

Classificação de objetos em imagens onidirecionais com uso de retificação de imagens e de múltiplos núcleos em máquinas de vetor de suporte <sup>1</sup>

**Author(s):**

Fábio R. Amaral

Anna H. R. Costa

---

<sup>1</sup>This work was supported by Fapesp Project LogProb, grant 2008/03995-5, São Paulo, Brazil.

# Classificação de objetos em imagens onidirecionais com uso de retificação de imagens e de múltiplos núcleos em máquinas de vetor de suporte

Fábio R. Amaral, Anna H. R. Costa  
Laboratório de Técnicas Inteligentes (LTI)  
Escola Politécnica da Universidade de São Paulo  
São Paulo, SP, Brazil  
fabioramaral@usp.br, anna.reali@poli.usp.br

## Resumo

*Apesar da popularidade das câmeras onidirecionais aplicadas na robótica móvel e da importância do reconhecimento de objetos no universo mais amplo da robótica e da visão computacional, é difícil encontrar trabalhos que relacionem ambos na literatura especializada. Este trabalho visa desenvolver um sistema eficaz e eficiente de classificação de objetos em imagens onidirecionais. Para isso, propõe o uso conjugado de um processo de retificação da imagem onidirecional e de múltiplos núcleos em um classificador baseado em máquinas de vetor de suporte. Resultados experimentais atestam a viabilidade da proposta.*

## Abstract

*Despite the popularity of omnidirectional cameras used in mobile robotics and the importance of object recognition in the broader universe of robotics and computer vision, it is difficult to find works that relate both in the literature. This work aims at developing an effective and efficient classifier of objects in omnidirectional images to be applied to tasks of self-localization and environment mapping made by mobile robots. For this we propose the combined use of a process for unwrapping omnidirectional images and a classifier based on a multiple kernels support vector machines. Experimental results show the advantage of such proposal.*

## 1. Introdução

O reconhecimento de objetos por sistemas computacionais autônomos pode tornar possível um número muito grande de tarefas, além de contribuir para o aprimoramento

de várias outras já desenvolvidas. Tarefas mais complexas que exijam interação com o ambiente ou com seres humanos dependem, em sua grande maioria, de tal habilidade que, por sua vez, depende de informações obtidas previamente ou mesmo observadas em tempo real, sobre o ambiente em questão. Neste panorama, a visão computacional pode oferecer um bom conjunto de informações do ambiente, uma vez que busca a construção de descrições explícitas e significantes de objetos físicos a partir de imagens [2], ou seja, almeja transformar a informação visual, obtida através de sensores visuais (câmeras digitais), em informação útil para tais sistemas autônomos.

Na literatura relacionada ao reconhecimento de objetos, há dois tipos distintos de tarefas: a classificação e a detecção. Na classificação, o objetivo é avaliar na imagem de teste se a mesma contém alguma instância do objeto de interesse, sem a necessidade de indicar sua localização. A tarefa de detecção, por sua vez, tem o objetivo de verificar a existência do objeto de interesse, bem como indicar sua localização na imagem. A classificação dos objetos é frequentemente realizada através de técnicas de aprendizagem de máquinas, onde padrões são aprendidos a partir de imagens de treinamento rotuladas e então reconhecidos em imagens de teste. Uma técnica amplamente adotada é a Máquina de Vetor de Suporte (MSV). Esta técnica foi inicialmente proposta por Vapnik [17] e atualmente tem sido aplicada com sucesso em diversos problemas de classificação e regressão.

Basicamente, uma MSV é uma máquina linear cuja idéia principal consiste em construir um hiperplano como superfície de decisão, de tal forma que seja máxima a margem de separação entre os exemplos rotulados como positivos e negativos. As MSVs podem ser aplicadas na classificação de padrões tanto linearmente separáveis quanto linearmente não separáveis, sendo que, para o segundo caso, uma etapa inicial é realizada, para mapear os vetores de entrada não linearmente separáveis, para um espaço de características de alta dimensionalidade, onde os padrões passam a ser lin-

earmente separáveis com alta probabilidade. A construção do hiperplano de decisão é então realizada neste novo espaço de características e não mais no espaço de entrada original [8].

