

Algoritmo Quantizado de Rastreamento em Redes de Sensores Sem Fio

Éfren L. Souza¹, André N. Campos¹, Eduardo F. Nakamura^{2,1}

¹ Universidade Federal do Amazonas – UFAM
Departamento de Ciência da Computação – DCC
Av. Gen. Rodrigo Octávio Jordão Ramos, 3000, Coroado I – 69077-000 Manaus, AM.

²Fundação Centro de Análise, Pesquisa e Inovação Tecnológica – FUCAPI
Av. Danilo de Matos Areosa, 381, Distrito Industrial – 69075-351 Manaus, AM.

efren@dcc.ufam.edu.br, eduardo.nakamura@fucapi.br, andre.campos@acm.org

Abstract. *Target tracking is an important application of sensor networks, particularly interesting in applications for Ecology, related to monitoring and tracking of animals. In this context, understanding the movement pattern and the territorial occupation of animals are fundamental for understanding their habits. In practice, target tracking is often discretized in occupied areas. In this work, we propose and evaluate a quantized target tracking approach in such a way that the network is organized in a grid, where each cell is a region occupied by the target (animal). The cell size is determined according to the desired granularity. The computation of the target's position obeys a voting scheme, so the technique is simple and low cost. To estimate the target's position, we use the Kalman or Particle filters. Results show that position computation errors are close to two cells, depending on the scenario.*

Resumo. *Rastreamento de alvos é uma importante aplicação de redes de sensores, particularmente interessante em aplicações de Ecologia, relacionadas ao monitoramento e acompanhamento de animais. Nesse contexto, entender a movimentação e a ocupação territorial dos animais é fundamental para compreensão de seus hábitos. Na prática o rastreamento é muitas vezes discretizado em áreas de ocupação. Assim, neste trabalho, propomos e avaliamos uma abordagem quantizada de rastreamento, em que a rede é organizada em grid, onde cada célula do grid é uma região ocupada pelo alvo (animal). O tamanho das células é determinado de acordo com a granularidade necessária. O cálculo de posição é feito por votação, portanto a técnica é simples e de baixo custo. Para estimar posição do alvo usamos os filtros de Kalman ou Partículas. Os resultados mostram erros de cálculo de posição próximos de zero, já os erros das estimativas são de aproximadamente duas células, dependendo do cenário.*

1. Introdução

Uma Rede de Sensores Sem Fio (RSSF) é um tipo especial de rede *ad-hoc* composta por dispositivos com recursos limitados, chamados nós sensores [Akyildiz et al., 2002]. Esses nós sensores são capazes de monitorar um ambiente, coletar dados, realizar processamento localmente e disseminar dados.

Rastreamento de alvos é uma importante aplicação dessas redes, sendo formada por três partes: detecção do alvo (como animais, pessoas e veículos), cálculo de sua posição atual e estimativa de sua posição futura [Nakamura et al., 2007]. Essa aplicação depende das posições dos nós, necessitando que eles conheçam suas posições a priori ou que estas sejam calculadas através de um algoritmo de localização [Souza et al., 2009].

Neste artigo propomos e avaliamos uma abordagem de rastreamento baseada em um esquema de votação para determinar a área em que o alvo se encontra. Para isso, a rede é organizada como um *grid*, em que as células formadas são as possíveis posições do alvo. Nessa abordagem, o cálculo da posição do alvo é feito por votação simples, em que todos os nós têm o mesmo peso, ou por votação ponderada, em que os nós que estão mais próximos do evento têm maior peso. Essas posições calculadas são medidas passadas para os filtros de Kalman (KF – *Kalman Filter*) ou Partículas (PF – *Particle Filter*) [Nakamura et al., 2007]. Dessa forma, esses filtros Bayesianos preveem a próxima posição do alvo.

Esta abordagem foi proposta no contexto dos projetos SAUIM (CNPq 55.4087/2006-5) e RastroAM (CNPq 47.4194/2007-8) para o rastreamento do sauim-de-coleira (primata ameaçado de extinção e presente apenas no Amazonas). Esta é uma aplicação típica de RSSFs para Ecologia, onde os sensores são comumente dispostos em *grids* devido às vias de acesso existentes em pequenas reservas ecológicas. Nesse caso, é importante determinar o padrão de movimentação desses animais, correlacionando, por exemplo, com seus hábitos alimentares e comportamento social. Nessa classe de aplicação prática, a posição exata do alvo é dispensável, sendo importante o conhecimento da região e não a coordenada exata do animal. Dessa forma, considerar uma área como a posição do alvo é relevante para a aplicação, reduz os requisitos de hardware/software e otimiza os custos de energia.

Nas abordagens de rastreamento tradicionais, em que os nós são distribuídos aleatoriamente, um algoritmo de localização deve ser executado antes do rastreamento, para que os nós da rede passem a conhecer suas posições. Esses algoritmos possuem um erro que é somado ao erro do rastreamento [Souza et al., 2009]. Sendo assim, um grande esforço é despendido para calcular a posição exata, mas uma área deve ser considerada devido ao erro acumulado. Outro ponto negativo da distribuição aleatória de nós sensores é que eles não são biodegradáveis, demandando uma coleta posterior de dispositivos inoperantes. Uma estrutura planejada torna viável o uso plataformas atuais de RSSFs.

O restante deste artigo está organizado como segue. A Seção 2 apresenta os fundamentos teóricos e trabalhos relacionados. A Seção 3 apresenta nossa proposta para rastreamento em redes de sensores. A Seção 4 apresenta a metodologia experimental e as avaliações quantitativas. Por fim, a Seção 5 apresenta as conclusões e trabalhos futuros.

