

# Descoberta de Posições Táticas em Ambientes Virtuais com a Utilização de Agrupamentos Espaço-Temporais

Ivan Medeiros Monteiro

Luis Otavio Alvares

Universidade Federal do Rio Grande do Sul

Vania Bogorny

Universidade Federal de Santa Catarina

## Resumo

A demanda por comportamentos convincentes em jogos de computador tem orientado a pesquisa em inteligência artificial com o objetivo em aprimorar o comportamento dos personagens em jogos. A fim de alcançar um comportamento convincente, os personagens devem usar o conhecimento sobre no mundo o qual ele atua. A representação do ambiente é muito importante para executar um comportamento mais sofisticado, mas na maioria dos jogos essa representação ainda continua sendo definida manualmente de forma estática pelo projetista do ambiente, levando os personagens a um comportamento previsível e pouco convincente. Apesar da geração automática de posições táticas vim sendo tratada por alguns trabalhos, não existe uma solução geral e as abordagens atuais possuem muitas limitações. Devido a isto, este trabalho apresenta alguns resultados preliminares do uso de agrupamento espaço-temporal para encontrar posições táticas em ambientes virtuais, permitindo que os bots se adaptem ao modo como as partidas são jogadas.

**Keywords::** Agrupamentos Espaço-Temporais, Mineração de Dados, Sistemas Multiagentes, Inteligência Artificial.

## Author's Contact:

{imonteiro,alvares}@inf.ufrgs.br  
vbogorny@inf.ufsc.br

## 1 Introdução

Na maioria dos jogos, informações táticas <sup>1</sup> são especificadas no ambiente pelo projetista do mundo virtual. Entretanto, a especificação dessas regiões interessantes durante a criação dos mapas é um trabalho que consome muito tempo. Em geral, essa tarefa é realizada manualmente, o que pode limitar as interações no ambiente, dado que algumas localizações interessantes podem não ser identificadas pelo projetista.

Algumas sequências de ações simples, como ações lineares predefinidas, podem ser orquestradas com sucesso pelo projetista do ambiente através da especificação de posições táticas. Entretanto, seu trabalho leva em consideração apenas o ambiente estático, e não avalia as interações entre personagens no ambiente. Além disso, a especificação manual de posições táticas pode ser um trabalho pesado, sendo suscetível a falhas, dado que o projetista pode não prever todos os pontos interessantes que podem surgir no ambiente durante as partidas.

A situação é um pouco mais crítica quando os mapas são gerados automaticamente, porque o projetista não pode especificar diretamente essas regiões. Uma solução para esse tipo de problema é a descoberta automática dessas regiões interessantes a partir das experiências dos bots no ambiente. Isto permitiria aos bots ter não apenas um melhor conhecimento sobre o ambiente no qual estão inseridos, como também habilitaria o agente a aprender sobre os comportamentos e as estratégias dos oponentes.

Existem alguns trabalhos que abordam o problema das posições táticas a partir de diferentes aspectos. Alguns deles descrevem uma representação espacial das localizações interessantes [Tozour 2001] [Isla 2006], enquanto outros provêm uma forma de gerar automaticamente essas posições táticas [Tozour 2004] [Straatman et al. 2006] [Darken and Paull 2006] [Dill and Sramek 2004]. En-

tretanto, nenhum deles realiza a descoberta de localizações táticas usando o conhecimento obtido nas partidas.

Assim, esse trabalho apresenta um caminho inicial para o uso dos métodos de descoberta de conhecimento baseado em agrupamento espaço-temporal, usando informações sobre a posição dos personagens para aprender qual tipo de evento ocorre em quais partes do ambiente. Nós também examinamos como os personagens se comportam em uma partida a fim de descobrir quais estratégias eles estão utilizando, através da análise de suas trajetórias no ambiente. Assim, conseguimos inicialmente descobrir locais perigosos e posições de camper usando os dados gravados das partidas. O algoritmo de agrupamento por densidade, DBScan [Ester et al. 1996], é utilizado para encontrar as regiões perigosas enquanto o algoritmo CB-SMoT [Palma et al. 2008] é usado sobre as trajetórias dos personagens para descobrir as posições de camper.