Outro ponto de discussão na construção de sistemas de visão computacional é a escolha do modelo de câmera, que depende da aplicação e do tipo de informação que se deseja obter dela. Aplicações que demandam a informação de profundidade, por exemplo, normalmente fazem uso de sistemas de visão estereoscópica, compostos por duas câmeras alinhadas de maneira que a diferença entre as imagens permita a inferência da topografia da cena e, por consequência, a representação tridimensional dos objetos. Na robótica móvel, a possibilidade da visão do ambiente todo ao redor do robô tem enorme valor. Tal visualização do ambiente é alcançada com as câmeras onidirecionais. A tarefa de SLAM, do inglês *Simultaneous Localization And Mapping*, é beneficiada com o uso deste sistema de visão, pois marcos visuais de todo o ambiente podem ser obtidos na cena ao longo do percurso do robô, alimentando a construção de um mapa mais detalhado do ambiente. Novas abordagens para o SLAM como em [14] exploram não mais os marcos visuais genéricos do ambiente para a construção do mapa, mas sim objetos identificados no ambiente. Nesta linha, o presente trabalho endereça a tarefa de classificação de tais objetos na cena.

Apesar da popularidade das câmeras onidirecionais aplicadas à robótica móvel e da importância do reconhecimento de objetos no universo mais amplo da robótica e da visão computacional, é difícil encontrar trabalhos que relacionem ambos na literatura especializada. Este trabalho visa propor e avaliar um método para classificação de objetos em imagens onidirecionais, analisando sua eficácia e eficiência, quando uma etapa de retificação da imagem onidirecional é previamente aplicada ao processo de detecção de características. Estas são então submetidas a um classificador de objetos baseado em MSVs estendidas para a utilização de múltiplos núcleos. A etapa de retificação é aplicada às imagens onidirecionais com o objetivo de aproximá-las das imagens convencionais, com as quais o classificador utilizado já demonstrou bons resultados [19]. A abordagem de múltiplos núcleos se faz necessária para possibilitar a utilização de três tipos distintos de detectores de características em imagens, ponderando, para cada classe, a importância de cada uma das características em sua descrição. Resultados experimentais atestam a viabilidade de tal proposta.

O texto está organizado da seguinte forma. Inicialmente, o método proposto é detalhado, sendo descritos na Seção 2 os processos de retificação de imagens e de detecção de características, e o classificador na Seção 3. Na Seção 4 são apresentados os experimentos conduzidos e seus resultados e, finalmente, as conclusões e trabalhos futuros são descritos na Seção 5.

## 2. Retificação e Detecção de Características

Três tarefas comuns à maioria dos classificadores podem ser definidas como a detecção das características da imagem, a aprendizagem e a classificação propriamente dita.

Uma visão geral da arquitetura do sistema proposto é mostrada na Figura 1. Nesta proposta, recomenda-se uma etapa prévia que realiza a retificação da imagem onidirecional, com o objetivo de aproximá-la a uma imagem em perspectiva. A seguir, efetua-se a detecção de características na imagem retificada, constituindo os dados de entrada para o classificador. Inicialmente, há uma fase de treinamento do classificador, onde um processo de aprendizado supervisionado é conduzido em um conjunto amostrado de dados de entrada com seus respectivos rótulos das classes corretas atribuídas por um supervisor, para que o mesmo aprenda a classificar corretamente os dados de entrada do treinamento, os quais correspondem às características extraídas nas imagens retificadas. Na fase de execução, que ocorre após o aprendizado do classificador, as imagens de teste são retificadas, suas características são extraídas e o classificador as rotula, de acordo com o aprendido na fase de treinamento. Os processos de retificação da imagem onidirecional e de detecção de características são descritos a seguir.

