

Uma Ferramenta Livre para Experimentos com Filtros de Partículas

João Bosco O. Monteiro, Hemerson Pistori (Orientador)
Grupo de Pesquisa em Engenharia e Computação, GPEC
Universidade Católica Dom Bosco, UCDB
Av. Tamandaré, 6000 - Jardim Seminário
79117-900 Campo Grande, MS
joabmonteiro@gmail.com, pistori@ec.ucdb.br

Resumo

Este artigo descreve um software livre para manipulação de filtros de partículas, especialmente projetado para facilitar o entendimento e a re-utilização desse tipo de filtro. Também é relatado um estudo de caso de utilização dessa ferramenta no contexto do rastreamento, através de visão computacional, de múltiplos animais, particularmente, camundongos.

1. Introdução

Os métodos de Monte-Carlo sequenciais [8] e os filtros de partículas [5] têm sido bastante utilizados na solução de problemas computacionalmente complexos envolvendo processos estocásticos. Na área de robótica, por exemplo, os filtros de partículas são empregados em problemas de navegação e planejamento de rotas [15]. Já em visão computacional, esses métodos encontram aplicação em problemas de rastreamento de múltiplos objetos [4, 10, 7]. O problema do rastreamento de múltiplos objetos é especialmente complexo, pois consiste na identificação de diferentes entidades, como pessoas ou animais, em seqüências de imagens envolvendo, geralmente, uma grande quantidade de informação.

Apesar da grande produção científica relacionada ao tema, não conhecemos qualquer ferramenta ou biblioteca livre que permita a fácil re-utilização de códigos na implementação de filtros de partículas. Existem diversas implementações com códigos fechados e até uma implementação aberta, o *The Location Stack*¹, mas com códigos-fonte de difícil re-utilização e com inúmeras dependências. Também desconhecemos ferramentas didáticas, com interface gráfica, que facilitem o aprendi-

zado dos algoritmos de rastreamento baseados em filtros de partículas.

Este artigo apresenta uma implementação livre de um filtro de partículas, especialmente projetada para facilitar o entendimento e a re-utilização desse tipo de filtro. Também será relatado um estudo de caso de utilização dessa ferramenta no contexto do rastreamento de múltiplos animais, particularmente, camundongos. O rastreamento de camundongos é uma fase importante na automatização de experimentos em laboratórios que exigem a análise visual de mudanças de comportamento. A construção de um sistema para automatizar esse tipo de experimento é um dos objetivos do projeto TOPOLINO², no qual se insere o trabalho aqui apresentado.

O desenvolvimento do filtro de partículas foi realizado com o apoio de ferramentas livres, como o ImageJ³, para processamento de imagens digitais, e o RandomJ⁴, para geração de números aleatórios. A próxima seção desse artigo apresenta uma breve introdução aos filtros de partículas. A Seção 3 introduz a biblioteca implementada. Em seguida, é apresentado um estudo de caso envolvendo a utilização de filtros de partículas para rastreamento de camundongos. Na Seção 5 encontram-se as considerações finais e as sugestões para trabalhos futuros.

2. Filtros de Partículas

O filtro de partículas é um tipo de filtro preditivo, como o clássico filtro de Kalman [6], mas que diferentemente desse, pode ser aplicado na solução de problemas envolvendo modelos não-lineares e não-gaussianos. Filtros preditivos dividem a solução do problema de rastreamento de um objeto em três etapas: (1) estimar a posição (ou um outro con-

¹ Disponível em <http://portolano.cs.washington.edu/projects/location/>

² Sítio na Internet: <http://www.gpec.ucdb.br/topolino>

³ Disponível em <http://rsb.info.nih.gov/ij/>

⁴ Disponível em <http://www.imagescience.org/meijering/software/randomj/>

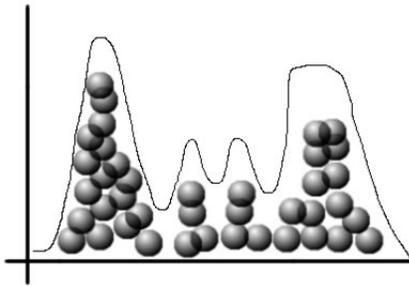


Figura 1. Representação de uma distribuição estatística através de partículas (ou amostras)