Os experimentos foram executados com o jogo Unreal Tournament 2004, gravando as partidas através da interface GameBots [Kaminka et al. 2002]. O ciclo de aprendizado é realizado por partidas, onde ao final de cada uma, os logs são analisados a fim de atualizar o modelo do ambiente e para identificar como os personagens têm se comportado. Assim, é possível desenvolver bots adaptativos, que podem atuar de acordo com o modelo de mundo aprendido, contornando comportamentos repetitivos e tornando-se mais convincentes. Além disso, surgem novas oportunidades para o planejamento de caminho, levando em consideração os pesos que podem ser alocados para cada região descoberta.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados, a Seção 3 descreve brevemente as técnicas de mineração utilizadas, a Seção 4 mostra como realizar a descoberta de posições táticas utilizando agrupamento espaço-temporal, na Seção 5 são apresentados os experimentos realizados, e na Seção 6 se conclui este trabalho.

## 2 Trabalhos Relacionados

A representação de locais de interesse em jogos é importante para permitir que sejam desenvolvidos comportamentos convincentes para os personagens. Essa representação dá liberdade ao bot para que ele raciocine sobre uma perspectiva tática do estado atual do jogo. Diferentes tipos de jogos usam formas distintas de representar posições interessantes. Influence Maps [Tozour 2001] e Occupancy Maps [Isla 2006] são exemplos de representações comumente utilizadas em RTS, enquanto sistemas de waypoints são mais frequentemente utilizadas em jogos de primeira e terceira pessoa.

Um Influence Map é uma representação espacial que armazena informações como: onde ocorreram batalhas importantes, quais regiões continuam inexploradas, ou onde o inimigo tem maior probabilidade de estar. Não existe um algoritmo único para criação Influence Map e esta representação pode ser aplicada de diferentes maneiras. As diversas formas de se construir um Influence Map depende das táticas e estratégias necessárias.

Os sistemas de waypoints, diferente do Influence Map, são usados principalmente para navegação através do ambiente. Eles são comumente representados por um grafo conexo e seus nós são posições no mundo. Assim, um sistema de waypoint representa todos os caminhos que os bots podem navegar através do ambiente. A fim de permitir comportamentos estratégicos, muitos desses nós são rotulados com informações táticas, criando uma representação espacial de locais interessantes, como um Influence Map faz.

Embora na maioria dos jogos as informações táticas sejam especificadas pelo projetista, existem alguns trabalhos que automatizam

<sup>1</sup> ex. regiões de batalhas, áreas de camper, área de comércio, etc

a definição de pontos interessantes. Lars Lidén [Liden 2002] apresenta como o pré-processamento da relação entre os waypoints pode ser usado para gerar posições táticas para bots em jogos de primeira pessoa. Ele gera informação de visibilidade estática entre waypoints, pré-calculando se um waypoint tem visibilidade de outro. Usando visibilidade é possível determinar quais locais são potencialmente seguros e quais são perigosos. Entretanto, a principal limitação desse trabalho é que a efetividade desse pré-cálculo depende da capacidade do projetista em especificar a posição dos nós de forma apropriada. Em outras palavras, embora o projetista não precise definir os pontos táticos, ele ainda necessita definir as informações de caminhos no ambiente. Além disso, as informações táticas geradas são estáticas e previsíveis.

Alguns trabalhos propõem a análise de posições táticas em tempo de execução [Tozour 2004] [Straatman et al. 2006] [Darken and Paull 2006]. Paul Tozour propôs uma base de dados espacial criada em tempo de execução para permitir que personagens tomem decisões sobre ela [Tozour 2004]. Essa abordagem usa camadas de grids 2D para armazenar informações como área de ocupação, área de busca e tantas outras também interessantes. Entretanto, ela apresenta sérios problemas para tratar um ambiente 3D.