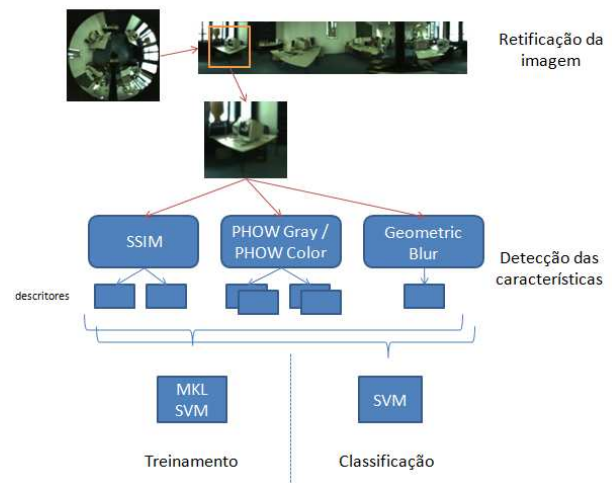


Figura 1. Arquitetura do sistema proposto.

### 2.1. Retificação da Imagem Onidirecional

A retificação da imagem onidirecional segue o método proposto em [12], mostrado na Figura 2. A partir da imagem original (quadro superior esquerdo), inicialmente detecta-se bordas que delimitam variações bruscas na intensidade da imagem (quadro central superior). As bordas são proces-

sadas de forma a indicarem o ponto central da imagem onidirecional (quadro superior direito), permitindo a conversão do sistema polar para o sistema cartesiano, no qual é representada a imagem convertida, que corresponde à imagem retificada (quadro inferior). Neste processo, de modo a melhorar a qualidade da imagem retificada, efetua-se um processo de interpolação, onde regiões previamente não existentes na imagem onidirecional são preenchidas de forma adequada.

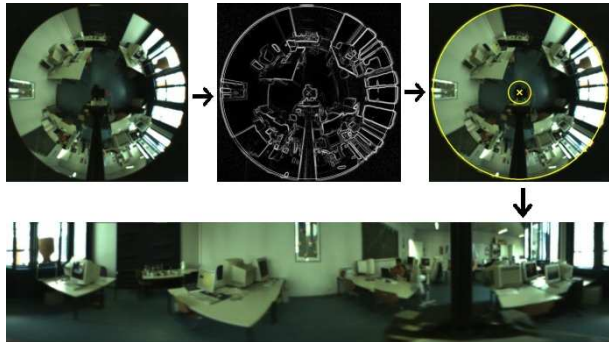


Figura 2. Processo de retificação da imagem onidirecional. Extraída de [12].

## 2.2. Detecção das Características na Imagem

Após a retificação, as imagens são submetidas a três processos diferentes que detectam tipos distintos de características utilizadas no processo de classificação. Estas características são *Geometric Blur* (GB), *Self-Similarity* (SSIM) e *PHOWGray* / *PHOWColor*, descritas a seguir.

**2.2.1. Geometric Blur** Este detector de características foi originalmente proposto em [4], com o objetivo de criar um descritor invariável a distorções geométricas para pontos de interesse na imagem, ou seja, invariável a transformações afim dos objetos na cena. O processo consiste em "borrar" (*to blur*, em inglês) a imagem padrão e então efetuar a correspondência entre pontos de interesse do padrão e pontos encontrados em partes da imagem teste, em diferentes resoluções. A idéia explorada pelos autores [4, 3] é que o borramento para estabelecer a correspondência deve ser pequeno perto dos pontos em correspondência e grande quando longe deles. Assim, para cada ponto de interesse encontrado, constrói-se o descritor GB, como ilustrado na Figura 3, onde o ponto extremo superior esquerdo do "T" é o ponto de interesse a ser descrito; a imagem é borrada com um operador gaussiano com borramento aplicado em função da distância do ponto de interesse e os demais pontos vizinhos mostrados na figura também terão seus val-

ores amostrados para a formação do descritor do ponto em questão. Este processo é repetido para a descrição de todos os pontos de interesse da imagem padrão, sempre a partir da imagem original, ou seja, o borramento realizado para descrição de um ponto não se acumula para os demais. Na fase de correspondência, estes descritores armazenados são comparados com os calculados nas partes consideradas da imagem teste, buscando aqueles com melhor correspondência.

