

Controle de um Foco de Atenção Móvel Utilizando *Subsumption Architectures*

CLAUDIO S. PINHANEZ *

Universidade de São Paulo
Instituto de Matemática e Estatística
Depto. de Ciência da Computação
Caixa Postal 20570 — São Paulo
Cep: 01498 — SP — Brasil
E-mail: pinhanez@ime.usp.br

Abstract. Subsumption architecture is a model of control based on the incremental construction of a network of simple processing elements assembled with special connectors, developed due to the research with highly reactive mobile robots. In this paper the employment of this methodology in computer vision is examined: a mobile focus of attention for detection of long edges in real outdoor scenes was implemented by a subsumption architecture. Some good results were obtained, in spite of the complexity of the images and the lack of extensive sensoring. The basic principles of subsumption architectures are also reviewed.

Têm atraído muita atenção os novos paradigmas de controle provenientes da pesquisa em navegação robótica, em especial no controle de sistemas altamente reativos fundeados na interação com o mundo real ([Chapnick 90]). Contudo, sua extensão e uso em outros campos além da robótica ainda é incipiente, em particular em processamento de imagens. Por outro lado, a utilização de câmeras móveis em visão computacional tem ganho muito impulso recentemente, apesar das dificuldades consideráveis encontradas no projeto de seus sistemas de controle.

Este trabalho é o resultado de uma pesquisa cujo objetivo é o estudo da adequação do uso de uma arquitetura representativa de tais métodos, conhecida como *subsumption architecture*, para o controle de uma câmera móvel, aqui simulada por um *foco de atenção* que percorre uma imagem de alta resolução.

Breves introduções aos conceitos fundamentais de *visão ativa* e *subsumption architectures* são apresentadas nas duas primeiras seções do trabalho. Segue-se uma descrição do problema real utilizado como referência para a avaliação do método proposto, seguido pelo detalhamento do sistema de controle implementado e de seus componentes. A penúltima seção apresenta resultados obtidos no processamento de imagens reais, e discute os méritos e deficiências

verificados. Futuros desenvolvimentos na pesquisa são também sugeridos, seguidos de uma lista de referências básicas para os diversos temas abordados no trabalho.

1 Visão Ativa

Sistemas de visão para robôs tradicionalmente se baseiam na captação de imagens através de uma câmera matricial fixa (em relação ao corpo do robô). Como consequência dessa estrutura, há que se negociar a amplitude do ângulo de visão em relação à resolução; contudo, mesmo quando se utilizam câmeras de resolução mais alta, o problema é transferido para a fase de processamento, inviabilizando, muitas vezes, algoritmos em tempo real.

Embora alguns pesquisadores sugiram o uso de câmeras com estruturas semelhantes à *fovea* do olho dos mamíferos [Sandini-Tagliasco 80], a principal corrente defende o uso de câmeras móveis, ou *visão ativa*, como método para superar tal deficiência ([Aloimonos et al. 87], [Ballard-Ozcandarli 88]). No entanto, o problema de controlar os movimentos da câmera de maneira útil e eficiente apresenta sérias dificuldades ([Clark-Ferrier 88]), e, muitas vezes, deve ser considerado em função dos propósitos do sistema ([Aloimonos 90]).

O objetivo da pesquisa é investigar como pode ser controlado o foco de atenção de uma câmera móvel. O controle deve ser robusto, capaz de operar com imagens do mundo real, incluindo tanto ruído e deficiências na resolução como obstruções ocasionais, mas com um custo computacional compatível

*Este trabalho foi parcialmente realizado no Depto. de Máquinas Controladas por Computador da Faculdade de Engenharia da Universidade de Osaka, Japão (*Denshi Seigyō Kikai, Kōgakka, Kōgakubu, Osaka Daigaku*), através do auxílio da bolsa para pesquisadores (*Kenkyu-sei*) do Ministério de Educação do Japão (*Mombushō*), complementada por bolsa concedida pela FAPESP (processo 90/2940-7).

com sua operação em tempo real. Idealmente, o sistema de controle deve ainda exibir padrões de movimentos equivalentes aos movimentos sacádicos e de perseguição dos mamíferos (conforme mostrado em [Noton-Stark 71]).