2. Fundamentos Teóricos e Trabalhos Relacionados

2.1. Filtros Bayesianos para Rastreamento

O problema de rastreamento de alvos envolve não somente a determinação da rota percorrida pelo alvo, mas também na previsão da próxima posição do alvo (tempo discreto) [Nakamura et al., 2007], aumentando a complexidade do problema.

Neste trabalho são utilizados dois filtros Bayesianos para rastreamento de alvos – Kalman e Partículas – considerados soluções canônicas para o problema [Souza et al.,

2009; Nakamura et al., 2007]. O filtro de Kalman (KF) é o método de fusão de dados mais popular usado para fundir um baixo nível de dados redundantes [Nakamura et al., 2007]. Se um modelo linear pode ser descrito como um sistema e o erro pode ser modelado como um ruído com distribuição de probabilidade Gaussiana, então o KF gera recursivamente estimativas estatisticamente ótimas. O filtro de Partículas (PF) é uma implementação recursiva dos métodos sequenciais de Monte Carlo (SMCs – *Sequential Monte Carlo Methods*) [Arulampalam et al., 2002] e, apesar do KF ser a abordagem clássica, o PF representa uma alternativa para problemas não-lineares com ruídos não-Gaussianos. Soluções adicionais baseadas em KF e PF também são discutidas por Nakamura et al. [2007].

Essas abordagens tradicionais dependem de informação de localização que, naturalmente, estão sujeitas a erros. Poucos trabalhos consideram os erros introduzidos por algoritmos de localização. Oliveira et al. [2009] mostram a influência dos erros de localização no controle de densidade e no roteamento. Já Souza et al. [2009] mostram como o erro de localização afeta o rastreamento e concluem que os filtros Bayesianos não filtram satisfatoriamente esses erros.

Em relação a rastreamento em áreas discretizadas, Ji et al. [2009] e Chen et al. [2010] propõem algoritmos de rastreamento em que a rede é dividida em células, entretanto essa divisão visa apenas ajudar a economizar energia, desabilitando os nós que pertencem a determinadas células. Esses trabalhos consideram que os nós são distribuídos aleatoriamente e que os nós conhecem sua localização após a distribuição, mas não avaliam como os algoritmos reagem quando a localização dos nós não é perfeita. Diferentemente, este trabalho visa o rastreamento sem a necessidade de um algoritmo de localização.

2.2. Modelo de Mobilidade

A mobilidade de um objeto móvel pode ser adquirida registrando a movimentação do objeto de interesse em cenários reais ou então utilizando os modelos de mobilidade. Esses modelos descrevem a posição, velocidade, direção e outros estados dinâmicos de objetos em movimento, descrevendo um padrão da trajetória [Bai & Helmy, 2004].

A Caminhada Aleatória (RW – *Random Walk*) [Bai & Helmy, 2004] é um dos modelos de mobilidade mais populares, entretanto sua natureza sem memória ocasiona manobras bruscas e aceleração repentina, gerando trajetórias muito diferentes dos cenários desejados. Por isso, surgiram modelos de mobilidade com dependência temporal, como a Caminhada Aleatória Correlacionada (CRW – *Correlated Random Walk*) [Wu et al., 2000], em que o estado atual do objeto móvel depende dos estados anteriores.

O CRW é o modelo de mobilidade usado nas avaliações deste trabalho, pois se trata de uma ferramenta da física estatística para obter uma representação aproximada do comportamento animal [Wu et al., 2000]. Esse modelo é um RW que tem a tendência de manter sua direção, ou seja, existe correlação entre os passos sucessivos da trajetória. Para isso, inclui o conceito de memória direcional, que determina o grau de relação em uma caminhada aleatória.

Para configurar esse modelo existem basicamente dois parâmetros: o tamanho do deslocamento e o grau de correlação (ω). O tamanho do deslocamento pode ser a distância entre cada deslocamento (movimento uniforme) ou a distância média. O parâmetro ω

pode variar entre 0 e 1 ($0 \leq \omega \leq 1$). Variando o parâmetro ω , os seguintes cenários de mobilidade podem ser obtidos:

- Aleatório: Nesse caso $\omega = 0$, significa que o modelo está sem memória, ou seja, a velocidade e direção atuais do objeto independem das velocidades e direções anteriores. Esse cenário se torna equivalente ao modelo RW sem tempo de espera.
- Determinístico: Ocorre quando $\omega = 1$, a mobilidade passa a ser totalmente determinística, pois a velocidade e direção atuais são iguais as do tempo anterior.
- Intermediário: Se $0 < \omega < 1$, a velocidade e direção atuais dependem das anteriores, entretanto ainda há um certo grau de aleatoriedade. Quanto mais próximo de um for o valor de ω mais determinístico é o modelo, se tornando mais aleatório quando ω se aproxima de zero.

2.3. Modelo de Detecção de Eventos

Grande parte das aplicações baseadas em eventos em redes de sensores consideram um modelo de detecção, que é usado na validação de algoritmos para diferentes problemas [Nakamura & Souza, 2010], visando aproximar os cenários de avaliação desses algoritmos dos cenários reais.

O raio fixo é um modelo de detecção simples e popular, em que um evento sempre é detectado caso a distância entre tal evento e o sensor seja menor que um raio de detecção [Chakrabarty et al., 2002]. Entretanto, a eficiência da detecção depende de outros fatores, como condições do ambiente e características do evento. Por isso, existem modelos mais elaborados que consideram que cada sensor tem uma probabilidade de detecção que depende da distância entre o sensor e o evento.

Neste trabalho usamos o modelo de detecção de eventos flexível proposto por Nakamura & Souza [2010] que descrevemos a seguir. Esse modelo de detecção de eventos flexível possui um conjunto de parâmetros que podem ser ajustados para representar diferentes situações, servindo como uma ferramenta para avaliar algoritmos e soluções que dependem da detecção de eventos.