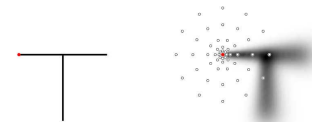


Figura 3. Exemplo de construção do descritor por *Geometric Blur*. Extraída de [3].

**2.2.2. Self-Similarity** O detector de características de auto-similaridade (SSIM, do inglês *Self-Similarity*) proposto em [15], visa definir um modelo para encontrar similaridade geométrica entre as regiões, independente de outras características como textura, luminosidade ou cor, e está intimamente relacionado com a noção de co-ocorrência estatística de intensidades de pixels entre regiões de uma imagem. A idéia consiste em associar um descritor SSIM  $dq$  a cada pixel de interesse  $q$  da imagem de entrada. Isso é feito correlacionando (por exemplo, usando a medida SSD de *Sum of Square Differences*) uma pequena região centrada em  $q$  (no caso, uma janela  $5 \times 5$ ) com uma região maior também centrada em  $q$  (no caso, uma janela  $40 \times 40$ ), resultando em uma superfície local de correlação, conforme ilustra a Figura 4. A superfície de correlação é então transformada em uma representação de coordenadas log-polar centradas em  $q$  e dividida em partes com valores normalizados (no caso, 80 partes, resultantes de 20 variações angulares e 4 divisões em cada fatia), a qual constitui o descritor  $dq$  de  $q$ , estirado em um vetor (extremo direito da figura). Um conjunto de descritores consecutivos é armazenado como modelo, chamado de Palavra Visual, e buscado em imagens teste, que também extraem tais descritores e os correspondem aos do modelo.

**2.2.3. PHOWGray / PHOWColor** O detector de características *Pyramid Histogram Of visual Words* (PHOW) teve seu conceito inicialmente proposto em [10] porém foi efetivamente definido em [5]. Ele faz uso de descritores SIFT extraídos densamente, ou seja, em uma grade regular a cada  $N$  pixels (para este caso  $N$  é 5). Após esta etapa, os descritores SIFT obtidos são agrupados em grupos (no caso,

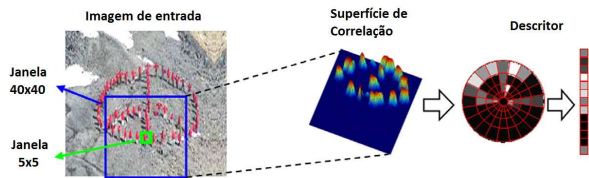


Figura 4. Criação do descritor de auto-similaridade [15].

300) através do processo K-means [11]. Após o agrupamento, cada descritor passa a ter o valor médio de seu grupo, formando as Palavras Visuais. Com o processo de quantização completo, é iniciada a construção do descritor PHOW. A idéia é dividir a imagem em partes iguais e, para cada parte, somar as ocorrências de cada Palavra Visual. Neste caso, a imagem é dividida duas vezes, em uma matriz  $2 \times 2$  e em outra, com  $4 \times 4$  regiões. O descritor PHOW é construído concatenando os valores acumulados para cada Palavra Visual em cada divisão da imagem, conforme está ilustrado na Figura 5. Nela, as Palavras Visuais são representadas pelas figuras bolas pretas, cruces e losangos. Como as imagens usadas são coloridas, dois descritores PHOW são obtidos, um apenas para as intensidades, ou níveis de cinzas, PHOWGray, e o outro para os canais de cores HSV, PHOWColor.

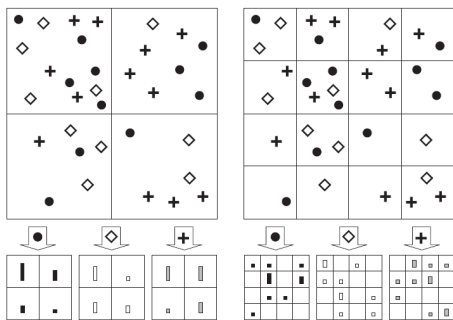


Figura 5. Construção do descritor PHOW [10].