Paralelamente, há interesse em averiguar a adequação de utilização de sistemas simulados de visão ativa no reconhecimento de imagens de alta resolução de grande amplitude, através da concentração de esforços em áreas de interesse encontradas por um foco de atenção móvel, evitando-se dessa maneira o processamento de toda a matriz de intensidades.

Como as principais propostas para o controle de visão ativa ([Ballard 89], [Ballard-Ozcanarli 88], [Clark-Ferrier 88], [Burt 88]) não satisfazem os requisitos do sistema desejado, optou-se por estudar mais detalhadamente propostas alternativas de controle de sensores. Dentre estas, foi escolhida a *subsumption architecture*, proposta originalmente para controle de robôs móveis por Rodney Brooks.

2 Subsumption Architectures

Sistemas empregando *subsumption architectures* surgiram em meados da década de 80, no laboratório de robôs móveis do MIT (Brooks 86), com o objetivo de demonstrar que comportamento complexo de robôs operando no mundo real pode ser obtido com o uso de redes compostas de elementos simples ligados diretamente aos sensores e atuadores ([Brooks-Flynn 89]), sem necessidade de níveis hierarquizados de abstração nem de controle central. Pode-se mapear seus fundamentos teóricos nos trabalhos pioneiros de Marvin Minsky, sobretudo em "The Society of Mind" [Minsky 86].

O elemento básico de *subsumption architectures* é uma *Augmented Finite State Machine*, ou *AFSM*, obtida pela união de uma máquina de estados finitos a um conjunto de *registradores* e de *temporizadores* (figura 1). Registradores são alterados pela própria *AFSM* ou por mensagens recebidas através de ligações, representadas por meio de setas: setas incidentes indicam que o registrador é modificado pela ação da chegada de mensagens, enquanto que setas partindo indicam valores que são transmitidos a registradores de outras *AFSM*'s.

Uma *AFSM* permanece inativa até que uma mensagem chegue a algum de seus registradores de entrada, ou até que um de seus temporizadores dispare. A partir de então, a rotina programada examina os registradores e, a partir de seus valores, altera registradores e/ou ativa temporizadores, retornando depois ao estado inativo.

Além da conexão direta, registradores podem ser interligados através dos *conectores* indicados na

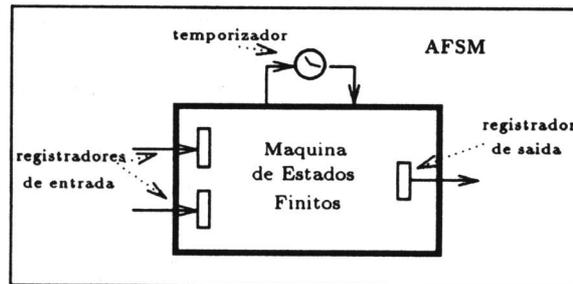


Figura 1: Diagrama de uma *AFSM* – *Augmented Finite State Machine*.

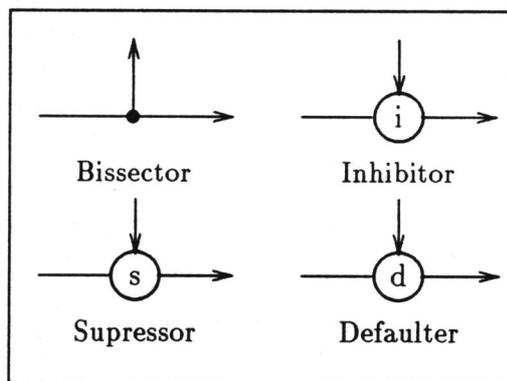


Figura 2: Diferentes tipos de conectores utilizados na interligação de *AFSM*'s.

figura 2. Há quatro tipos de conectores:

Bissetor : a mensagem que circula pelo fio é duplicada e enviada às duas direções indicadas;

Inhibitor : uma mensagem recebida pelo fio interceptor inibe, por um curto período de tempo, a comunicação normal;

Supressor : a mensagem recebida pelo fio interceptor suprime o fluxo normal de comunicação, tomando o seu lugar;

Defaulter : realiza a função inversa do conector **Supressor**.

