aquisição de imagens por meio do dispositivo de captura disponível, e utilizando um *template*, realiza o cálculo do coeficiente CCN do template com relação a cada região possível da imagem.

Dado um *template* com dimensões  $W \times H$ , se a imagem processada possuir dimensões  $M \times N$ , então existem  $(M - W + 1) \times (N - H + 1)$  formas de posicionar o *template* sobre a imagem para o cálculo de correlação. Logo uma matriz de tamanho apropriado é utilizada para o armazenamento dos coeficientes de CCN, obtidos após o processamento da imagem com o método já descrito.

Para estimar a posição de uma provável ocorrência do objeto de interesse, uma chamada a uma função da OpenCV, nomeada *cvMinMaxLoc*, é utilizada para encontrar o ponto de máxima local na referida matriz dos coeficientes de CCN. Caso o valor de máxima local for superior a um limiar parametrizado, então a posição da imagem correspondente a esse coeficiente é informada como a posição de uma ocorrência do *template*.

Toda a lógica relacionada com o processo de busca do *template* foi encapsulada dentro de uma classe nomeada *TemplateMatch*. Desse modo, para estender a capacidade da aplicação para o reconhecimento de múltiplas formas, provenientes de diferentes templates, é necessário apenas a inclusão de código para instanciar novos objetos da classe *TemplateMatch*, com diferentes imagens *template*.

## 4.2 Projeção por Histogramas

Essa abordagem se baseia no uso das projeções de objetos extraídos de uma imagem por meio de um processo de segmentação, e de sua comparação com a projeção de uma imagem de referência do objeto a ser reconhecido. Para facilitar o cálculo das projeções e do processo de comparação, histogramas foram utilizados como forma de representação para as projeções.

A projeção de um objeto proporciona uma estrutura mais conveniente para fins de processamento de sua forma, isto é, as projeções de um objeto fornecem um meio simplificado para expressar informações sobre características da forma geral de tal objeto. No contexto desse trabalho, as projeções foram utilizadas para facilitar o reconhecimento do objeto. A Figura 1 ilustra o resultado da projeção vertical da imagem de uma seta. Os resultados ilustrados na Figura 1 foram obtidos gerando o gráfico do histograma produzido pela aplicação desenvolvida.

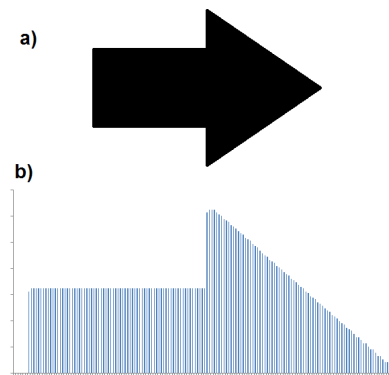


Figura 1 – Projeção vertical da imagem de uma seta; (a) Imagem da seta; (b) Gráfico da projeção vertical gerada a partir de dados produzidos pela aplicação.

Nesta segunda implementação, as imagens foram processadas por meio de uma operação de limiarização. Uma vez que as setas impressas eram de cor preta (níveis de cinza próximos de zero), definir limiares que pudessem realizar uma boa segmentação (para discriminar a seta do fundo em que se encontra) não foi uma tarefa complicada.

Com o uso de uma função disponibilizada pela *OpenCV*, os segmentos presentes na imagem limiarizada foram extraídos. A partir das regiões retangulares que envolvem cada segmento, o histograma de sua projeção vertical foi calculado. Regiões com áreas inferiores a um limiar de tamanho foram descartadas, evitando o processamento desnecessário de regiões muito pequenas para serem candidatas a uma ocorrência do objeto. Na projeção das imagens, histogramas foram instanciados para o armazenamento dos valores das projeções.

O processo de projeção foi feito dividindo a imagem em faixas horizontais ou verticais (para as projeções horizontais e verticais respectivamente), uma vez que a imagem foi limiarizada e seus segmentos poderam ser vistos como regiões de pixels binários, o número de pixels não nulos (isto é, dos pixels que foram discriminados como parte de um objeto) em cada faixa resultou no histograma de projeção.

O número de intervalos ou faixas utilizados na projeção do histograma foi um fator importante. Na implementação em questão, esse valor foi um dado parametrizado e precisou ser escolhido com cautela.

Se uma região for dividida em faixas muito largas, a capacidade expressiva do histograma resultante, como meio para a representação da forma presente em tal região, é

comprometida; grande parte dos aspectos característicos da forma é perdida. De modo análogo, utilizar faixas muito estreitas pode tornar a projeção muito característica de um exemplar específico da forma do objeto, comprometendo a capacidade de generalização do modelo.

