

Experimentos em Aprendizado de Máquina para Fusão de Sensores

Katti Faceli, André C. P. L. F. de Carvalho, Solange O. Rezende
Departamento de Ciências da Computação e Estatística
Instituto de Ciências Matemáticas e de Computação
USP - São Carlos, SP, Brasil
E-mails: {katti, andre, solange}@icmc.sc.usp.br

Abstract

Robot navigation systems need accurate distance measures to describe the robot environment (distances to target and obstacles). These measures are acquired by sensors, which usually do not have the necessary precision. In order to obtain more accurate distance measures, this paper investigates the behaviour of machine learning techniques for sensor fusion. For such, Multi-Layer Perceptron networks and Support Vector Machines are used. The accuracy of the distances measured by the fusion models are compared to those achieved by the individual sensors.

1. Introdução

Muitos sistemas atualmente interagem com o ambiente através de dispositivos que são capazes de captar alguma informação desse ambiente. Porém, em geral esses dispositivos, chamados de sensores, têm suas restrições, fornecendo muitas vezes dados incompletos, inconsistentes ou imprecisos. Dentre esses sistemas estão os robôs móveis autônomos que captam do ambiente, entre outras informações, a distância que o robô se encontra de algum obstáculo ou alvo.

Esse trabalho investiga a utilização de diferentes técnicas de aprendizado de máquina (AM) para fusão de sensores, com o objetivo de obter distâncias mais precisas e acuradas entre um robô e um obstáculo ou alvo, do que aquelas adquiridas diretamente pelos sensores. Os algoritmos de AM investigados são *Support Vector Machines* (SVM) [1] e redes neurais do tipo *Multi-Layer Perceptron* (MLP) [2, 3].

Este artigo está organizado da seguinte forma. Na Seção 2 é feita uma breve apresentação sobre fusão de sensores. Na seção 3 são descritos os métodos utilizados nas investigações: SVM e redes MLP. A Seção 4 contém uma descrição dos dados utilizados. Na Seção 5 são descritos os experimentos realizados bem como os resultados obtidos. Por fim, na Seção 6 são apresentadas as conclusões desse trabalho.

2. Fusão de Sensores

Diversas áreas da ciência e engenharia precisam de sistemas que capturem, processem e integrem

informações provenientes de várias fontes [4]. Muitos destes sistemas interagem com o mundo real e requerem informações precisas e confiáveis do ambiente ao seu redor. As fontes de informação utilizadas por esses sistemas geralmente fornecem dados incompletos, inconsistentes ou imprecisos. Um caso específico desse tipo de sistema são os robôs móveis autônomos. Esses robôs têm como tarefa essencial navegar por um ambiente dinâmico, onde devem realizar alguma atividade. Para isso, coletam informações do ambiente através de diversos tipos de sensores, utilizando-as para criar modelos do mundo à sua volta e acompanhar as mudanças que nele ocorrem [5, 6].

Diferentes tipos de sensores têm sido desenvolvidos com diferentes vantagens e desvantagens. Para selecionar o melhor sensor para uma determinada aplicação, deve-se levar em consideração suas características, tais como campo de visão (alcance), faixa de operação (flexibilidade), precisão e resolução, habilidade de detectar todos os objetos no ambiente, concisão e facilidade na interpretação dos dados, redundância, simplicidade (facilidade de operação, manutenção), consumo de energia, tamanho, calibração, confiabilidade e custo [5, 7].

Muitas vezes, algumas características necessárias para uma aplicação, como boa precisão e confiabilidade, ficam limitadas por restrições de custo ou complexidade dos sensores.

Um dado importante obtido através de sensores nos robôs móveis é a distância entre o robô e um objeto qualquer que ele encontre no caminho ou um alvo que ele tenha que atingir. Porém, as distâncias obtidas através de um sensor utilizado isoladamente muitas vezes são imprecisas e inacuradas e não se adequam à utilização em atividades que necessitem de medidas precisas e acuradas [5, 8].