junto qualquer de descritores) atual do objeto com base em um modelo de dinâmica e na posição do objeto “no passado”, (2) estimar a posição do objeto a partir de um modelo de observação e de atributos extraídos do último quadro (*frame*) analisado, e (3) combinar os resultados obtidos nas etapas 1 e 2 para obter a estimativa final sobre a posição do objeto. Essas três etapas são também conhecidas como etapas de predição, observação e correção [3]. Diferentes filtros preditivos sugerem diferentes formas para se realizar essas etapas.

A diferença essencial entre um filtro de Kalman e um filtro de partículas está na maneira em que as variáveis ou vetores aleatórios, que representam o estado atual do sistema (e.g. as coordenadas do centro de massa do objeto), são modelados. No filtro de partículas, as variáveis aleatórias podem seguir distribuições não normais e as transformações do modelo de dinâmica não precisam ser lineares. A idéia central, na representação não-paramétrica das distribuições estatísticas, utilizada nos filtros de partículas, é bastante simples, como ilustrado na Figura 1, e consiste em manter uma lista de possíveis valores (partículas) para a variável aleatória, de forma que os valores mais prováveis, segundo a distribuição estatística, apareçam em maior número nessa lista. A Figura 1 apresenta um exemplo de função de distribuição estatística não-gaussiana e sua representação através de partículas (regiões em que o valor da função é mais alto concentram uma maior quantidade de partículas).

Em um filtro de partículas, durante a etapa de predição, o modelo de dinâmica é aplicado a cada uma das partículas, para se calcularem seus novos valores, com base nos valores anteriores. Em seguida, na etapa de observação, com base nas informações obtidas pelo modelo de observação e por atributos extraídos a partir da imagem corrente, cada partícula recebe um peso, que é, grosso modo, proporcional à sua proximidade com os atributos extraídos das imagem.

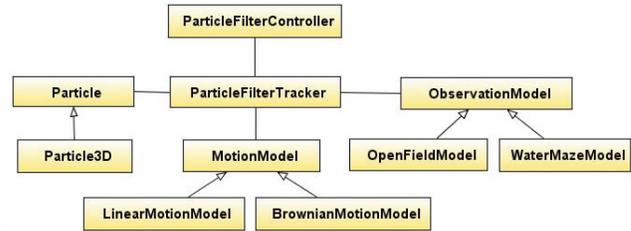


Figura 2. Diagrama de Classes UML para a biblioteca proposta.

Ou seja, partículas “mais similares” à observação recebem pesos maiores. Na correção, também chamada de reamostragem (*resampling*), uma nova lista de partículas é gerada a partir da lista anterior, levando em consideração os pesos, de forma que partículas com maior peso tenham maior probabilidade de serem escolhidas para participar da nova lista [15]. Este processo é repetido iterativamente, para cada novo quadro (*frame*) da imagem.

3. Implementação de Filtros de Partículas

Foi desenvolvida uma biblioteca em Java que permite que um filtro de partículas seja utilizado na solução de problemas de rastreamento de múltiplos objetos, em diferentes contextos. O projeto de implementação visou facilitar a re-utilização de código e a separação dos componentes para permitir a substituição dos modelos de dinâmica e observação, dependendo do problema a ser resolvido. É possível também a utilização de vários filtros de partícula na solução de um mesmo problema, tendo cada filtro seu próprio modelo de dinâmica e observação. Por exemplo, ao rastrear a face e as mãos de uma pessoa, em um problema de tradução de língua de sinais para português [13, 12], poderia-se ter um mesmo modelo de dinâmica para as duas mãos e um diferente para a face.

A Figura 2 apresenta um diagrama UML das classes implementadas. O modelo de observação é especificado na classe *ObservationModel*, que deve ser estendida e especializada de acordo com o problema em questão. Dois modelos de observação, criados especialmente para o rastreamento de camundongos nos experimentos do campo aberto [2] e do *Water-Maze* [11], aparecem no diagrama UML.