A única restrição quanto à interligação de conectores e registradores é que não devem haver ciclos (*loops*). Idealmente, a rede deve ser implementada por um conjunto de processadores, um para cada *AFSM*, assíncronos, mas com *clocks* semelhantes. O tempo de inibição e de supressão deve corresponder a 2 ticks, de forma que uma *AFSM* pode controlar uma

conexão se enviar mensagens à taxa máxima ([Brooks 89]). Contudo, ressalte-se que muitos dos sistemas construídos pela equipe de Brooks utilizaram-se de simuladores de *subsumption architectures* ([Brooks 86], [Angle-Brooks 90], [Mae 89]).

Subsumption architectures devem ser empregadas preferencialmente na construção de sistemas altamente reativos, que sensoriem e atuem em ambientes reais. Redes de *AFSM*'s são incrementalmente construídas, partindo-se de uma configuração básica muito simples, à qual são conectados novas *AFSM*'s, que ampliam o repertório de comportamentos do sistema. Não há necessidade de se manter uma hierarquia de qualquer tipo quando da inclusão de uma *AFSM*, e, tipicamente, se observam conexões entre os diversos sub-sistemas da rede. Um bom exemplo do processo de utilização de *subsumption architectures* é visto em [Brooks 89], no qual um robô articulado de seis pernas é controlado por uma rede de 57 *AFSM*'s.

3 Detectando Arestas Longas com um Foco de Atenção Móvel

Um problema real foi escolhido como “bancada de teste” para o sistema de controle de foco de atenção: o reconhecimento de pontos de referência em imagens com prédios e outros marcos arquitetônicos. Através da identificação de construções, a navegação em um ambiente urbano – seja automática por um robô, seja como auxílio a um motorista — pode ser bastante facilitada.

Para o reconhecimento de fachadas em situações reais, é comum buscar determinar, inicialmente, os pontos de fuga da imagem, a partir do processamento de segmentos de reta detectados ([Nakatani et. al. 80], [Brillaut-O'Mahony 91]). Com base nos pontos de fuga, a segmentação das fachadas torna-se viável, permitindo-se também que a distorção resultante da perspectiva seja compensada. Fachadas desse modo “regularizadas” são mais facilmente pareadas (*matching*) com modelos previamente armazenados.

O sistema desenvolvido até agora tem por objetivo realizar a primeira fase, ou seja, encontrar segmentos de retas longos presentes na imagem original. A decisão de se utilizar um sistema de visão ativo para detecção de segmentos de reta advém de duas considerações básicas:

Obstruções: em situações reais, é de se esperar a presença de elementos que obstruam ocasionalmente a fachada, como veículos e pessoas muito próximas da câmera; assim, um sistema que capta uma única imagem de alta resolução e a processa por um certo período de tempo é mais sensível do que um outro que sensoria pequenas

áreas à medida do necessário.

Robustez: os métodos clássicos de detecção de linhas (transformada de Hough, etc.), necessitam de calibração por vezes exaustiva, e são bastante sensíveis ao número grande de segmentos que aparecem em uma fachada típica.

Considerando as dificuldades mecânicas e eletrônicas da construção de uma câmera real, optou-se por testar as técnicas de controle do foco de atenção em um sistema simulado de visão ativa, em que um *foco de atenção* “passeia” por uma imagem de alta resolução. Este foco de atenção decide uma direção a ser percorrida, sensoriando somente um número pequeno de *pixels* a cada passo, e considerado também o estado atual da rede de *AFSM*'s que controla o sistema.

4 O Foco de Atenção Móvel

O foco de atenção que percorre a imagem de alta resolução em busca de segmentos de retas longos, é constituído por uma rede de 26 *AFSM*'s; um diagrama simplificado é mostrado na figura 3, na qual cada conjunto de *AFSM*'s do mesmo tipo é representado por um único *AFSM*.

O sensoriamento da imagem é feito de quatro maneiras distintas por quatro tipos de sensores (na figura, marcados pela tarja negra no canto superior esquerdo). Há também um *AFSM* responsável pela movimentação do centro do sensoriamento, e outro pela geração das coordenadas das extremidades de segmentos de reta detectados pelo sistema (*AFSM*'s com tarja negra no canto inferior direito).