Nesse modelo, dados um nó sensor s e um evento e , sendo que e está a uma distância d de s , a probabilidade de s detectar e é modelada como

$$P(d) = \alpha \gamma^{-(\beta d)^\theta}, \quad (1)$$

em que:

- i) $0 < \alpha \leq 1$ é o parâmetro de precisão. Esse parâmetro representa a probabilidade de detecção máxima, sob condições ideais $\alpha = 1$.
- ii) $\gamma > 1$ e $\beta > 0$ são, respectivamente, os parâmetros de espaço vertical e horizontal. A probabilidade de distribuição é baseada no ponto de referência definido por (d_r, P_r) . Isso significa que quando e está a d_r unidades de distância de s , a probabilidade de detecção associada é P_r . Esse ponto é definido fazendo $\beta d_r = 1$, de modo que $P_r = \alpha \gamma^{-1}$. Assim, é possível escolher um ponto de referência (d_r, P_r) configurando os parâmetros de espaço como $\gamma = \alpha P_r^{-1}$ e $\beta = d_r^{-1}$.
- iii) $\theta > 0$ é o parâmetro de inclinação, que define como a probabilidade de detecção diminui de α até 0, de acordo com d .

3. Abordagem Proposta

No modelo de rastreamento proposto, a topologia da rede é disposta em forma de *grid* com I linhas e J colunas. As interseções dessas linhas e colunas representam N nós sensores afastados entre si a uma distância d . Cada nó $n_{i,j}$, com $0 \leq i < I - 1$ e $0 \leq j < J - 1$, conhece a linha e coluna a que pertence, isso possibilita que a fusão de dados para determinar a posição atual seja adquirida somente com operações matemáticas básicas com valores inteiros.

Diferente das abordagens tradicionais, que utilizam coordenadas absolutas para identificar a posição do alvo, nossa abordagem considera a posição do alvo como sendo um célula $c_{x,y}$, com $0 \leq x \leq I - 1$ e $0 \leq y \leq J - 1$. Assim, existem $I \times J$ posições para o alvo. A Figura 1 ilustra a disposição dos nós, as células formadas e o raio de alcance dos nós que estão detectando o alvo.

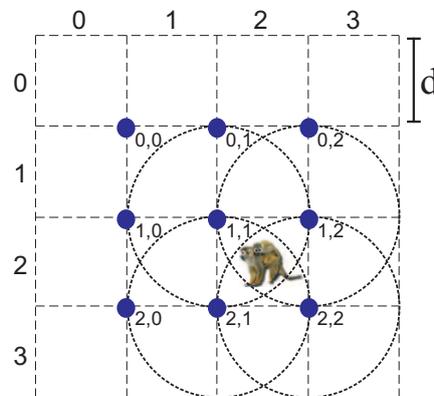


Figura 1. Topologia.

Dessa forma, o alvo divulga sua presença em intervalos de tempo definidos (ex. através de uma coleira com um transmissor). Esse evento é detectado pelos nós do *grid* que encaminham esses dados para o *sink*, onde a fusão de dados é feita para identificar a célula onde o alvo se encontra e a célula em que ele estará na próxima detecção. Os intervalos de tempo para a divulgação devem ser ajustados com base na velocidade do alvo e tamanho das células.

Idealmente, o alcance dos nós deve ser de pelo menos d , para permitir a comunicação entre todos os nós e cobertura para detecção dos eventos. Entretanto, em cenários reais esse alcance deve ser maior para compensar as interferências do ambiente ou alguma imprecisão da distribuição dos nós no *grid*. A distância entre os nós determina a área das células. Então, dependendo do porte do alvo que se pretende rastrear é possível ajustar a distância d e o alcance dos nós para obter células de tamanho adequado.

Para determinar a célula em que o alvo se encontra (tempo t) é realizado um processo de votação. Já a previsão da próxima posição do alvo (tempo $t + 1$) é realizada com estimativas de filtros *Bayesianos* alimentados com histórico de mobilidade do alvo.

3.1. Calculando a Posição do Alvo

Calcular a posição do alvo consiste em definir em que célula ele se encontra no tempo atual t . Para isso, são utilizadas apenas as coordenadas (i, j) contidas nos nós, fazendo

uma votação entre os nós que detectam o alvo no tempo t . Esse processo de votação pode ser simples ou ponderado, como mostramos a seguir.

3.1.1. Votação Simples

Na votação simples, cada nó atribui um voto para suas células candidatas. As células candidatas de um nó $n_{i,j}$ são as células ao seu redor e, a posição do alvo é definida como a célula mais votada.

Como prova-de-conceito, os processos de estimação de posição e rastreamento são centralizados no nó *sink*. Os nós sensores que detectam o alvo no tempo t encaminham para o *sink* uma mensagem sinalizando que detectaram o alvo. Essa mensagem contém a posição (i, j) do nó que a originou. Baseado nessa informação, o *sink* determina as células candidatas e elege a mais votada. A Figura 2 exemplifica o processo de votação simples quando quatro nós detectam o evento. Nesse caso, existem nove células candidatas, a célula $c_{1,1}$ recebe voto de todos os nós, sendo a mais votada e, portanto, escolhida como a posição do alvo.

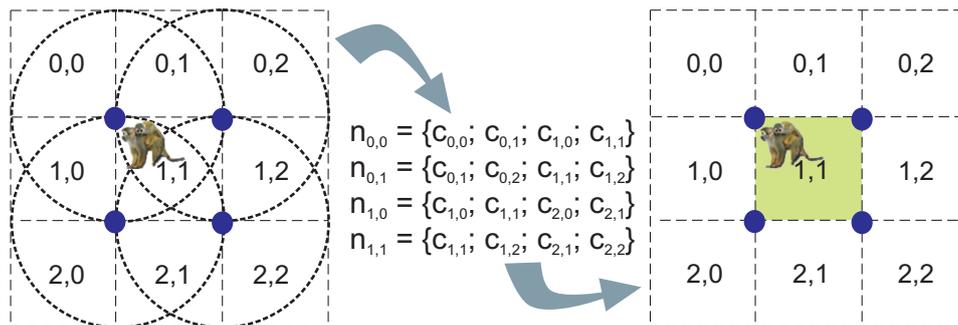


Figura 2. Detecção simples.