### 3. Máquinas de Vetor de Suporte de Múltiplos Núcleos

Abordagens utilizando Aprendizagem de Múltiplos Núcleos (do inglês *Multiple Kernel Learning* (MKL)) são relativamente recentes [1, 9, 16], no entanto têm despertado grande interesse na sua utilização devido aos bons resul-

tados divulgados [19] mas principalmente pela sua proposta, que pode ser aplicada em inúmeras áreas de pesquisa e desenvolvimento, para tarefas de classificação de dados. O conceito de Múltiplos Núcleos pode ser aplicado não só para as MSVs, mas também para outros algoritmos de aprendizagem de máquinas, como Análise dos Componentes Principais (PCA), agrupamento de dados, Redes Neurais, etc.

Pode-se extrair de imagens uma grande quantidade de características como bordas, cantos, cores, intensidades, entre muitas outras. Cada característica possui um grau de importância para ajudar na representação de um dado objeto de interesse, a ser pesquisado nas imagens. A utilização do MKL permite combinar tais características de maneira ponderada para representar as classes de objetos a serem detectados. A ponderação aplicada a esta combinação pode ser interpretada como o equilíbrio entre as características necessárias para a melhor representação de uma categoria de objetos.

As MSVs, quando aplicadas a dados linearmente não-separáveis, são formuladas apenas através da função Núcleo  $K$ , deixando implícito o mapeamento dos dados de entrada, do espaço linearmente não separável para o espaço linearmente separável de alta dimensionalidade. A abordagem MKL realiza a combinação linear de um Núcleo para cada tipo de característica extraída da imagem. O problema MKL pode ser formulado como proposto em [18]. Considerando um conjunto de  $M$  tipos de características para serem combinadas, cada uma contendo  $N$  amostra rotuladas de treinamento  $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$ , onde  $\mathbf{x}_i$  representa a amostra e  $d_i$  o rótulo associado, teremos para cada característica, um Núcleo  $K_m$ . O objetivo é realizar a combinação linear destes Núcleos em uma matriz de Núcleos ótima  $\mathbf{K}_o = \sum_{m=1}^M \eta_m K_m$ , onde  $\eta_m$  é o peso atribuído ao Núcleo  $K_m$ . Adaptando o funcional de custo da MSV, a ser minimizado, tem-se:

$$T(\eta) = \min_{\mathbf{w}, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i + \gamma^T \eta \quad (1)$$

sujeito às restrições

$$\begin{aligned} d_i(\mathbf{w}^T \varphi(\mathbf{x}_i) + b) &\geq 1 - \xi_i & \text{para } i = 1, 2, \dots, N \\ \xi_i &\geq 0 & \text{para todo } i \end{aligned}$$

onde  $C$  é um parâmetro positivo,  $\xi_i$  é uma variável solta que permite a suavização da margem da MSV,  $\mathbf{w}$  é o vetor de pesos, e  $\gamma$  é o vetor de inicialização para o vetor de pesos  $\eta$  que pondera os núcleos  $K$ . Na restrição,  $b$  representa o viés do hiperplano de decisão da MSV e o termo  $\varphi$  representa o conjunto de transformações não lineares do espaço de entrada para o espaço de características aplicado às amostras  $\mathbf{x}_i$ .

Usando os multiplicadores de Lagrange, o problema dual pode ser formulado:

$$W(\eta) = \max_{\alpha} \sum_{i=1}^N \alpha_i + \gamma^T \eta - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \sum_{m=1}^M \eta_m K_m(\mathbf{x}_i, \mathbf{x}_j) \quad (2)$$

sujeita às restrições

- $\sum_{i=1}^N \alpha_i d_i = 0$
- $0 \leq \alpha_i \leq C$  para  $i = 1, 2, \dots, N$

onde  $\alpha_i$  são os multiplicadores de Lagrange e  $\gamma$  é o vetor de inicialização para os pesos  $\eta$ .