A classe *Particle* corresponde a uma amostra (partícula) multidimensional. Cada dimensão corresponde a alguma característica utilizada para representar o objeto, como por exemplo, as coordenadas de seu centro de massa, sua circularidade, inclinação em relação ao eixo horizontal, etc. A dinâmica do sistema é definida pela classe *MotionModel*, que já possui especializações para modelos baseados

em movimento com velocidade uniforme, *LinearMotionModel*, e em movimento Browniano, *BrownianMotionModel*.

No centro da biblioteca está a classe *ParticleFilterTracker*, responsável pela implementação dos módulos de predição, atualização e reamostragem. Esta classe armazena referências para os modelos de observação e de dinâmica, e a coleção de partículas que representa a distribuição do sistema. A classe *ParticleFilterController* permite a sincronização de vários filtros de partículas em um problema de rastreamento de múltiplos objetos. A classe *ParticleFilterController* deve instanciar os rastreadores (*ParticleFilterTracker*) e configurar seus atributos de acordo com o experimento. Além disso, esta classe é responsável por interpretar os resultados obtidos pelo filtro de partículas. Portanto, para cada novo problema é necessário implementar ou estender novos controladores para lidar com situações específicas. Adicionalmente, a classe *ParticleFilterController* provê métodos para representação gráfica dos resultados intermediários durante a execução do filtro, servindo assim como ambiente para aprendizagem. A Figura 3 mostra um tipo de visualização oferecida pela biblioteca, em que informações baseadas em momentos estatísticos de primeira e segunda ordem, extraídos para cada um dos objetos de interesse da imagem (lado esquerdo), são apresentados através de pontos (centros de massa) e elipses coloridas (lado direito).



Figura 3. Visualização através de elipses dos momentos de primeira e segunda ordem extraídos da imagem segmentada dos camundongos.

A Figura 4 ilustra um outro recurso do pacote, que é a visualização do conjunto atual de partículas através de círculos com raios proporcionais aos pesos de cada partícula. Nesse modo de visualização também é apresentado o centro de massa do objeto estimado na etapa de observação. Através deste recurso, é possível inspecionar o estado do filtro de partícula em diferentes iterações.

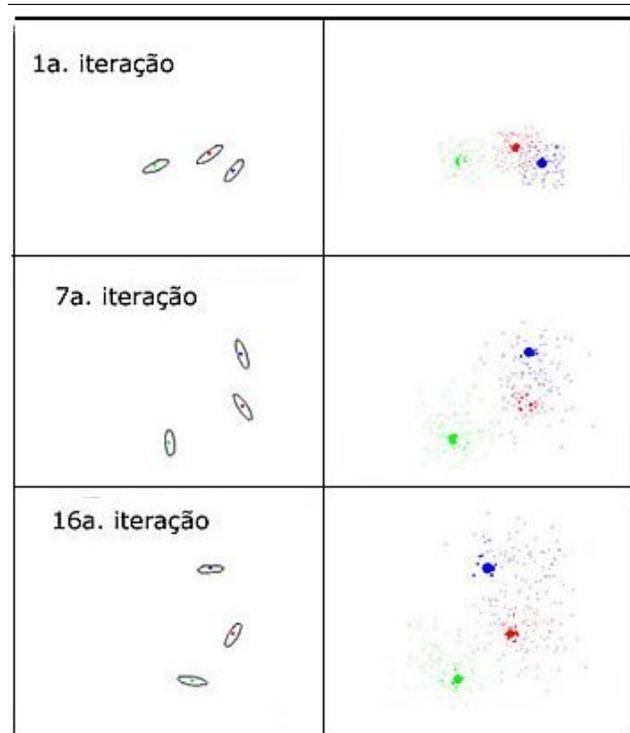


Figura 4. Visualização do conjunto de partículas utilizando a biblioteca implementada.