Empates podem ocorrer no processo de votação de um evento. Um possível caso de empate ocorre quando poucos nós (um ou dois) detectam o evento. Empates também ocorrem com frequência quando o alcance o alvo, ao divulgar sua posição, é maior do que distância entre os mesmos. Nos casos de empate, a célula eleita é a mais central dentre as candidatas, sendo calculada da seguinte forma:

$$c(x, y) = c \left(\frac{\sum_{k=1}^m x_k}{m}, \frac{\sum_{k=1}^m y_k}{m} \right), \quad (2)$$

onde m é a quantidade de células candidatas e (x_k, y_k) é a coordenada da k -ésima célula candidata.

3.1.2. Votação Ponderada

A detecção ponderada utiliza estimativas de distância (obtida através da potência do sinal) para classificar os nós sensores com maior poder de decisão. Os nós com menor distância estimada, provavelmente os mais próximos do evento, têm maior peso, de forma que o valor do voto de um nó $n_{i,j}$ é dado por $v_{i,j} = e^{-d_{i,j}}$, onde $d_{i,j}$ é a distância estimada entre

o nó $n_{i,j}$ e o alvo. Dessa forma, os nós mais próximos têm voto de maior peso, mesmo que a diferença de distância seja pequena.

Em aspectos gerais a detecção ponderada é bastante similar a detecção simples. Entretanto, calcular a posição do alvo classificando os nós por peso é mais preciso quando se utiliza raios de detecção maiores dos que a distância entre os nós. Quanto maior é o alcance do alvo maior é a quantidade de referências que confirmam que a posição está correta. Por outro lado, esse método está sujeito a imprecisão das estimativas de distância que pode resultar em uma distribuição imprecisa de pesos.

3.2. Estimando a Posição do Alvo

Estimar a posição do alvo consiste em determinar a posição (célula) em que o alvo estará no tempo $t + 1$, baseando-se apenas na posição do mesmo no tempo t . A Figura 3 mostra um exemplo em que o algoritmo de rastreamento no tempo t estima a célula $c_{1,5}$ como a posição do alvo no tempo $t + 1$.

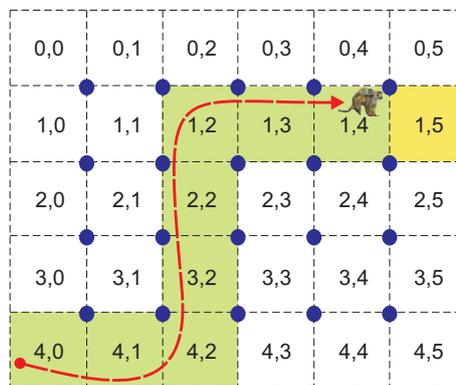


Figura 3. Estimativa de posição do alvo.

Neste trabalho, utilizamos os filtros de Kalman e Partículas como estimadores. O valor obtido desses métodos na fase de previsão no tempo t é definido como a provável posição do alvo no tempo $t + 1$, sendo que esses são alimentados na fase de correção pelas posições calculadas com a votação simples ou ponderada. Esses filtros *Bayesianos* são métodos normalmente usados para trabalhar com dados contínuos, mas no caso deste trabalho, em que as coordenadas são representadas por valores inteiros e discretos, os filtros executam apenas operações com números inteiros (menor custo).

4. Avaliação

Nesta seção descrevemos a metodologia de avaliação e os resultados obtidos.

4.1. Metodologia

As avaliações são realizadas por meio de simulações com Sinalgo [Group, 2008]. Como prova-de-conceito a estimação de posição e rastreamento são executados pelo *sink*. A configuração padrão da rede é composta de 2401 nós sensores distribuídos em *grid* em um campo de sensores de $500 \times 500 \text{m}^2$, além do *sink* e do alvo que são posicionados aleatoriamente. Dessa forma, os nós ficam a 10m de distância uns dos outros, resultando em 50×50 células de 10m^2 cada.

Os alcances dos nós do *grid* e do *sink* são de 10m. Em um cenário ideal isso possibilita que todos os nós se comuniquem através de múltiplos saltos. O roteamento dos dados é feito com uma árvore de roteamento em que o *sink* é a raiz. Já o alcance do alvo é de 30m, permitindo que mais de três nós detectem o evento.

O modelo de mobilidade seguido pelo nó móvel é a Caminhada Aleatória Correlacionada (ver Seção 2.2) com grau de correlação de 0.99 e velocidade de 1m/s. Agendamos 300 eventos (alvo divulgando sua presença) em intervalos 10s. Nessa configuração, a trajetória do alvo cobre todo o campo de sensores, garantindo que o alvo divulgue sua posição quando mudar de célula.

O *sink* calcula a posição do alvo por votação simples (ver Seção 3.1.1) ou ponderada (ver Seção 3.1.2). Já a previsão é feita usando os Filtros de Kalman ou Partículas. O filtro de Partículas utiliza 1000 partículas, pois valores obtidos em avaliações preliminares mostraram que mais partículas não resultam em maior precisão.

Modelamos o consumo de energia de acordo com Yupho & Kabara [2007]. A Tabela 1 mostra os parâmetros usados nas simulações. Consideramos que cada mensagem tem um tamanho 64bytes.