Baseada em sistemas de waypoints, a avaliação dinâmica de posições táticas [Straatman et al. 2006] usa informações estáticas e dinâmicas sobre o mundo para criar decisões táticas em tempo real. Ela usa a proximidade para selecionar waypoints e visibilidade para pontuar essas posições, onde os maiores escores são os locais mais atrativos. O maior problema em usar essa avaliação em tempo de execução é devido ao grande número de operações custosas como *ray casts*. Assim, muitas restrições devem ser feitas para tratar o problema de performance, como: limitar o número de posições avaliadas, restringir personagens para certas áreas do mapa e usar tabela de visibilidade pré-calculada.

Existem trabalhos voltados a encontrar um tipo específico de posições táticas, como posições de cobertura [Darken and Paull 2006], enfatizando na adaptabilidade ao ambiente dinâmico. Essa técnica combina anotações do ambiente com algoritmos de grade de sensores. A anotação do ambiente ainda é feita pelo projetista enquanto o algoritmo de grade de sensores testa a posições de cobertura por *ray cast* em tempo real. Assim, um conjunto de locais escondidos pode ser gerado dinamicamente.

De formas diferentes, nossa abordagem descobre posições táticas com métodos baseados em agrupamento, minerando informações espaço-temporal. Nós estamos interessados em encontrar dois tipos distintos de posições táticas, que são as áreas de camper e as regiões de perigo. Assim, aplicamos duas diferentes técnicas para achar essas regiões.

### 3 Mineração de Dados Espaço-Temporal

Neste trabalho, utilizamos a mineração de dados espaço-temporal para formação de agrupamentos que irão representar pontos de interesse no ambiente. Um agrupamento é um grupo de dados que compartilha propriedades similares. Usualmente, cada agrupamento é um subconjunto de um conjunto inteiro de dados. O termo clusterização é bastante utilizado na comunidade de pesquisa para descrever métodos que agrupam dados não rotulados.

#### 3.1 DBSCAN

DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) é um importante método de clusterização baseado em densidade. Ele foi desenvolvido por Ester [Ester et al. 1996] e teve influência de muitos outros métodos baseados em densidade. A principal vantagem do DBSCAN é a capacidade de gerar agrupamentos de diferentes formas no conjunto de dados espaciais. A Figura 1 mostra alguns conjuntos de dados onde é possível descobrir os agrupamentos visualmente intuitivos.

DBSCAN procura por pontos de núcleo, que dão origem a um agrupamento, e tenta expandir o agrupamento inicial agregando os vizinhos mais próximos que atendem a algumas condições. Essas condições são baseadas em dois parâmetros: MinPts e Eps. MinPts

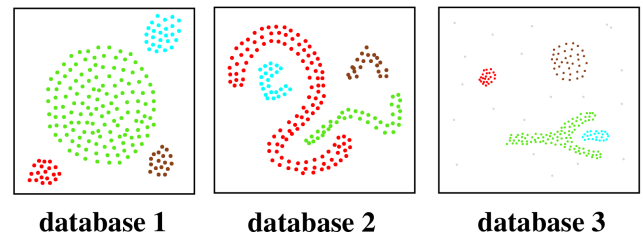


Figura 1: Agrupamentos descobertos pelo DBScan [Ester et al. 1996].

é a medida de densidade que indica quantos pontos precisam estar na vizinhança para que tal ponto seja considerado pertencente a um agrupamento. Eps é a distância utilizada para delimitar a vizinhança. A Figura 2 mostra como são expandidos os agrupamentos a partir dos pontos de núcleo.

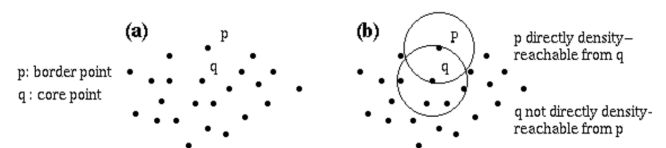


Figura 2: Formação de agrupamentos a partir de pontos de núcleo. [Ester et al. 1996].

#### 3.2 CB-SMoT

CB-SMoT (*Clustering Based - SMOt*) [Palma et al. 2008] usa a técnica de clusterização para identificar *stops*, de acordo com a definição de *stops* de [Spaccapietra et al. 2008]. CB-SMoT é um algoritmo que identifica seções lentas em uma trajetória, permitindo a associação dessas seções com lugares conhecidos em um ambiente.