4. Experimentos e Resultados

O pacote desenvolvido foi utilizado para implementar e testar os filtros de partículas para rastreamento de três camundongos durante um experimento denominado Campo Aberto. Este experimento, inicialmente descrito por Calvin S. Hall em 1930, avalia o comportamento locomotor e o estado emocional do animal[2]. O experimento Campo Aberto é realizado em uma arena cilíndrica, de 40cm de diâmetro com paredes de acrílico translúcido, com 30cm de altura colocada sobre uma base de madeira recoberta com fórmica. A vista superior da arena descrita é apresentada na Figura 5.(a), com a presença de dois camundongos e a vista lateral do experimento é ilustrada na Figura 5.(b).

Para a realização dos experimentos iniciais foram utilizadas três seqüências com 294 (A), 476 (B) e 549 (C) quadros, respectivamente, capturadas de três vídeos distintos, do experimento Campo Aberto. Os vídeos foram gravados com resolução de 320 x 240 pixels, por uma câmera digital Canon Powershot A80. As imagens foram convertidas para tons de cinza e posteriormente foram organizadas em pilhas⁵ representando cada seqüência, através do ImageJ.



Figura 5. (a) Vista superior do experimento Campo Aberto, (b) Vista lateral do mesmo experimento.

Todos os testes foram realizados em um computador pessoal, com processador AMD Athlon 64 3000+ (Clock de 1.8GHz), 1GB de memória RAM e sistema operacional Conectiva Linux 10.

Antes de aplicar o filtro de rastreamento, cada quadro da imagem é pre-processado. Como o experimento foi concebido para facilitar a identificação visual da posição do animal, costuma-se utilizar um camundongo branco sobre um fundo preto, ou vice-versa. Assim, a fase de segmentação torna-se mais simples. No presente experimento, utilizamos uma técnica de limiarização iterativa (*iterative thresholding*) proposta por Ridler e Calvard [16], seguida de subtração de fundo utilizando uma imagem de referência. Para diminuição de ruídos na imagem binarizada, aplicou-se ainda um fechamento morfológico. A Figura 6 ilustra a seqüência de operações realizadas durante a segmentação dos camundongos em um imagem real.

A partir das três seqüências de testes foram produzidas 9 sub-seqüências, contendo trechos das filmagens em que os três camundongos encontram-se mais próximos, dificultando assim o rastreamento. A meta do rastreamento foi o acompanhamento do centro de massa de cada um dos camundongos. Três filtros de partículas foram associados a cada um dos camundongos (que encontram-se sempre separados no primeiro quadro de cada seqüência). Todos os filtros utilizam um mesmo modelo de observação, que mapeia diretamente o centro de massa calculado para cada um dos grupos distintos de pixels encontrados na imagem segmentada. Assim, quando os camundongos encontram-se encostados uns aos outros, o modelo de observação pode mapear um mesmo centro de massa para dois camundongos distintos. A atribuição desses centros de massa, a cada um dos três objetos rastreados, é feita pela distância euclidiana em relação ao centros previstos na iteração anterior do filtro. O modelo de dinâmica considera um movimento Browniano [9, 1], com matriz de covariância identidade.

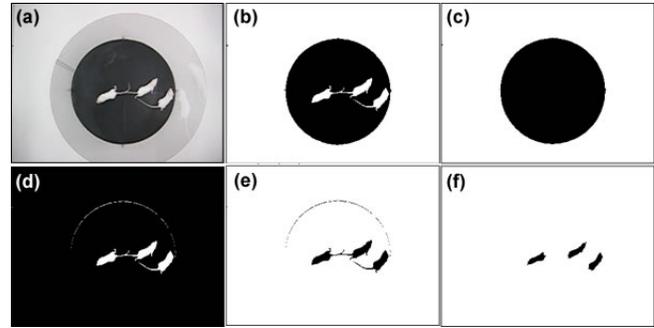


Figura 6. (a) Imagem capturada pela câmera, (b) Imagem Limiarizada, (c) Imagem de Referência, (d) Resultado da subtração da imagens, (e) Inversão dos pixels e (f) Resultado da operação de fechamento.