Parâmetro	Valor
Energia inicial	50 joules
Transmissão	342×10^{-7} joules/byte
Escutando	1888×10^{-8} joules/s

Tabela 1. Parâmetros de energia.

Considerando que c_{x_1, y_1} é a célula em que o alvo está e que c_{x_2, y_2} é a célula estimada, o erro é dado por

$$\epsilon(c_{x_1, y_1}, c_{x_2, y_2}) = \max(|x_1 - x_2|, |y_1 - y_2|). \quad (3)$$

Os resultados dos experimentos são obtidos da média de 50 execuções diferentes. As barras de erro representam um intervalo de confiança de 99%.

4.2. Resultados das Simulações

4.2.1. Alcance do Alvo

O alcance do alvo determina a quantidade de nós que podem detectar o evento. O ideal seria que um nó $n_{i,j}$ detectasse o evento sempre que o alvo estivesse na célula $c_{i,j}$, $c_{i+1,j}$, $c_{i,j+1}$ ou $c_{i+1,j+1}$. Entretanto, considerando que o evento é detectado com o modelo de raio fixo, o alvo deve ter um alcance próximo da distância entre os nós do *grid* para que, na maioria dos casos, o evento seja detectado por pelo menos três nós. Se menos de três nós detectam o alvo é impossível determinar com exatidão a célula em que o alvo está, mas é possível obter uma célula próxima.

Desse modo, para avaliarmos como a quantidade de nós que participam da votação influenciam na precisão do rastreamento, variamos o alcance do alvo de 10m a 50m. O

modelo de detecção de eventos usado é o raio fixo, então é possível executar o rastreamento satisfatoriamente com essa variação. Os resultados são mostrados na Figura 4.

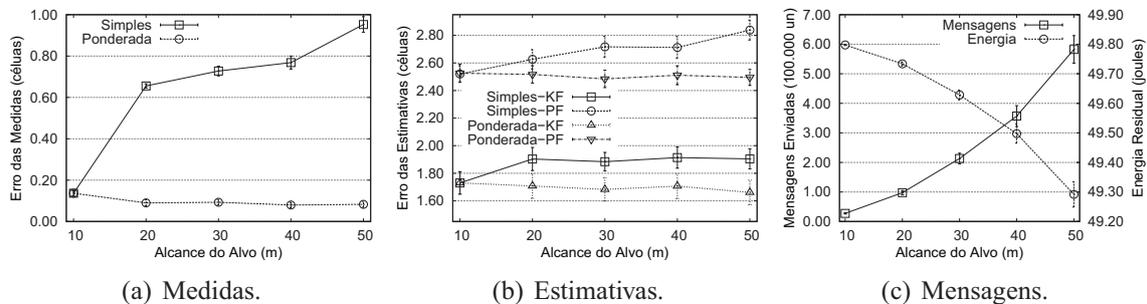


Figura 4. Influência do alcance do alvo.

Notamos na Figura 4(a) que no cálculo de posição com alcance de 10m, as votações simples e ponderada obtêm o mesmo resultado, pois o peso do voto não interfere no resultado com a baixa quantidade de nós que participam da votação. Quando o alcance do alvo aumenta há uma melhora sutil na votação ponderada, que só erra quando o alvo está nas bordas do campo de sensores. Já a votação simples é prejudicada com esse aumento, pois ocorrem muitos empates.

A Figura 4(b) mostra as estimativas combinando as técnicas de votação simples e ponderada com os filtros de Kalman e Partículas. Verificamos que o filtro tem mais influência nas estimativas do que o tipo de votação, embora as estimativas reflitam os erros das medidas. Por ser a solução ótima, o filtro de Kalman alcança resultados mais precisos que o filtro de Partículas.

Cada nó que detecta o evento gera uma mensagem que é encaminhada para o *sink*, onde é realizada a votação e a estimativa. Por isso, quando o alcance do alvo aumenta mais mensagens são geradas na rede, incrementando o consumo de energia dos nós, como mostra a Figura 4(c).

4.2.2. Detecção de Eventos

A eficiência de detecção depende de diversos fatores, como condições do ambiente e características do evento. Na avaliação anterior consideramos que o evento sempre é detectado pelos nós dentro do raio de alcance do alvo. Essa suposição é interessante para verificar o cenário ideal, mas é pouco realista.

Por isso, nesta seção usamos o modelo de detecção de eventos flexível (ver Seção 2.3) para verificar como nossa abordagem reage a falhas de detecção de eventos. Configuramos $\gamma = 2$, $\theta = 1000$ e variamos o parâmetro de precisão α de 0,5 a 1,0. Para que o alvo tenha alcance de 10m, 20m ou 30m, usamos o valor de β como $\frac{1}{10}$, $\frac{1}{20}$ e $\frac{1}{30}$, respectivamente. Os resultados são mostrados na Figura 5.