Pelo princípio da dualidade  $T(\eta) = W(\eta)$ . Além disso, sendo  $\alpha$  a solução única que maximiza a função objetivo  $W(\eta)$ , esta pode ser diferenciada em função de  $\eta$  como segue:

$$\frac{\partial T}{\partial \eta_m} = \frac{\partial W}{\partial \eta_m} = \gamma_m - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j K_m(\mathbf{x}_i, \mathbf{x}_j) \quad (3)$$

Em uma MSV de núcleo único, o objetivo da etapa de aprendizagem é determinar o valor do vetor de pesos ótimo  $\mathbf{w}_o$ , que definem o hiperplano ótimo de separação. Para o caso da MSV de Múltiplos Núcleos, além do valor do vetor de pesos  $\mathbf{w}_o$ , deve-se determinar o valor do vetor de pesos  $\eta$ , que pondera a combinação linear dos núcleos  $K(\mathbf{x}_i, \mathbf{x}_j)$  em uma matriz de núcleos ótima  $\mathbf{K}_o = \sum_{m=1}^M \eta_m K_m$ .

A aprendizagem é realizada em duas etapas. Na primeira etapa os valores de  $\eta$  e  $K_m$  são fixados. A função objetivo  $W$  pode ser maximizada (Equação 2) como numa MSV de Núcleo único  $K_m$ . Para este problema de otimização, algoritmos como [1] podem ser usados, contudo, outras propostas podem ser adotadas. Na segunda etapa  $T(\eta)$  é minimizado usando o método do gradiente descendente na Equação 3. As duas etapas se repetem até a convergência ou até o número de repetições máximo ser atingido.

Como a estratégia aqui utilizada faz uso da abordagem Um-Contra-Todos para operar com múltiplas classes, deve-se treinar uma MSV para cada uma das classes. O dado de entrada é testado em todas as MSVs e a classe com maior número de votos é dada como a classe do objeto testado. Observando individualmente, em uma MSV treinada para diferenciar entre sua classe e as demais, o processo de classificação é baseado em

$$\text{sgn}(\mathbf{w}^T \mathbf{x} + b),$$

onde o sinal  $\text{sgn}$  retornado representa o lado do hiperplano de separação em que o vetor de entrada  $\mathbf{x}$  se enquadra. De

acordo com este valor, a classe é atribuída ou não ao dado testado.

## 4. Resultados

As imagens para a realização dos experimentos foram obtidas de quatro conjuntos de dados diferentes. Para as imagens em perspectiva, os conjuntos de dados Caltech [7] e Pascal VOC 2008 [6] foram utilizados. Para as imagens onidirecionais, foram utilizados os conjuntos COLD [13] e Cogniron [20]. A partir deste conjunto de imagens onidirecionais, um conjunto de imagens onidirecionais retificadas foi produzido através do processo descrito na Seção 2.2. Com isso, os objetos observados nos testes serão os mesmos, apenas diferenciados pela imagem de onde foram obtidos.

Todos os experimentos foram realizados no ambiente computacional Matlab 2009. Em todos os conjuntos de dados, imagens representando quatro objetos distintos foram selecionadas: monitor, sofá, máquina copiadora e extintor de incêndio. Estes objetos foram escolhidos por serem encontrados tanto nas imagens convencionais quanto nas onidirecionais das bases de dados, além de serem encontrados em ambientes internos nos quais robôs móveis se deparam, facilitando estudos posteriores.

O treinamento do modelo foi realizado com as imagens convencionais dos objetos avaliados, utilizando 15 amostras para cada tipo de objeto. Os testes foram realizados com este modelo treinado aplicado sobre as imagens onidirecionais e retificadas, de modo a avaliar possíveis benefícios inerentes da etapa de retificação proposta neste trabalho. Os resultados dos testes são mostrados na Tabela 1.