A técnica mais comumente empregada hoje em dia para medir distâncias em robôs móveis que navegam em ambientes fechados é a TOF (*Time of Flight*) com ultrassom. Sua popularidade se deve a seu baixo custo e facilidade de utilização [7]. Porém, esse tipo de sensor apresenta restrições com respeito à confiabilidade. Se forem feitas duas leituras seguidas do sensor, sem que sua posição tenha sido modificada, essas leituras provavelmente serão diferentes. Esse erro não é sistemático, mas esse tipo de sensor também está sujeito a um erro sistemático que se manifesta através de erros na leitura quando ele está inclinado em relação à sua linha de visão

(quanto maior o ângulo formado entre a frente do sensor e o objeto a ser medido, maior tende a ser o erro na leitura). Além disso, existem superfícies que absorvem a radiação do SONAR¹, fazendo com que esse tipo de sensor não as detecte.

Uma alternativa que vem se mostrando viável para a obtenção de informações de distância é a utilização de técnicas simples de processamento de sinais visuais, tais como *Visual Looming* [9, 10], *Visual Sonar* baseado na proposta de Horswill [11] e métodos que usam diversas formas de triangulação ativa ou passiva [7]. Entretanto, sensores baseados em informações visuais também têm suas restrições. Exemplos destas restrições são: a precisão pode ser reduzida com o aumento da distância e a presença de sombras de objetos e mudanças nos padrões das superfícies podem ser confundidas com objetos.

Outras alternativas de sensores podem obter os mesmos dados, inclusive com maior precisão e acurácia, porém associadas a um custo mais elevado.

Geralmente, as restrições apresentadas por um tipo de sensor não se aplicam a outros e vice-versa. Nesses casos, a combinação das informações obtidas a partir desses sensores pode fornecer uma informação mais completa, precisa e acurada do ambiente.

A fusão de sensores [4, 12, 13, 14, 15, 16] envolve justamente a combinação de informações provenientes de diferentes sensores para captar dados do ambiente cuja obtenção está além da capacidade isolada de cada um dos sensores envolvidos, principalmente quando são consideradas acurácia e precisão. Além disso, a fusão de sensores tanto torna o sistema mais tolerante a falhas, como também pode fornecer informações novas, que nenhum dos sensores isoladamente consegue obter.

Dentre os métodos mais comumente utilizados para fusão de sensores, podem ser citados [12, 14, 17]: Técnicas de Probabilidade e Inferência Bayesiana, Teoria de Dempster-Shafer, Lógica Fuzzy e Redes Neurais Artificiais.

Esses métodos geralmente executam um mapeamento que reduz a dimensão dos dados originados de múltiplas entradas (leituras brutas dos sensores, valores de pixels, características extraídas, avaliadores de sinais, ou sinais de controle) para um pequeno número de saídas (estado estimado, objetos ou eventos reconhecidos, características realçadas e sinais de controle).

Em trabalho anterior já foram investigados outros modelos de redes neurais, bem como um método de aprendizado simbólico para fusão de sensores aplicada ao mesmo problema descrito nesse artigo [18]. O trabalho de Martens et. al [8] também envolve a aplicação de redes neurais para fusão de sensores objetivando a melhoria da qualidade das distâncias entre um robô e um obstáculo, obtidas a partir de sensores simples e baratos. Pasika [19] utiliza redes neurais e SVM para, a partir de diferentes tipos de sensores já instalados, obter uma estimativa da

altura das nuvens correspondente à dada por um LIDAR (Laser Radar), que é um dispositivo caro e que não se encontra disponível em muitas regiões.

Neste trabalho, a fusão de sensores é empregada como forma de melhorar a precisão e acurácia da distância medida entre um conjunto de sensores (em um robô) e um objeto. Para isso foram utilizadas as técnicas de aprendizado de máquina: redes neurais artificiais (*Multi-Layer Perceptron* - MLP) e *Support Vector Machines* (SVM).

3. Métodos utilizados

Nesta Seção são descritos os algoritmos de AM utilizados nesse trabalho: SVM e redes neurais do tipo MLP.