A Figura 5(a) mostra que quanto menor o raio de alcance do alvo, menor é o erro

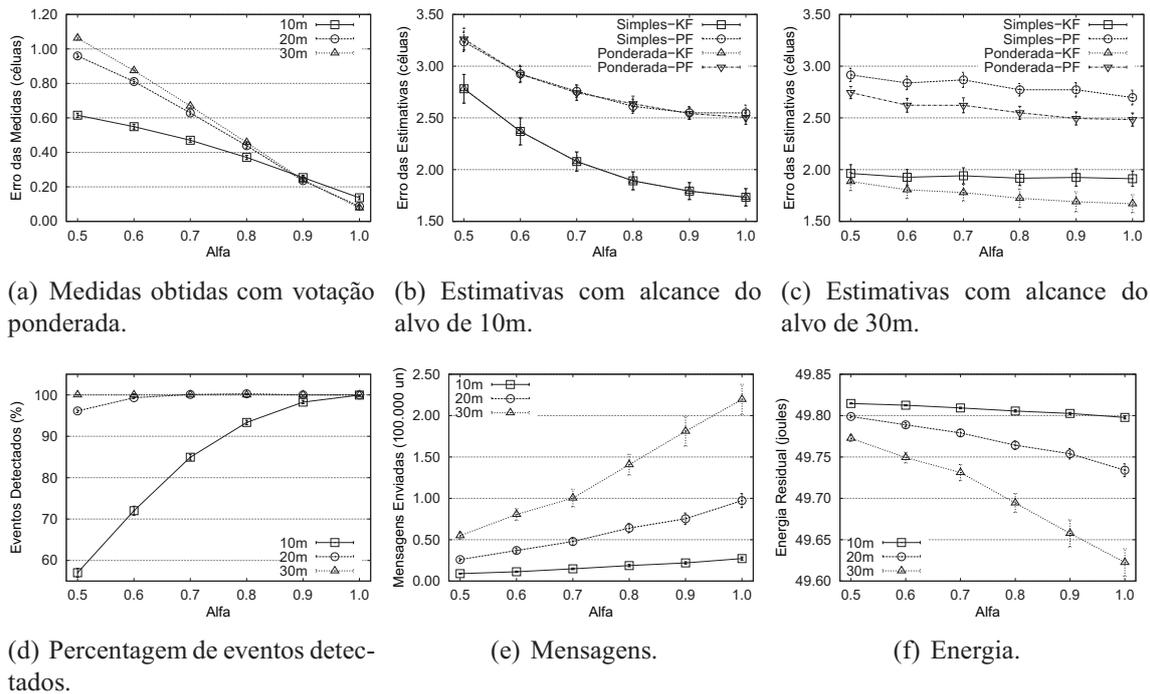


Figura 5. Influência das falhas de detecção de eventos.

das medidas¹, principalmente quando o valor de α é baixo (sensor impreciso). Entretanto, é errado concluirmos que configurar o alvo com um alcance pequeno é vantajoso, pois o erro das medidas se torna menor porque muitos eventos não são detectados, já que os nós falham em detectá-lo, como mostra a Figura 5(d). Nessa figura 100% dos eventos são garantidamente detectados, independente do valor de α quando o alcance do alvo é de 30m, para valores menores, eventos são perdidos.

As Figuras 5(b) e 5(c) mostram que o rastreamento é mais preciso a medida que a detecção de eventos é melhor, independente do alcance do alvo. Entretanto, com alcance de 30m, o rastreamento é menos prejudicado, pois a divulgação do evento alcança mais nós, logo há uma probabilidade maior do evento ser capturado. Isso não ocorre quando o alcance é de 10m, pois os eventos são perdidos com mais facilidade. Ainda podemos notar nessas figuras que, embora a votação ponderada e a votação simples tenham o mesmo resultado quando o alcance do alvo é de 10m, a votação ponderada tem vantagem sobre a simples nos demais casos. Além disso, o KF tem vantagem sobre o PF, logo a combinação que tem melhor resultado é a votação ponderada com KF.

Tanto o alcance do alvo quanto a precisão da detecção de eventos têm influência na quantidade de mensagens enviadas pela rede e conseqüentemente no consumo de energia da mesma (Figuras 5(e) e 5(f)). Quanto maior for o raio de alcance do alvo, maior é a quantidade de nós que detectam o evento, logo mais mensagens precisam ser enviadas, aumentando o consumo de energia. Além disso, a quantidade de mensagens e o consumo de energia diminuem quando a imprecisão da detecção de eventos aumenta, pois os nós que falham em detectar o evento não enviam mensagens.

¹Mostramos apenas os resultados da votação ponderada para simplificar o gráfico, pois ela apresentou melhores resultados que a votação simples.

4.2.3. Interferências

Nesta seção avaliamos como nosso modelo reage às interferências provocadas pelo ambiente. Para isso, usamos a relação sinal/ruído mais interferência (SINR – *Signal to Interference plus Noise Ratio*) [Gupta & Kumar, 2000], que provavelmente é o modelo de interferência mais conhecido. Esse modelo assume que a intensidade de um sinal diminui exponencialmente com a distância. Três parâmetros precisam ser configurados nesse modelo: τ , parâmetro de perda pela distância; η , limiar usado para definir se o pacote chegou ou não; e ν , ruído do ambiente. Neste experimento fixamos $\tau = 2$, $\nu = 0$ e variamos η de 0 a 0,5. Os resultados são mostrados na Figura 6.

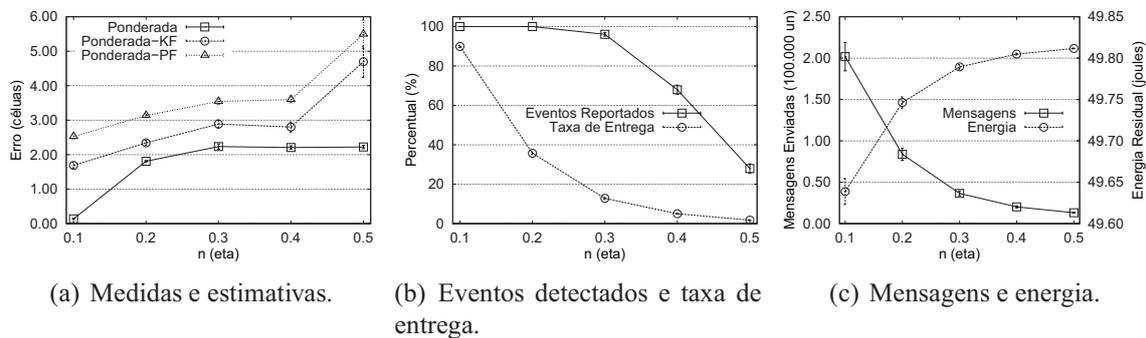


Figura 6. Influência da interferência do ambiente.