Objeto	onidirecional	retificada
Extintor	33,3%	46,7%
Copiadora	63,3%	90,0%
Sofá	94,4%	77,8%
Monitor	0	33,3%

**Tabela 1. Comparação dos valores de revocação de objetos para imagens onidirecionais e retificadas**

A exatidão média do classificador aplicado diretamente às imagens onidirecionais é de 42,0%. Quando aplicada a etapa prévia de retificação das imagens onidirecionais a exatidão média do classificador sobe para 49,33%. É possível observar uma melhora considerável nos resultados obtidos sobre as imagens retificadas. Individualmente, apenas para a categoria sofá não houve melhora no resultado, podendo ser

atribuído este fato à pequena quantidade de imagens desta categoria obtidas para a realização dos testes.

## 5. Conclusões e Trabalhos Futuros

Neste trabalho foi proposto um sistema para a integração de duas importantes tecnologias da robótica e da visão computacional, o reconhecimento de objetos e as imagens onidirecionais. Um método baseado em MSVs utilizando múltiplos núcleos, já aplicado com sucesso no reconhecimento de objetos em imagens convencionais, foi aqui utilizado como base para o desenvolvimento do sistema proposto. A introdução de uma etapa, computacionalmente simples, para a retificação da imagem onidirecional foi aplicada de maneira a aproximar a imagem onidirecional das imagens convencionais, visando também reduzir as distorções causadas pelos espelhos utilizados na construção das câmeras onidirecionais. É sabido que descritores de características observadas nas imagens, como o SIFT e outros baseados neste modelo, não são invariantes ao modelo de câmera adotado; por este motivo a etapa de retificação permitiu uma melhora considerável nos resultados de classificação dos objetos aqui adotados para avaliação do modelo.

Para a evolução deste trabalho, a utilização de uma matriz híbrida para um melhor mapeamento entre as imagens onidirecionais e em perspectiva, complementando a etapa de retificação, deve ser avaliada. Além disso, novos tipos de características ou descritores, com melhor invariância às distorções causadas pelos espelhos utilizados na construção das câmeras onidirecionais, devem ser buscados. Com o objetivo de garantir a aplicabilidade deste modelo em tarefas da robótica móvel, avaliações de desempenho em sistemas SLAM para navegação robótica também precisam ser realizadas.

## Agradecimentos

Os autores agradecem ao CNPq Proc. N. 475690/2008-7 e N. 305512/2008-0 e à FAPESP Proc. N. 2008/03995-5 pelo apoio recebido.

## Referências

- [1] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proc. of the 21st International Conference on Machine Learning (ICML'04)*, page 6, New York, NY, USA, 2004. ACM.
- [2] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, April 1982.
- [3] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 26–33. IEEE, 2005.
- [4] A. C. Berg and J. Malik. Geometric blur for template matching. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, pages 1–607–1–614 vol.1. IEEE, 2001.
- [5] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proc. of the IEEE 11th International Conference on Computer Vision (ICCV'07)*, pages 1–8. IEEE, 2007.
- [6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>.
- [7] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
- [8] S. Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 2 edition, July 1998.
- [9] G. Lanckriet, N. Cristianini, P. Bartlett, and L. E. Ghaoui. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.
- [10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. of the 2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 2169–2178. IEEE, 2006.
- [11] D. J. C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 1st edition, June 2002.
- [12] A. Pronobis. Unwrapping omnidirectional images. <http://www.pronobis.pro/software/unwrap>, 2009.
- [13] A. Pronobis and B. Caputo. COLD: COsy Localization Database. *The International Journal of Robotics Research (IJRR)*, 28(5), May 2009.
- [14] A. Selvatici, A. H. R. Costa, and F. Dellaert. Object-based visual slam: How object identity informs geometry. In *Workshop de Visão Computacional (WVC)*, volume 1, pages 82–87, 2008.
- [15] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, pages 1–8. IEEE, 2007.
- [16] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.
- [17] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [18] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proc. of the International Conference on Computer Vision (ICCV'07)*, pages 1–8, 2007.
- [19] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proc. of the International Conference on Computer Vision (ICCV'09)*, 2009.
- [20] Z. Zivkovic, O. Booij, B. J. A. Kröse, E. A. Topp, and H. I. Christensen. From sensors to human spatial concepts: An annotated data set. *IEEE Transactions on Robotics*, 24(2):501–505, 2008.