Através da interface gráfica do pacote, foi possível observar que na maioria dos quadros rastreados, o centro de massa previsto para cada um dos camundongos esteve sempre dentro dos limites definidos pelo corpo do camundongo. A Figura 7 mostra situações nas quais o filtro de partículas obteve sucesso na identificação e rastreamento dos camundongos, mesmo quando eles estiverem muito próximos. Na primeira coluna da figura, à esquerda, tem-se as imagens provenientes da seqüência, já segmentadas. Na segunda coluna está a visualização gráfica dos dados obtidos pelo modelo de observação e na última coluna, o conjunto de partículas de cada filtro após a atualização de pesos.

Para se obter uma primeira estimativa do erro entre o centro de massa calculado pelo filtro, e o centro de massa “real”, três seqüências de imagens foram escolhidas e a partir delas, um observador humano marcou manualmente, um ponto que correspondesse, aproximadamente, ao centro de massa daquele camundongo. A Tabela 1 mostra a média das distâncias euclidianas entre o centro de massa marcado e o obtido pelo filtro, medidas em pixels. O comprimento máximo de um camundongo é de 10 pixels, quando ele está em pé, e 25 pixels quando não está. Por isso, mesmo com o erro observado entre o centro “real” e o estimado, o centro estimado ainda continua na região do camundongo inicialmente rastreado pelo filtro.

Em alguns casos, no entanto, ocorreram inversões entre os camundongos rastreados, principalmente quando os camundongos permanecem juntos por muito tempo e trocam de posições no final da interação, fazendo com que o modelo de observação avalie de maneira errada a correspondência entre os camundongos e seus respectivos centros de massa, como pode ser visualizado na Figura 8. Como houve, num primeiro momento, a atribuição errada

⁵ Recurso existente no ImageJ que permite agrupar várias imagens a fim de facilitar a manipulação das mesmas.

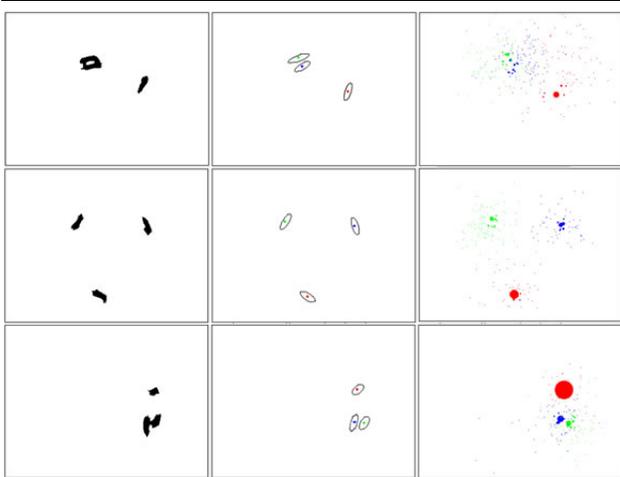


Figura 7. À esquerda, imagens segmentadas. Ao centro, visualização gráfica da etapa observação. À direita, conjunto de partículas de cada filtro após a etapa de atualização de pesos.

	Camund. 1	Camund. 2	Camund. 3
Média	3,61	3,60	2,49
Desvio Padrão	2,89	2,32	1,73

Tabela 1. Médias da distância euclidiana, em pixels, entre a marcação manual do centro de massa dos camundongos e o resultado obtido pelo filtro de partículas, utilizando um conjunto de 100 amostras

de um centro de massa observado a um determinado camundongo, as demais atribuições também se tornam incorretas.