Os parâmetros utilizados pelo algoritmo CB-SMoT são: avg, que define o percentual da velocidade média que representa um limite superior para um trecho de *stop* da trajetória; o minTime, que define o tempo mínimo que um trecho da trajetória deve estar abaixo do avg para ser considerado um *stop*; e SL, um fator que limita a velocidade para novos pontos pertencerem ao agrupamento.

### 4 Posições Táticas

Em jogos, posições táticas são frequentemente usadas para representar posições defensivas, pontos de cobertura ou regiões de camper. O projetista de cenário normalmente marca essas posições no ambiente baseado no seu conhecimento sobre a dinâmica do jogo. É comum que essas localizações sejam representadas como anotações em waypoints, fazendo diferentes usos para o grafo de caminho, já que os waypoints são originalmente projetados para planejamento de caminho.

Essas informações táticas poderiam também se referir a posições de inventários, onde armas e munição podem ser encontradas; localização de kit-médico, onde é possível recuperar a vitalidade; pontos de reconhecimento, onde uma grande área é visível a partir dali; e outros pontos relevantes no ambiente. Esses pontos podem também representar locais que devem ser evitados, como áreas expostas ou locais de emboscadas. São os tipos de jogos e as estratégias que determinam as posições táticas nos jogos, podendo serem eles tanto negativos como positivos.

Embora essas posições táticas possam ser associadas com waypoints, que são usados principalmente para navegação, algumas dessas localizações não são úteis como parte de um grafo de planejamento de caminho. Uma posição de camper, por exemplo, provavelmente não estará localizada em um caminho entre dois waypoints do ambiente, pois eles geralmente são o fim da linha. O mesmo ocorre com outras localizações, como pontos de cobertura. Assim, esses pontos não precisam necessariamente estar associados

com os dados de procura de caminho, na realidade, o planeamento de caminho poderia ser mais eficiente se essas informações estivessem separadas.

A definição de todos os pontos de interesse no ambiente pode produzir um grande número de pontos, e isso é necessário para alcançar uma boa qualidade de comportamento. Embora localizações como cantos escuros e pontos de cobertura sejam facilmente identificadas por seres humanos, um projetista pode não detectar todos os pontos táticos interessantes e essa especificação manual pode conduzir os personagens a apresentar comportamentos pouco convincentes. Com isso, surge uma necessidade em definir automaticamente pontos táticos nos ambientes.

A partir do registro das ações de jogadores humanos é possível obter informações de localizações taticamente interessantes. Se um personagem se mantém numa mesma região por algum tempo ou se a mesma região é utilizada diversas vezes ao longo da partida, existe uma grande probabilidade dela ser uma região de algum interesse. Com o conjunto de regiões táticas candidatas, é possível avaliar tanto a qualidade tática, como pontos de cobertura e pontos de emboscada, automaticamente a partir de algoritmos.

#### 4.1 Regiões Perigosas

É importante que os personagens consigam identificar quais regiões no ambiente são perigosas, para que eles possam evitá-las sempre que possível. As regiões com grande ocorrência de mortes são bastante críticas, pois normalmente representam áreas expostas que devem ser levadas em conta pelo personagem. Para identificar essas regiões levamos em consideração a densidade espacial das ocorrências de morte no ambiente.

Assim, utilizamos o DBScan, um algoritmo de agrupamento espacial baseado em densidade, para identificar as regiões perigosas a partir da mineração sobre os dados espaciais de mortes dos personagens no ambiente. Duas características do DBScan são importantes para realizar essas tarefas. A primeira é a formação de agrupamentos de formatos variados (não-esféricos), o que permite uma representação adequada dentro do ambiente. A segunda é a possibilidade de filtrar ruídos, ignorando casos isolados de mortes para a formação de agrupamentos relevantes.

#### 4.2 Pontos de Camper

Outro importante tipo de pontos que buscamos identificar neste trabalho são os pontos de camper. O objetivo dessa identificação é permitir que bots consigam executar comportamentos eficientes para a dinâmica do jogo e também consigam combater inimigos que se utilizam dessas regiões. Para isso, utilizamos o algoritmo CB-SMoT sobre a trajetória de cada personagem, identificando os pontos de baixa velocidade.