SVM [1] engloba uma classe de algoritmos de aprendizado e se baseia na teoria de aprendizado estatístico, combinando controle de generalização com uma técnica para tratar o problema da dimensionalidade. Sua formulação resulta em um problema de otimização quadrática e envolve o princípio de Minimização do Risco Estrutural (*Structural Risk Minimization* - SRM), em vez do princípio de Minimização do Risco Empírico (*Empirical Risk Minimization* - ERM) empregado pelas redes neurais convencionais. SRM minimiza o "Erro de generalização" em oposição a ERM, que minimiza o erro nos dados de treinamento, dando ao algoritmo SVM uma maior capacidade de generalização.

Esse tipo de algoritmo foi originalmente desenvolvido para reconhecimento de padrões. Nesse algoritmo, uma fronteira de decisão é representada por um subconjunto dos exemplos de treinamento, tipicamente pequeno, chamado de vetores de suporte (*support vectors*). Para o caso de regressão, Vapnik desenvolveu uma função de perda chamada ε -insensitive loss function que não penaliza erros inferiores a algum $\varepsilon > 0$ escolhido previamente. Esse algoritmo estima a função

$$f(x) = (w \cdot x) + b, \text{ com } w, x \in R^N, b \in R \quad (1)$$

com base nos dados $(x_1, y_1), \dots, (x_l, y_l) \in R^N \times R$, minimizando

$$\frac{1}{2} \|w\|^2 + C \cdot R_{emp}^\varepsilon, \quad (2)$$

em que C é uma constante que faz um balanço entre minimizar os erros do treinamento e minimizar a complexidade do modelo, e

$$R_{emp}^\varepsilon = \frac{1}{l} \sum_{i=1}^l \max\{0, |y_i - f(x_i)| - \varepsilon\}. \quad (3)$$

Maiores informações sobre SVM podem ser encontradas no endereço <http://www.kernel-machines.org/> que contém diversos links para artigos e softwares.

A rede neural do tipo *Multi-Layer Perceptron* (MLP) [2, 3] é o modelo de rede neural mais utilizado em aplicações. Como o nome indica, nesta rede os neurônios

¹SONAR (sound navigation and ranging): determina a distância a um alvo medindo o tempo que uma onda ultra-sônica leva para viajar até o alvo e ecoar de volta a um receptor.

são distribuídos em pelo menos uma camada intermediária e uma camada de saída. O algoritmo mais comum utilizado para treinar redes com múltiplas camadas, como as redes do tipo MLP, é o algoritmo *backpropagation*. Esse algoritmo utiliza pares (entrada, saída desejada) para ajustar os pesos da rede através de um mecanismo de correção de erros. O treinamento através desse algoritmo ocorre em duas fases, cada uma percorrendo a rede em um sentido: fase *forward*, utilizada para definir a saída da rede para um dado padrão de entrada, e fase *backward*, utilizada para atualizar os pesos das conexões da rede através da saída desejada e da saída da rede. Existem diversas variações deste algoritmo que têm como objetivo acelerar o processo de treinamento e reduzir as taxas de erros obtidas.

4. Dados empregados

Para os experimentos realizados neste trabalho foram utilizados dados coletados por K. Faceli e F. R. Melfi e descritos em [18] e [20]. Esses dados foram coletados utilizando um robô móvel Pioneer 1 da *Real World Interface* e consistem de medidas de distâncias obtidas de sete sonares ultra-sônicos posicionados em diferentes ângulos na frente e laterais do robô e a partir de imagens obtidas com uma câmera posicionada sobre o robô.

As distâncias foram tomadas com o robô posicionado em 7 ângulos diferentes e a 9 diferentes distâncias com relação a um alvo (quadrado preto em imagem com fundo branco) posicionado em uma parede, conforme mostra o esquema da Figura 1.

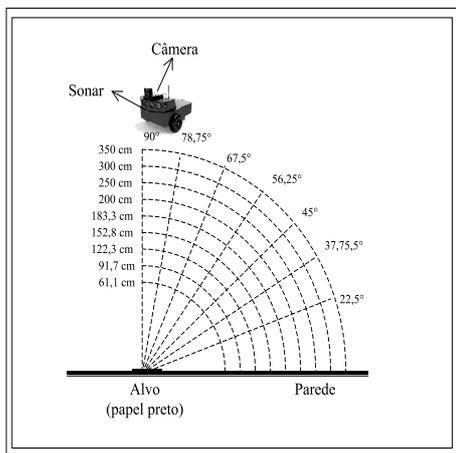


Figura 1: Distâncias em que foram coletados os dados dos sensores do robô Pioneer 1.