A Figura 6(a) mostra que o incremento de η afeta o desempenho da votação, mas o erro estabiliza quando $\eta \geq 0,3$. Isso ocorre porque, nesses casos, a quantidade de dados que chegam ao *sink* são insuficientes para gerar uma votação consistente, pois a taxa de entrega de dados é inferior a 20% (Figura 6(b)).

O erro das medidas, gerado pela baixa taxa de entrega de dados, também interfere nas estimativas do rastreamento, como mostra a Figura 6(a). Entretanto, o aumento do erro das estimativas fica mais evidente na diferença entre 0,4 e 0,5 de η . Isso ocorre porque com $\eta = 0,5$ a quantidade de eventos detectados é inferior a 30% (Figura 6(b)), tornando o trabalho dos filtros Bayesianos mais difícil, pois se formam lacunas entre os valores passados como medidas. O excesso dessas lacunas faz com que os filtros fiquem constantemente instáveis.

Por fim, a quantidade de mensagens e, conseqüentemente, a energia de toda a rede também são influenciadas pelas interferências do ambiente. Quanto maior for a interferência do ambiente menor é a quantidade de mensagens que são enviadas, como mostra a Figura 6(c), pois alguns dos dados que deveriam chegar até o *sink* se perdem ao longo do processo de roteamento. Por esse mesmo motivo, a rede acaba economizando energia, como mostra a mesma figura.

4.2.4. Imprecisão das Estimativas de Distância

A distância estimada pelos nós sensores não é perfeita. Os erros dessas estimativas podem ser altos dependendo do ambiente monitorado, afetando o desempenho do rastreamento. Em geral, esses erros dependem da distância e podem ser modelados como uma variável

Gaussiana de média zero, em que o desvio padrão é um percentual da distância atual [Oliveira et al., 2009].

Na nossa abordagem, a votação ponderada depende das estimativas de distância. Desse modo, para avaliarmos diferentes situações, variamos o desvio padrão de 0% a 50% da distância. O desvio padrão de 0% resulta em estimativas de distância perfeitas, 10% da distância corresponde às estimativas obtidas por técnicas baseadas no tempo como Tempo de Chegada do Sinal (TOA – *Time Of Arrival*), que tem erros menores que 1m. As demais situações representam as estimativas obtidas por métodos mais imprecisos como Intensidade do Sinal Recebido (RSSI – *Received Signal Strength Indicator*). Os resultados são mostrados na Figura 7.

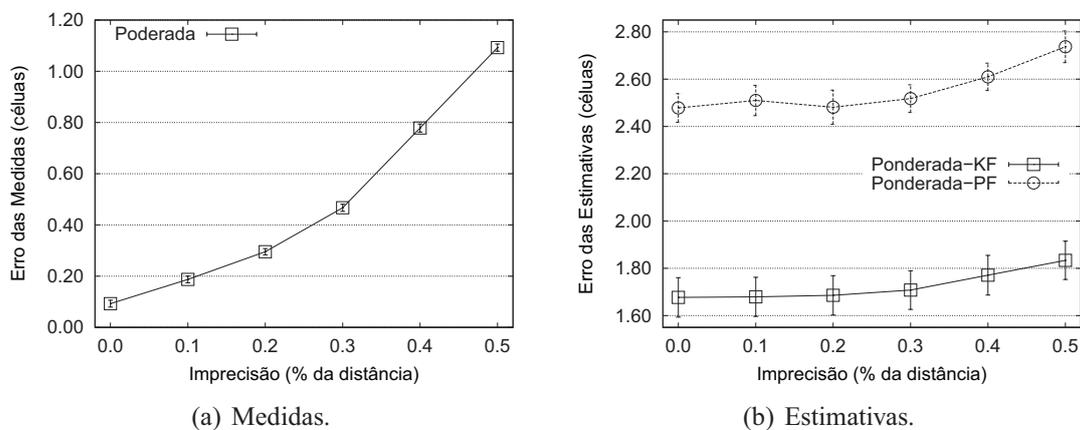


Figura 7. Influência da imprecisão das estimativas de distância.

Notamos na Figura 7(a) que a votação ponderada erra mesmo quando as estimativas de distância são perfeitas. Essas falhas ocorrem somente se o alvo estiver nos limites do campo de sensores, pois nessa situação o alvo não fica cercado pelos nós do *grid*. O aumento da imprecisão das estimativas de distância afeta diretamente o desempenho da votação ponderada, já que os pesos de votação acabam sendo distribuídos de forma incorreta.

As estimativas também são afetadas com esse aumento no erro das medidas, mas em menor escala (Figura 7(b)). A diferença do erro das medidas entre as imprecisões de 0% e 50% é de aproximadamente 1 célula, mas essa mesma diferença nas estimativas é de aproximadamente 0,15 células. Isso acontece porque os filtros reduzem a influência dos erros das medidas.

4.2.5. Correlação da Mobilidade do Alvo

A maneira como o alvo se move é uma característica importante que deve ser considerada pelas aplicações de rastreamento, pois a correlação da mobilidade influencia no desempenho dos estimadores. Nesta avaliação variamos o parâmetro de correlação (ω) de 0,95 a 1,00. Os resultados são mostrados na Figura 8.