Após gerar os histogramas das formas extraídas da imagem segmentada e do *template* do objeto de interesse, a busca pelo objeto se resumiu na comparação dos histogramas computados com o histograma do *template*. A localização do objeto detectado foi estabelecida como sendo a região retangular envolvendo o segmento reconhecido como ocorrência do objeto.

Qualquer um dos métodos de comparação de histogramas citados na sessão 3.3 podem ser utilizadas na etapa de comparação dos histogramas. O critério para rejeitar ou reconhecer um dado segmento como uma possível ocorrência do objeto de interesse, deve ser estabelecido de acordo com a interpretação de cada medida de comparação.

De acordo com Bradski e Kaehler (2008), a comparação pelo critério de interseção para a comparação de histogramas é melhor em aplicações onde a precisão não é um fator tão importante quanto o desempenho; ao passo que os critérios chi-quadrado e Bhattacharyya proporcionam medidas mais acuradas, porém ao custo de um desempenho inferior.

## 5 TESTE COMPARATIVO

Para avaliar o desempenho das aplicações implementadas, foi utilizado uma amostra com quarenta imagens digitais com resolução de 640x480; dez imagens exibindo um retângulo; dez imagens exibindo uma seta indicando para a direita; dez imagens exibindo uma seta indicando para esquerda; e dez imagens sem nenhuma seta.

O objetivo desse teste foi avaliar a precisão ao detectar o objeto correto, a capacidade entre distinguir uma seta indicando para a direita e para a esquerda, e quanto à robustez do algoritmo com respeito a falsos positivos, isto é, indicar a presença do objeto de interesse em imagens nas quais o objeto não estava presente.

Nos testes realizados, foram utilizadas imagens de um retângulo com o intuito de verificar se as aplicações confundiriam os sinais de interesse com outros sinais semelhantes; testes com imagens sem a presença de uma seta foram utilizadas para avaliar taxas de falsos positivos; e imagens com setas indicando para esquerda e direita para testar a capacidade de distinguir entre ambos os sinais.

O tempo de processamento para o reconhecimento de cada imagem individual também foi monitorado. Uma vez que aplicações envolvendo a detecção de objetos para oferecer autonomia para robôs possuem o requisito de processamento em tempo real, o propósito dessa medida foi o de fornecer uma estimativa do tempo de processamento médio para cada aplicação, no processamento de uma imagem.

A Tabela 1 ilustra os resultados de teste para a aplicação baseada no *template matching*; e a Tabela 2 ilustra os resultados para a aplicação baseada no uso de histogramas.

A Figura 2 exibe um gráfico comparativo do desempenho de ambas as aplicações. Para cada conjunto com dez imagens, as respostas corretas obtidas pelas aplicações, em cada situação, foram sumarizadas e seu percentual de acerto é ilustrado no gráfico.

Tabela 1 – Resultados do teste da aplicação baseada no *template matching*.

Teste	Quantidade	
	Seta para direita	Seta para esquerda
Imagens com um retângulo	0	0
Imagens com uma seta para direita	8	0
Imagens com uma seta para esquerda	0	6
Imagens sem a presença de setas	0	0
Média do tempo de execução por imagem (milissegundos)	294,7 milissegundos	

Tabela 2 – Resultados do teste da aplicação baseada na projeção por histogramas.

Teste	Quantidade	
	Seta para direita	Seta para esquerda
Imagens com um retângulo	0	2
Imagens com uma seta para direita	4	4
Imagens com uma seta para esquerda	0	4
Imagens sem a presença de setas	1	0
Média do tempo de execução por imagem (segundos)	21 milissegundos	

Para as imagens com uma seta para direita ou esquerda, a resposta correta é quando a aplicação identifica uma seta para direita e para esquerda respectivamente; nas imagens com um retângulo, a resposta correta é quando o algoritmo não identifica nenhuma seta; e para imagens sem a presença de setas, a resposta correta é quando a aplicação não identifica nenhuma seta.

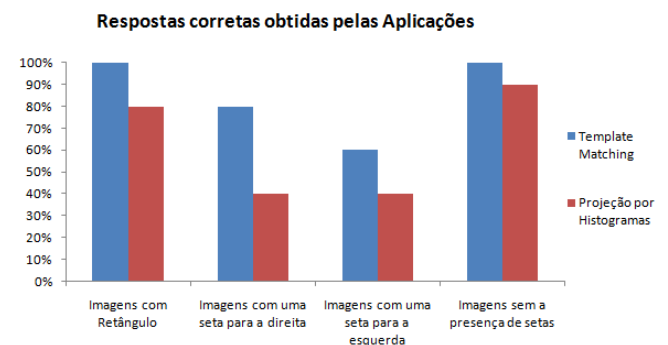


Figura 2 – Gráfico comparativo do desempenho das aplicações.