O arquivo gerado a partir dessas medidas contém as informações de ângulo (em graus), distância baseada nas imagens obtidas com a câmera, distâncias coletadas com os sonares e distância real para cada posição do esquema (cinco medidas dos sensores para cada posição). Portanto, essa base é constituída de 315 exemplos, cada um com 10 atributos.

Antes de serem utilizados, esses dados foram pré-processados. Inicialmente a medida de ângulo foi retirada, uma vez que foi obtida manualmente e não com um sensor. Assim, para o treinamento dos algoritmos foram utilizadas como dados de entrada as medidas obtidas com a câmera e com os 7 sonares do robô. A distância real medida manualmente serviu como saída desejada dos algoritmos. Em seguida os dados foram normalizados e divididos da seguinte maneira: 75% dos exemplos para treinamento e 25 % para teste. Para os experimentos com as redes neurais do tipo MLP os exemplos de treinamento foram subdivididos, ficando 50% do total de exemplos para treinamento e 25% para validação das redes.

Para obtenção de um resultado estatisticamente mais confiável, três partições diferentes com a divisão descrita anteriormente foram geradas. Cada resultado apresentado a seguir é formado pela média e desvio padrão dos resultados obtidos nas três partições.

5. Experimentos realizados e análise dos resultados

A avaliação dos resultados foi feita com base nos valores do MSE (*Mean Squared Error*) obtidos no treinamento, validação (redes) e teste dos algoritmos. As referências seguintes ao melhor algoritmo (rede MLP e SVM) se referem ao algoritmo em que o teste produziu, na média, o menor MSE.

Para as redes neurais do tipo MLP, foram feitos experimentos com diversas topologias e várias taxas de aprendizado (η) utilizando o algoritmo de aprendizado *backpropagation* padrão.

Para cada topologia testada, foram utilizadas taxas de aprendizado variando de 0,1 a 1,0. As topologias testadas apresentavam uma camada intermediária, 8 unidades de entrada e 1 neurônio de saída, variando apenas o número de neurônios na camada intermediária. Na camada intermediária foram colocados 1, 10, 50, 100 e 150 neurônios. Uma rede *perceptron* (sem camadas intermediárias) também foi treinada com esses mesmos parâmetros.

O gráfico da Figura 2 mostra os MSEs obtidos para as fases de treinamento, validação e teste da melhor rede (na média) treinada para cada topologia. Nesse gráfico, as barras representam a média do MSE para as três partições e as linhas sobre as barras representam o desvio padrão. O melhor resultado foi obtido pela rede com 10 neurônios na camada intermediária treinada com uma taxa de aprendizado igual a 0,6.

Para as SVM, foram realizados experimentos com 4 diferentes *kernels* e diversos valores para os parâmetros ajustáveis.

O gráfico da Figura 3 mostra o MSE obtido para os melhores modelos (na média) gerados para cada *kernel*. As barras no gráfico representam a média do MSE para as três partições e as linhas sobre as barras representam o

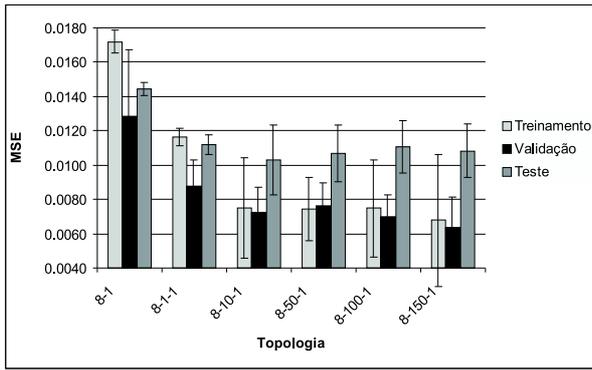


Figura 2: MSE da melhor rede MLP treinada para cada topologia.

desvio padrão. Nesse caso pode ser observado que o *kernel* que forneceu o modelo com melhor desempenho foi o gaussiano, embora o *kernel* polinomial também tenha apresentado bons resultados.