5. Considerações Finais

A principal contribuição deste trabalho foi a criação de uma nova ferramenta computacional, com códigos-fonte abertos e multiplataforma, que pode ser utilizada na implementação de filtros de partículas. A ferramenta tem como um dos principais diferenciais, um módulo gráfico, que permite acompanhar o funcionamento de um filtro preditivo, facilitando assim a aprendizagem de conceitos como previsão, observação e correção; e permitindo ao projetista detectar eventuais problemas com seus modelos de dinâmica e observação. O software apresentado já está sendo utilizado no projeto TO-

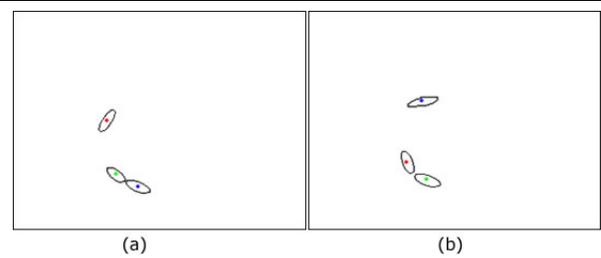


Figura 8. (a) Observação realizada no quadro 10, (b) Observação realizada no quadro 11.

POLINO, que visa a automatização de experimentos com animais de laboratório. A sua utilização no rastreamento de mãos e faces humanas, em um sistema de comunicação homem-máquina baseado em língua de sinais [14], também já foi iniciada.

Pretende-se, futuramente, criar uma biblioteca de modelos de dinâmica e observação que possam ser diretamente utilizados e comparados na solução de outros problemas de rastreamento. Também deverão ser adicionadas algumas possibilidades de reconfiguração do módulo gráfico permitindo uma maior adaptação a diferentes tipos de problemas. Os resultados experimentais com os filtros de partículas precisam de uma investigação mais profunda, com diferentes modelos de observação e dinâmica, mas para o problema em questão, mesmo estes resultados iniciais, indicam a viabilidade de sua utilização.

Agradecimentos

Este trabalho recebeu apoio da Universidade Católica Dom Bosco, UCDB, e da Fundação de Apoio ao Desenvolvimento do Ensino, Ciência e Tecnologia do Estado de Mato Grosso do Sul, FUNDECT.

Referências

- [1] E. Cuevas, D. Zaldivar, and R. Rojas. Particle filter in vision tracking. Technical report, Freie Universität Berlin - Department of Mathematics and Computer Science, 2005.
- [2] D. Eilam. Open-field behavior withstands drastic changes in arena size. *Behavioural Brain Research*, 142:53–62, 2003.
- [3] S. K. Goldenstein. A gentle introduction to predictive filters. *Revista de Informática Teórica e Aplicada (RITA)*, 11:61–89, 2004.
- [4] C. Hue, J. P. L. Cadre, and P. Pérez. Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 38:791–812, 2000.
- [5] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *Int. J. Computer Vision*, 1:5–28, 1998.

- [6] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [7] Z. Khan, T. Balch, , and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision - ECCV'04*, 2004.
- [8] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.
- [9] S. J. Maybank, A. D. Worall, and G. D. Sullivan. A filter for visual tracking based on a stochastic model for driver behaviour. In *ECCV (2)*, pages 540–549, 1996.
- [10] E. F. d. Morais, M. F. M. Campos, F. L. C. Pádua, and R. L. Carceroni. Particle filter-based predictive tracking for robust fish counting. In M. A. F. Rodrigues and A. C. Frery, editors, *Proceedings. IEEE Computer Society*, 9–12 Oct. 2005 2005.
- [11] R. Morris. Developments of a water-maze procedure for studying spatial learning in the rat. *Journal of Neuroscience Methods*, 11(1):47–60, 1984.
- [12] S. C. W. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6):873–891, 2005.
- [13] H. Pistori, P. S. Martins, M. C. Pereira, A. A. C. Jr., and J. J. Neto. Plataforma de apoio ao desenvolvimento de sistemas para inclusão digital de pessoas com necessidades especiais. In *IV Congresso Iberdiscap: Tecnologias de Apoio a Portadores de Deficiência*, Vitória, ES, Fevereiro 2006.
- [14] H. Pistori and J. J. Neto. An experiment on handshape sign recognition using adaptive technology: Preliminary results. *Lecture Notes in Artificial Intelligence. XVII Brazilian Symposium on Artificial Intelligence - SBIA'04*, 3171, September 29 2004.
- [15] I. M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, 3480 University St., Montreal, Québec, CANADA H3A 2A7, 2004.
- [16] T. W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE transactions on Systems, Man and Cybernetics*, August 1978.