## 6 CONCLUSÃO

Os resultados dos testes mostraram que o *template matching* proporcionou o reconhecimento das formas com maior precisão. Contudo, a abordagem baseada na projeção por histogramas teve um desempenho muito superior quanto ao tempo médio de processamento por imagem, demonstrando um

tempo de execução equivalente a cerca de 7,12% do tempo médio de execução do *template matching*.

A menor precisão no reconhecimento das formas, demonstrada pela abordagem baseada no uso da projeção por histogramas, se deu em função da pouca robustez quanto a variações de iluminação no processo de segmentação, comprometendo o resultado das projeções.

Para aumentar a precisão da abordagem baseada nas projeções por histograma, uma alternativa seria o emprego de métodos de segmentação mais robustos quanto à iluminação, com relação à técnica de limiarização, utilizada na abordagem em pauta.

Na abordagem utilizando o método *template matching* por correlação, o cálculo do coeficiente CCN é efetuado em cada possível região que possa ser ocupada pelo *template*. Na abordagem baseada na projeção de histogramas, o cálculo de comparação é realizado apenas sobre regiões onde segmentos candidatos foram detectados.

Com o uso da projeção por histogramas pretende-se oferecer um método de reconhecimento que se apoie em uma forma de representação, que torne mais expressiva as características da forma do objeto de interesse. Contudo, dois objetos distintos podem produzir um histograma semelhante. Por exemplo, uma seta orientada para cima produziria uma projeção vertical semelhante a uma seta orientada para baixo.

Uma maneira de contornar tal situação seria fazer a comparação das projeções verticais e horizontais do *template* com as formas candidatas. Nessa abordagem, em uma etapa inicial, uma das projeções seria utilizada na comparação. Os segmentos que possuem um critério de similaridade alto seriam então sujeitados a uma segunda comparação com a projeção restante.

É interessante salientar que a abordagem do reconhecimento de objetos pelo uso de projeções por histograma, obteve um desempenho significativamente maior do que a técnica de *template matching*. O desempenho é um requisito importante para aplicações de tempo real, principalmente no contexto de robótica móvel.

Apesar da técnica *template matching* ter demonstrado maior precisão, a abordagem com o uso de projeções pode ser melhorada com o emprego de um algoritmo de segmentação mais robusto quanto à iluminação.

Trabalhos futuros devem ser realizados para o aprimoramento da abordagem aqui relatada, combinando outras técnicas ou metodologias de processamento de imagens e reconhecimento de objetos. A abordagem de reconhecimento de objetos apresentada nesse trabalho está em fase de avaliação, a plataforma robótica adotada para avaliar os algoritmos implementados é o *RoboDeck*, da empresa brasileira *Xbot* ([www.xbot.com.br](http://www.xbot.com.br)).

## REFERÊNCIA BIBLIOGRÁFICA

- AMIT, Y.. 2D Object Detection and Recognition: Models Algorithms, and Networks. Massachusetts: The MIT Press, 2003.
- BRADSKI, G.; KAEHLER, A.. OpenCV: Computer Vision with OpenCV Library. California: O'Reilly Media, Inc., 2008.
- BURGER, W.; BURGE, M. J.. Principles of Digital Image Processing: Fundamental Techniques. London: Springer, 2009.

CHAN, F. H. Y.; LAM, F. K.; ZHU, H.. Adaptive Thresholding by Variational Method. In: IEEE Transactions on Image Processing. Vol. 7, No. 3, 1998.

CHANG, P.; KRUMM, J.. Object Recognition with Color Cooccurrence Histograms. In: Computer Vision and Pattern Recognition - CVPR , pp. 2498-2504, 1999.

DOS SANTOS, R. P.; CLEMENTE, G. S.; REN, T. I.; CAVALCANTI, G. D. C.. Text Line Segmentation Based on Morphology and Histogram Projection. In: International Conference on Document Analysis and Recognition - ICDAR, pp. 651-655, 2009.

FORSYTH, D. A.; PONCE, J.. Computer Vision: A Modern Approach. New Jersey: Prentice Hall, 2003.

JAIN, R.; KASTURI, R.; SCHUNK, B. G.. Machine Vision. New York: McGraw-Hill, 1995.

PEDRINI, H.; SCHWARTZ, W. R.. Análise de Imagens Digitais – Princípios, Algoritmos e Aplicações. São Paulo: Thomson Learning, 2008.

RUSS, J. C.. The Image Processing Handbook. Sixth Ed. New York: CRC Press, 2011.

TREIBER, M.. An Introduction to Object Recognition. New York: Springer, 2010.