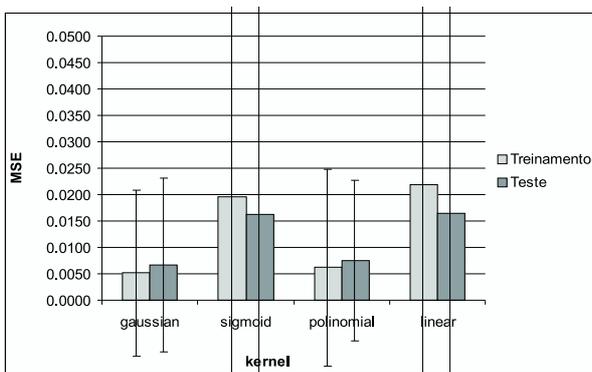


Figura 3: MSE para o melhor modelo (na média) gerado com cada *kernel*.

O gráfico da Figura 4 mostra um resumo das médias dos resultados obtidos no treinamento e teste dos diversos métodos de aprendizado investigados. A partir desse gráfico pode-se observar que, na fase de treinamento, algumas topologias de redes MLP obtiveram resultados semelhantes àqueles obtidos pelos melhores modelos gerados com o algoritmo SVM. Na fase de teste, porém, os modelos gerados com o SVM obtiveram desempenho superior. Esse resultado era o esperado, uma vez que as características do SVM dão a esse algoritmo uma melhor capacidade de generalização. Observa-se ainda que o melhor resultado, tanto no treinamento quanto no teste, foi obtido com o algoritmo SVM utilizando um *kernel* gaussiano. O MSE obtido com esse método foi de 0,00512 na fase de treinamento e 0,00673 na fase de teste.

Em seguida, todos os exemplos disponíveis foram submetidos ao melhor modelo obtido com cada algoritmo.

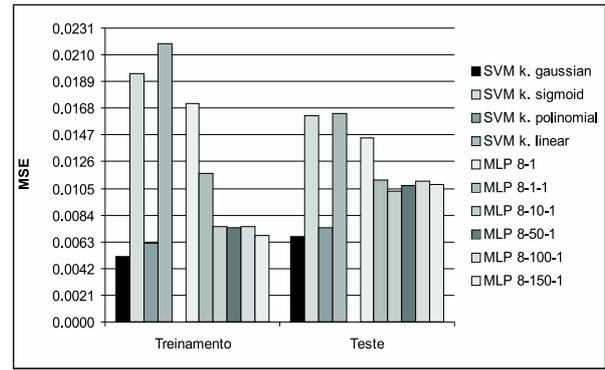


Figura 4: Comparação entre os métodos de aprendizado investigados.

mo. Com isso obteve-se estimativas das distâncias baseadas nesses modelos. De agora em diante a distância calculada por cada algoritmo será chamada simplesmente pelo nome do algoritmo. Para as análises seguintes foram observados principalmente os sensores Câmera e Sonar 4, uma vez que eles obtiveram as medidas mais precisas.

No gráfico da Figura 5 pode-se observar a variação entre a distância real e as distâncias estimadas pelos algoritmos MLP e SVM (distância MLP e distância SVM). Cada barra no gráfico representa o valor $|real - MLP| - |real - SVM|$ associado a cada ponto. Os valores acima de 0 representam os casos em que a diferença entre a distância real e distância MLP foi maior que a diferença entre a distância real e a distância SVM. Por esse gráfico verifica-se que a distância SVM se aproximou mais da real do que a distância MLP, pois a maioria das barras está acima de 0. Pode-se notar também que as diferenças entre a distância real e a distância SVM foi, em geral, menor em módulo, pois as barras abaixo de 0 são, em geral, menores que as barras acima de 0.

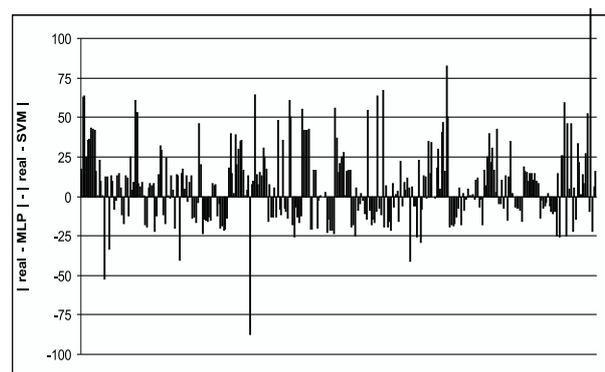


Figura 5: Comparação entre a variação das distâncias MLP e SVM.