Os princípios básicos de *subsumption architecture* (conforme [Brooks 91]) consideram que não devem haver “modelos do mundo”; os módulos devem, sempre que possível, comunicar-se pela interação com o “mundo real”, e não diretamente entre si; o comportamento “inteligente” deve ser obtido pela dinâmica da interação com o mundo; etc. Estas características foram, na medida do possível, respeitadas na implementação, conforme pode-se verificar na descrição sucinta dos diversos sub-sistemas de *AFSM*'s elaborada a seguir:

Gerador de Passeios Aleatórios

A movimentação básica do foco de atenção é obtida pela iteração dos seguintes *AFSM*'s:

mo – move : sempre que recebe uma mensagem — que indica uma direção — o *AFSM mo* altera o centro do foco de atenção para o *pixel* vizinho correspondente à direção recebida, utilizan-

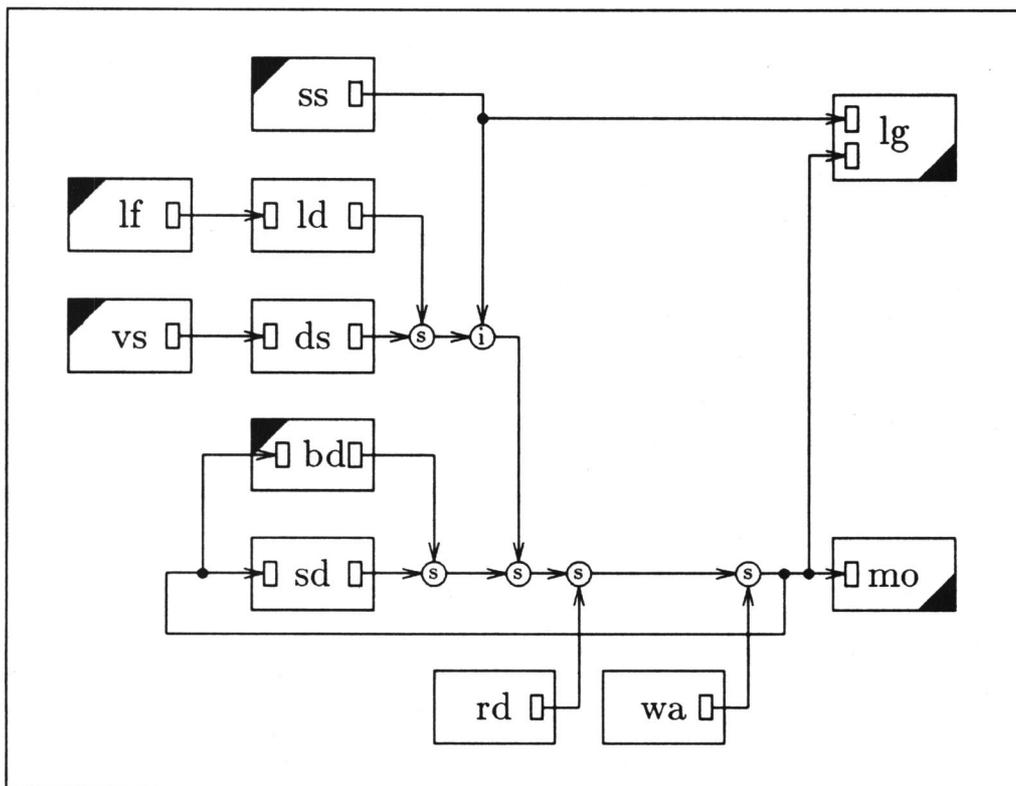


Figura 3: Diagrama simplificado do foco de atenção.

do um algoritmo similar ao de Bresenhan para traçado de retas.

sd – *same direction* : funciona como um repetidor, retransmitindo sempre a direção que recebe; assim, uma vez iniciado o movimento do foco em alguma direção, a interação entre **mo** e **sd** é suficiente para manter o foco movendo-se naquela direção até que algum outro *AFSM* a altere, através do envio de uma mensagem supressiva para o canal que liga **sd** à **mo**.

rd – *random direction* : a cada 1000 unidades de tempo, gera uma direção aleatória, evitando, assim, que o foco fique preso em eventuais *loopings*.

wa – *wall avoider* : responsável em evitar que o sistema saia para fora da área da imagem, refletindo a direção sempre que o foco se aproximar demais da borda; como está conectado pelo nó do tipo *Supressor* mais próximo de **mo**, tem prioridade absoluta sobre os demais.