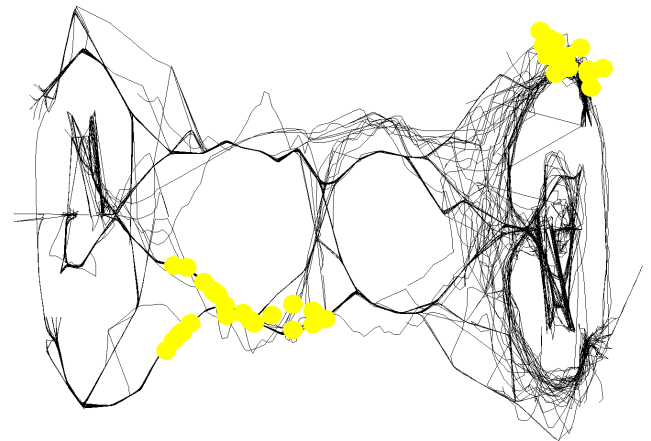
Uma característica dos pontos de camper é que eles são utilizados, geralmente, quando o personagem está parado ou em baixa velocidade, já que ele se dirige para esses pontos onde aguardam o aparecimento de um inimigo. Por isso, buscamos identificar esses pontos de baixa velocidade observando a trajetória de cada personagem. Também é característica importante a ocorrência de muitas mortes originadas nesses pontos, por isso, também é interessante associar os pontos de baixa velocidade com as regiões onde são causadas as mortes no ambiente, todavia, em nossa abordagem inicial utilizamos apenas os pontos de baixa velocidade nas trajetórias, o que efetivamente encontra os pontos de camper, mas poderia identificar também pontos que um personagem tenha ficado preso no ambiente.

### 5 Experimentos e Avaliações Preliminares

Os experimentos foram realizados com o jogo Unreal Tournament 2004, utilizando a interface GameBots [Kaminka et al. 2002] para a aquisição das informações do jogo. Foram realizadas partidas de 30 minutos em dois mapas distintos. Nestes experimentos, tanto o método baseado em densidade, quanto o método baseado

em velocidade apresentaram resultados interessantes. O DBScan conseguiu encontrar as regiões do cenário que ofereceram maior perigo, a partir da concentração de morte. Vale ressaltar que o DBScan foi bastante tolerante a ruído, quando seus parâmetros foram definidos de forma adequada, apesar da identificação desses parâmetros não ter sido um processo imediato.

Nas Figuras 3 e 4, que representam os ambientes utilizados, foi possível perceber que o DBScan encontrou zonas de perigo (representadas pelo conjunto de regiões destacadas) em campo aberto e nas proximidades do *respawn* de personagens, o que é bastante coerente para um jogo de primeira pessoa. O CB-SMoT também apresentou resultados muito interessantes, identificando os pontos de camper utilizados. Na Figura 5 foram encontrados pontos dentro das duas torres existentes no cenário e na Figura 6 foram encontrados pontos dentro de um pequeno bunker do cenário.



**Figura 3:** Regiões perigosas descobertas no mapa FaceClassic com o DBSCAN ( $Eps=0.01$ ,  $MinPoints=15$ ).

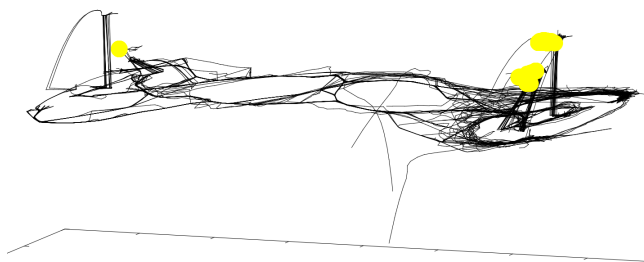


**Figura 4:** Regiões perigosas descobertas no mapa December com o DBSCAN ( $Eps=0.01$ ,  $MinPoints=6$ ).

Apesar dos bons resultados iniciais obtidos, a necessidade do ajuste manual dos parâmetros nos dois algoritmos foi o principal ponto negativo encontrado. No caso do DBSCAN, mesmo existindo trabalhos que analisam a estrutura de agrupamentos através da variação do  $Eps$ , os mesmos não foram utilizados nos experimentos. No caso do CB-SMoT, todos os parâmetros ainda precisam ser definidos manualmente, o que dificulta, mas não impossibilita, a identificação das regiões de interesse de forma totalmente automática.