No gráfico da Figura 6 pode-se observar a média das distâncias para cada posição medida no ângulo de 90° e no gráfico da Figura 7 a mesma informação no ângulo

de $22,5^\circ$. A partir desses gráficos pode-se observar que, no ângulo de 90° , as medidas obtidas com os sensores foram, em muitos casos, melhores que as estimadas, embora as distâncias do SVM tenham sido melhores algumas vezes. Isso ocorre porque esse ângulo corresponde aos sensores Sonar 4 e Câmera voltados diretamente para o alvo, realizando sua melhor medida. Porém no ângulo de $22,5^\circ$ pode-se observar, principalmente nas longas distâncias, que as estimativas baseadas nos algoritmos foram melhores que as medidas dos sensores. Nesse caso, o Sonar 4 nem chegou a captar o sinal do objeto, devido à sua inclinação.

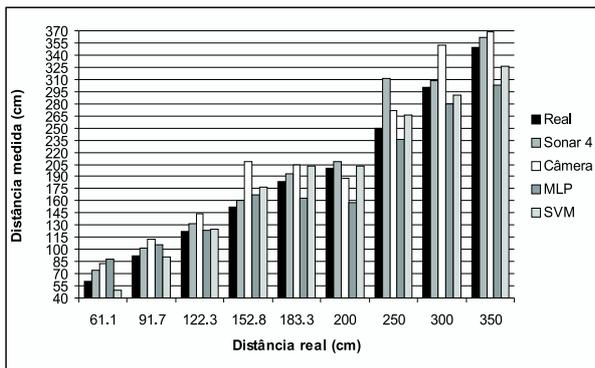


Figura 6: Média das distâncias a 90° .

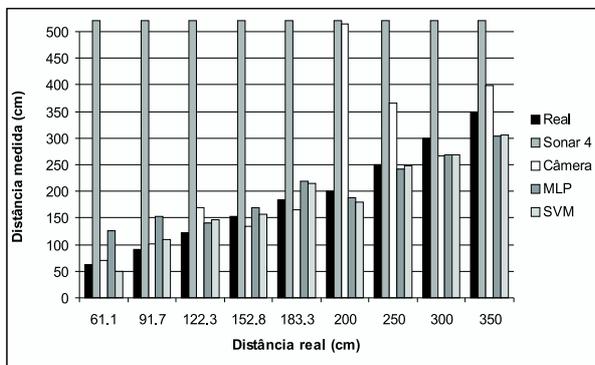


Figura 7: Média das distâncias a $22,5^\circ$.

O gráfico da Figura 8 mostra a média das distâncias obtidas em cada ângulo. A partir desse gráfico observa-se que, na média e em um mesmo ângulo, as distâncias obtidas com os algoritmos foram mais próximas da real do que as obtidas diretamente com os sensores. Mesmo no ângulo de 90° , em que as medidas da câmera e do sonar 4 são as melhores possíveis, as medidas estimadas continuam sendo mais próximas da real, em média.

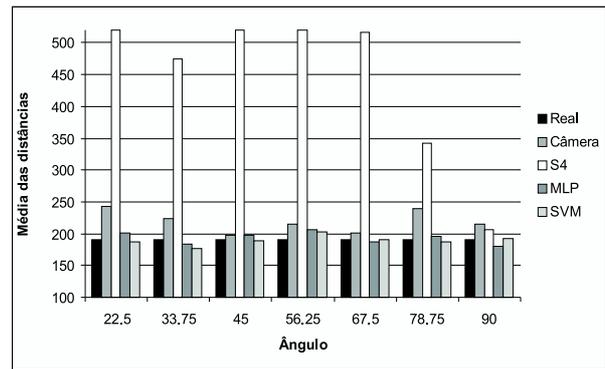


Figura 8: Média das distâncias por ângulo.