Não há sensores de imagem neste sub-sistema, do que se conclui que o movimento gerado pelo mes-

mo independe da imagem, e se constitui em um passeio retilíneo refletido pelas bordas, alterado somente pela intervenção aleatória de **rd**.

Direcionador a Regiões de Alta Variância

O foco de atenção deve ser, de alguma forma, atraído para regiões da imagem que possuam maior riqueza em detalhes, o que é realizado pelo sub-sistema composto pelos *AFSM*'s abaixo:

vs – *variance sensor* : mede a variância da imagem, a uma distância de 50 *pixels* do centro do foco de atenção, e envia o valor da variância para **ds** se este for maior que um certo limiar (ajustado adaptativamente). Há quatro sensores do tipo **vs**, dispostos em cruz, cada um examinando 27 *pixels* a cada 10 unidades de tempo.

ds – *detail searcher* : considerando a posição e o valor da variância de cada **vs** que lhe enviou mensagem, calcula a resultante ponderada das direções de variância, enviando-a para **mo**.

Com a inclusão deste sub-sistema, obtém-se um padrão característico no qual as regiões com maior

informação são exaustivamente percorridas pelo foco de atenção. Sua simplicidade, porém, admite a ocorrência de *loops*, razão pela qual *rd* tem poder supressivo sobre este sub-sistema de direcionamento a regiões de alta variância.

Detector de Arestas

Conforme será visto mais adiante, o foco de atenção será capaz de identificar arestas somente quando se movimentar **exatamente** sobre as mesmas. Desse modo, é necessário que o foco de atenção, ao cruzar uma aresta, mude bruscamente a direção, com o objetivo de segui-la. Os *AFSM*'s responsáveis por este serviço são:

lf – *line finder* : 12 *AFSM*'s deste tipo são responsáveis pelo sensoriamento de direções a cada 30°. Cada **lf** examina 8 *pixels* a cada 3 unidades de tempo, quatro de cada lado da possível aresta: se a média de um lado for maior, significativamente, que a do outro, envia o valor da diferença para **ld**, a menos que a variância dos valores em qualquer dos lados seja muito alta.

ld – *line director* : recebe as mensagens dos **lf**'s ativados, e gera a direção correspondente ao valor máximo.

Na medida em que é mais interessante seguir uma aresta encontrada do que dirigir-se a uma região de maior variância, as direções provenientes de **ld** têm efeito supressivo àquelas geradas por **ds**.

Corretor de Direção

O sub-sistema detector de arestas descrito acima tem capacidade de alterar a direção do foco de atenção somente para múltiplos inteiros de 30°. Para superar essa limitação, foi concebido o *AFSM* **bd** :

bd – *best direction* : examina 7 *pixels* localizados em uma linha perpendicular à direção atual do movimento (obtida no seu registrador de entrada), e mede gradientes considerando os 5 *pixels* centrais, os 5 *pixels* superiores e os 5 inferiores. Com base no gradiente máximo, envia, se necessário, a nova direção (uma pequena melhora da atual), suprimindo, nesse caso, a mensagem de **sd**.

Embora projetado com o intuito de atenuar as mudanças bruscas de direção produzidas por **ld**, verificou-se que este *AFSM*, na posição que foi colocado, permite também que sejam percorridas arestas encontradas por acaso ou arestas de baixa curvatura.

Sensor de Movimento sobre Arestas

Unindo todos os sub-sistemas explicados até agora, obtém-se um comportamento para o foco de atenção caracterizado por alterações constantes na direção, correspondentes às múltiplas atuações de **ld** e **ds**. Como o objetivo era determinar segmentos de reta longos, foi implementado:

ss – *step sensor* : este *AFSM* detecta se as regiões de cada lado da direção do movimento possuem baixa variância e uma média de intensidade significativamente diferente. Dois **ss**'s são empregados para examinar a imagem, cada um acessando 10 *pixels* por unidade de tempo.

A ativação de um **ss** indica que uma aresta está sendo percorrida, e, através de um nó *Bisector*, inibe-se a alteração da direção comandada tanto por **ld** como por **ds**. As correções de **bd** continuam ativas, permitindo que o movimento seja corrigido e se adapte à variações pequenas na curvatura.