### 6 Conclusões

A especificação de posições táticas dentro de ambientes virtuais é bastante importante para permitir comportamentos mais convincentes dos bots. Por isso, é interessante viabilizar métodos que permitam a definição desses pontos com a mínima interferência humana.



**Figura 5:** Regiões de camper descobertas no mapa *FaceClassic* com o CB-SMoT ( $avg=0.1$ ,  $minTime=12$ ,  $SL=1$ ).



**Figura 6:** Regiões de camper descobertas no mapa *December* com o CB-SMoT ( $avg=0.1$ ,  $minTime=20$ ,  $SL=1$ ).

Com este trabalho concluímos que é viável realizar a descoberta automática de posições táticas em ambientes virtuais, utilizando técnicas de agrupamento espacial e espaço-temporal para diminuir o trabalho do projetista, encontrando algumas vezes regiões não previstas e pouco intuitivas. A aplicação dessas técnicas torna possível o desenvolvimentos de bots adaptativos, que adquirem mais experiência sobre seu ambiente à medida em que jogam novas partidas. Assim, a aplicação do processo de descoberta de conhecimento sobre jogos é vista como uma área promissora, uma vez que diminui o esforço humano e aumenta o grau de aceitabilidade e convencimento sobre a inteligência dos bots. Alguns passos, entretanto, ainda precisam ser trilhados de modo a permitir que se diminua cada vez mais a intervenção humana na avaliação dos modelos gerados.

Dentre os principais trabalhos futuros podemos citar o estudo sobre o ajuste automático dos parâmetros dos algoritmos; a associação de pontos onde são causadas as mortes com os pontos de baixa velocidade na trajetória, para uma filtragem dos agrupamentos candidatos; e a utilização dessas informações para o raciocínio dos bots.

## Referências

- DARKEN, C. J., AND PAULL, G. H. 2006. Finding cover in dynamic environments. In *AI Game Programming Wisdom 3*, Charles River Media, 405–416.
- DILL, K., AND SRAMEK, A. 2004. Performing qualitative terrain analysis in master of orion 3. In *AI Game Programming Wisdom 2*, Charles River Media, 391–398.
- ESTER, M., PETER KRIEGLER, H., S. J., AND XU, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. AAAI Press, 226–231.
- ISLA, D. 2006. Probabilistic target tracking and search using occupancy maps. In *AI Game Programming Wisdom 3*, Charles River Media, 379–388.
- KAMINKA, G. A., VELOSO, M. M., SCHAFFER, S., SOLLITTO, C., ADOBBATI, R., MARSHALL, A. N., SCHOLER, A., AND TEJADA, S. 2002. Gamebots: a flexible test bed for multiagent team research. *Commun. ACM* 45, 1, 43–45.
- LIDEN, L. 2002. Strategic and tactical reasoning with waypoints. In *AI Game Programming Wisdom*, Charles River Media, 211–220.
- PALMA, A. L. T., ALVARES, L. O., AND BOGORNY, V. 2008. A Clustering-Based Approach for Discovering Interesting Places in Trajectories. Master's thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul.
- SPACCAPIETRA, S., PARENT, C., DAMIANI, M. L., DE MACEDO, J. A., PORTO, F., AND VANGENOT, C. 2008. A conceptual view on trajectories. *Data Knowl. Eng.* 65, 1, 126–146.
- STRAATMAN, R., BEIJ, A., AND VAN DER STERREN, W. 2006. Dynamic tactical position evaluation. In *AI Game Programming Wisdom 3*, Charles River Media, 389–404.
- TOZOUR, P. 2001. Influence mapping. In *Game Programming Wisdom Gems 2*, Charles River Media, 287–297.
- TOZOUR, P. 2004. Using a spatial database for runtime spatial analysis. In *AI Game Programming Wisdom 2*, Charles River Media, 381–390.