A Tabela 1 e o gráfico da Figura 9 mostram a média do erro (barras no gráfico) para todos os ângulos e a 90° , com os dados de teste, bem como os desvios-padrão (linhas sobre as barras no gráfico). Pelo gráfico da Figura 10 pode ser observado o comportamento do erro dos algoritmos no ângulo de 90° . Nesse ângulo, a câmera e o sonar 4 apresentam sua melhor performance. Assim, é interessante observar o comportamento dos algoritmos nesse ângulo.

A partir dessas informações pode-se destacar que os sonares cometem, em média, erros muito maiores do que a câmera e os algoritmos. Pode-se observar ainda que a melhor rede MLP reduziu em 81,6% o erro da câmera e o melhor modelo do SVM reduziu em 87,9% o esse erro. Isso mostra que a utilização dos algoritmos melhorou a precisão das distâncias. Outro aspecto a se considerar é que ambos os algoritmos tiveram um desvio-padrão menor que os sensores, indicando um comportamento mais estável (acurado) e confiável.

Observando o gráfico da Figura 10 e ainda a Tabela 1 pode-se notar que, apesar de, a 90° , a câmera e o sonar 4 apresentarem erros muito menores do que no geral, os algoritmos, nesse ângulo, também apresentam melhores resultados, tanto no que se refere à precisão quanto ao que se refere à acurácia.

Origem	MSE \pm Desvio Padrão	
	Todos ângulos	90°
Câmera	0,0558 \pm 0,2528	0,0061 \pm 0,0078
Sonar 1	0,2996 \pm 0,2244	0,2791 \pm 0,1216
Sonar 2	0,3145 \pm 0,2627	0,5003 \pm 0,2021
Sonar 3	0,3421 \pm 0,2626	0,5003 \pm 0,2021
Sonar 4	0,3250 \pm 0,2506	0,0075 \pm 0,0437
MLP	0,0103 \pm 0,0285	0,0021 \pm 0,0034
SVM	0,0067 \pm 0,0166	0,0017 \pm 0,0034

Tabela 1: MSE e desvio padrão do erro.

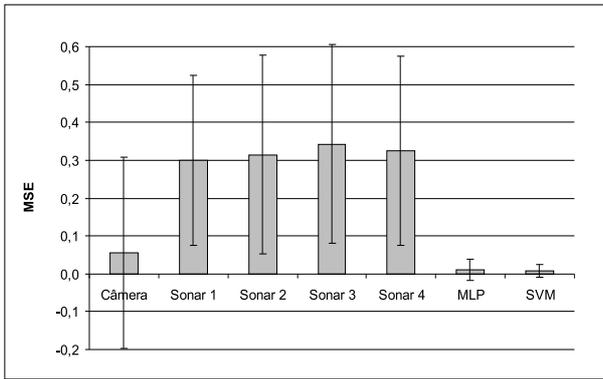


Figura 9: MSE e desvio padrão de cada sensor/algorithm.

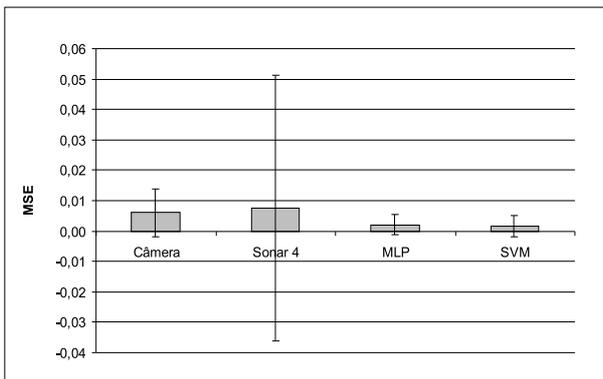


Figura 10: MSE e desvio padrão de cada sensor/algorithm a 90°.

6. Conclusões

Este trabalho investigou a utilização dos algoritmos SVM e redes neurais MLP de AM para fusão de sensores em um robô móvel.