A Figura 8(a) mostra que a votação não é influenciada pela correlação da mobilidade, pois para esse processo não importa como o alvo se move, mas sim onde ele está

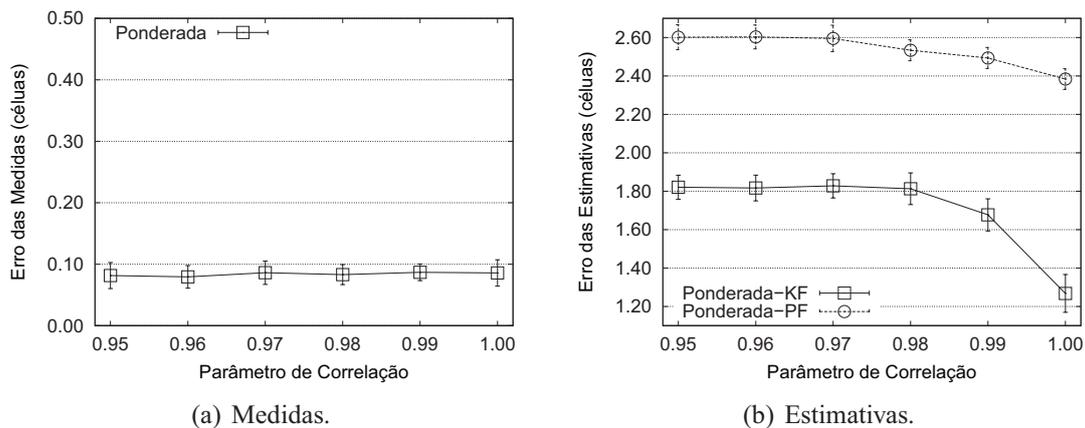


Figura 8. Influência da correlação de mobilidade do alvo.

no momento em que divulga sua posição. Já a Figura 8(b) mostra que quanto mais correlacionada for a mobilidade menor será o erro das estimativas. Isso ocorre porque uma correlação baixa faz o alvo realizar mais manobras, levando o filtro a errar até estabilizar novamente. Quando a mobilidade é 100% correlacionada, os estimadores erram apenas quando o alvo chega nos limites do campo de sensores, sendo obrigado a manobrar para ficar dentro desse campo.

5. Conclusões e Trabalhos Futuros

Este trabalho propõe e avalia uma técnica de rastreamento em redes de sensores com uma estrutura em *grid*. Para calcular a posição do alvo são usadas duas técnicas de votação: a simples, em que todos os nós têm o mesmo peso; e a ponderada, em que os nós mais próximos do alvo têm maior peso. O rastreamento é realizado com o filtro de Kalman ou Partículas, alimentados com as coordenadas das células obtidas pela votação.

Como conclusão geral, a votação simples parece ser uma boa opção quando o raio de alcance do alvo é igual a distância entre os nós do *grid*, entretanto quando consideramos cenários em que ocorrem falhas de comunicação, a votação ponderada obtém melhores resultados, mesmo nas situações em que a imprecisão das estimativas de distância são altas. Além disso, o filtro de Kalman mostra-se mais eficiente que o filtro de Partículas independente do cenário adotado. São necessários aproximadamente quatro nós envolvidos na votação para determinar com exatidão a célula em que o alvo está, portanto é importante que o alcance do alvo seja configurado para cobrir mais nós, afim de evitar que alguns eventos sejam perdidos.

Indicações para o futuro incluem:

- Implementar esta técnica de rastreamento de forma distribuída, por enquanto toda a fusão de dados é feita no nó *sink*, apenas para verificar a precisão da técnica;
- Utilizar fusão de dados para reduzir a quantidade de pacotes enviados pelos nós, agrupando dados em um mesmo pacote;
- Avaliar uma abordagem de roteamento mais eficiente para esta abordagem;
- Reduzir os erros que ocorrem na votação quando alvo está nos limites do campo de sensores.

Referências

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422.
- Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on Particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.
- Bai, F. and Helmy, A. (2004). A survey of mobility models in wireless adhoc networks. In *Wireless Adhoc and Sensor Networks*, chapter 1, pages 1–30. Kluwer Academic Publishers.
- Chakrabarty, K., Iyengar, S., Qi, H., and Cho, E. (2002). Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers*, 51(12):443–454.
- Chen, J.-F., Wang, Y.-H., Huang, K.-F., and Chang, T.-W. (2010). Grid-based mobile target tracking mechanism in wireless sensor networks. *Journal of Communications*, 5(6):475–482.
- Group, E. D. C. (2008). Sinalgo - simulator for network algorithms. <http://dcg.ethz.ch/projects/sinalgo/>.
- Gupta, P. and Kumar, P. (2000). The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404.
- Ji, X., Zhang, Y.-Y., Hussain, S., Jin, D.-X., Lee, E.-M., and Park, M.-S. (2009). FOTP: Face-based Object Tracking Protocol in wireless sensor network. In *Proc. of the Conference on Computer Sciences and Convergence Information Technology*, pages 128–133, Los Alamitos, USA.
- Nakamura, E. F., Loureiro, A. A. F., and Frery, A. C. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys*, 39(3):1–55 (A9).
- Nakamura, E. F. and Souza, E. L. (2010). Towards a flexible event-detection model for wireless sensor networks. In *Proc. of the Symposium on Computers and Communications*, pages 459–462, Riccione, Italy.
- Oliveira, H. A. B. F., Boukerche, A., Nakamura, E. F., and Loureiro, A. A. F. (2009). An efficient directed localization recursion protocol for wireless sensor networks. *IEEE Transactions Computing*, 58(5):677–691.
- Souza, E. L., Nakamura, E. F., and de Oliveira, H. A. (2009). On the performance of target tracking algorithms using actual localization systems for wireless sensor networks. In *Proc. of the Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 418–423.
- Wu, H., Li, B.-L., Springer, T. A., and Neill, W. H. (2000). Modelling animal movement as a persistent random walk in two dimensions: expected magnitude of net displacement. *Ecological Modelling*, 132(2):115–124.
- Yupho, D. and Kabara, J. (2007). The effect of physical topology on wireless sensor network lifetime. *Journal of Networks*, 2(5):14–23.