A análise dos resultados mostrou que a utilização desse tipo de algoritmo para a fusão de sensores melhora a qualidade das distâncias medidas, aumentando a precisão e tornando-as mais estáveis (acuradas), principalmente em ângulos em que os sensores não obtêm boas medidas individualmente. O melhor algoritmo (SVM) reduziu em 87,9% o erro médio produzido pelo melhor sensor.

Trabalhos futuros envolvem a investigação de outros algoritmos, bem como a sua combinação em comitês, em que vários modelos diferentes são agrupados de forma a combinar as vantagens de cada um deles e superar ou reduzir as deficiências individuais.

Referências

- [1] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.
- [2] A. P. Braga, A. C. P. L. F. Carvalho, and T. B. Ludermir. *Redes Neurais Artificiais: Teoria e Aplicações*. Livros Técnicos e Científicos (LTC), 2000.
- [3] C. M. Bishop. *Neural Networks for Pattern Recognition*. Claredon Press, New York, NY, 1995.

- [4] M. A. Abidi and R. C. Gonzalez, editors. *Data Fusion in Robotics and Machine Intelligence*. Academic Press, 1992.
- [5] M. P. Groover, M. Weiss, R. N. Nagel, and N. G. Odrey. *Robótica: Tecnologia e Programação*. McGraw-Hill, 1989.
- [6] S. G. Goodridge. *Multimedia Sensor Fusion for Intelligent Camera Control and Human-Computer Interaction*. PhD thesis, Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, 1997.
- [7] H. R. Everett. *Sensors for Mobile Robots: Theory and Application*. A. K. Peters, Wellesley, MA, 1995.
- [8] S. Martens, P. Gaudiano, and A. Carpenter. Mobile robot sensor integration with fuzzy ARTMAP. Technical Report CAS/CNS-TR-98-011, Boston University, 1998. <http://cns-web.bu.edu/~gaudiano/Publications.html>.
- [9] E. Sahin and P. Gaudiano. Visual looming as a range sensor for mobile robots. *Fifth International Conference on Simulation of Behavior*, 1998.
- [10] D. P. Huttenlocher, M. E. Leventon, and W. J. Rucklidge. Visually-guided navigation by comparing edge images. *Algorithmic Foundation of Robotics*, pages 85–96, 1995.
- [11] I. Horswill. Collision avoidance by segmentation. *Proceedings of the 1994 International Conference on Intelligent Robots and Systems (IROS-94)*, Nov. 1994.
- [12] L. A. Klein. *Sensor and Data Fusion Concepts and Applications*. SPIE Press, 1999.
- [13] R. R. Brooks and S. S. Iyengar. *Multi-Sensor Fusion: Fundamentals and Applications with Software*. Prentice Hall, 1998.
- [14] R. Joshi and A. C. Sanderson. *Multisensor Fusion: A Minimal Representation Framework*. World Scientific, 1999.
- [15] M. M. Kokar, J. A. Tomasik, and J. Weyman. A formal approach to information fusion. *Proceedings of the Second International Conference on Information Fusion (Fusion-99)*, 1:133–140, 1999. <http://www.coe.neu.edu/~kokar/publications/f99-mjj.ps>.
- [16] R. R. Murphy. Biological and cognitive foundations of intelligent sensor fusion. *IEEE Transactions on Systems, Man and Cybernetics*, 26(1):42–51, January 1996.
- [17] R. C. Luo and M. G. Kay. *Data Fusion and Sensor Integration: State-of-the-art 1990s*, chapter 2, pages 7–135. In Abidi and Gonzalez [4], 1992.
- [18] K. Faceli, A. C. P. L. F. Carvalho, S. O. Rezende, and F. R. Melfi. Sistemas inteligentes para fusão de sensores. *Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação (II Encontro Nacional de Inteligência Artificial)*, 4:163–175, 1999.
- [19] H. J. C. Pasika. *Neural Network Sensor Fusion Engines for Remote Sensing*. PhD thesis, McMaster University, 1999. <http://soma.crl.mcmaster.ca/~pasika/pubs.html>.
- [20] K. Faceli and A. C. P. L. F. Carvalho. Treinamento de uma rede neural artificial para fusão de sensores para previsão de distâncias em um robô móvel, 1998. Monografia apresentada para o exame final da disciplina SCE-191 - Projeto Supervisionado.