Gerador de Segmentos de Reta

Para a geração dos segmentos de reta, tem-se:

lg – *line generator* : monitorando o sensor **ss** de movimento sobre arestas, este *AFSM* detecta longas seqüências de ativação, gerando as coordenadas de início e fim do segmento de reta por elas definido. Interrupções temporárias são permitidas e somente linhas com mais de 20 *pixels* são geradas; contudo, mudanças bruscas de direção, monitoradas pelo registrador ligado à entrada de **mo**, causam quebra do segmento em detecção.

A saída de **lg** constitui-se das coordenadas inicial e final do segmento detectado. Do ponto de vista do foco de atenção, podemos considerar esta saída como uma atuação do sistema sobre o mundo real; em extensões futuras, deverão ser definidos registradores que recebem o segmento encontrado, permitindo seu uso por outros *AFSM*'s.

A ligação de **lg** a diferentes sub-sistemas (e internamente aos mesmos) é típica das *subsumption architectures*, e pode ser encarada como uma implementação da idéia de *sociedades de agentes* ([Minsky 86]).

Repare-se que o foco de atenção não armazena de nenhuma forma as regiões pelas quais já passou: poderá, dessa forma, percorrer e gerar várias vezes a mesma aresta. Na verdade, este é um fator de melhora da robustez, na medida em que, no caso de

uma câmara ativa real, ruídos e obstruções variam com o tempo.

Como a *subsumption architecture* exige a existência de múltiplos processos assíncronos, executando paralelamente, o foco de atenção móvel foi implementado com auxílio de um simulador de *subsumption architectures*, projetado e programado em linguagem "C" pelo autor, cuja descrição pode ser encontrada em [Pinhanez 1992]. Nesse simulador, uma *unidade de tempo* corresponde ao tempo de execução de todas as rotinas programadas nos *AFSM's*, do instante de ativação até a volta ao estado inativo.

5 Testes com Imagens Reais

O foco de atenção móvel foi desenvolvido e testado utilizando-se imagens de conjuntos de prédios, obtidas pela digitalização de fotos coloridas tiradas em um centro comercial de Toyonaka, Osaka. São imagens de 512×480 *pixels*, com 256 níveis de cinza.

As figuras 4 e 5 mostram as imagens originais (nestas figuras exibidas em uma versão binarizada por um algoritmo simples de *half-toning*), e os segmentos de reta longos detectados pelo foco de atenção, plotados na posição correspondente da imagem. Foram percorridos, em cada uma das imagens, 200.000 *pixels*, correspondentes a 200.000 unidades de tempo.

Conforme pode-se deduzir das informações da seção anterior, o foco de atenção examina uma média de 70 *pixels* por unidade de tempo e, portanto, um total de 14×10^6 *pixels* foram utilizados em todo o processo. Portanto, para estas imagens de 245.760 *pixels*, a média de exame é, aproximadamente, 57 *pixels* para cada *pixel* da imagem, inferior, por exemplo, ao de uma convolução discreta com uma janela de 8 *pixels* de lado.

Os resultados podem ser considerados bons, dada a complexidade das imagens e a baixa taxa de sensoriamento indicada acima. O foco de atenção móvel mostrou-se capaz de detectar segmentos de reta bastante longos, muitos dos quais apresentam interrupções na imagem original. Em particular na figura 5, os longos segmentos de reta obtidos na parte superior dos grupos horizontais de janelas não estão presentes na imagem original, constituindo-se, do ponto de vista de variação de intensidades, em algo similar a uma reta tracejada. A inércia para a alteração da direção de movimento, incorporada principalmente à *ss*, é a responsável por esse interessante capacidade.

Note-se que, na figura 4, os resultados fracos na determinação dos segmentos do prédio da porção esquerda da imagem devem-se ao fato de que *ss* não é capaz de encontrar linhas finas, mas somente, arestas

entre regiões de intensidade uniforme.

Acredita-se que os segmentos gerados possam produzir bons resultados se forem utilizados como entrada de um algoritmo de detecção de pontos de fuga; contudo, para manter a coerência do método — e portanto, sua robustez —, deveria-se buscar uma implementação de um algoritmo de extração de pontos de fuga também baseado em *subsumption architectures*.

Para avaliação da robustez do foco de atenção, decidiu-se testá-lo em uma imagem substancialmente diferente, no caso a de um estudante trabalhando em frente a *workstations*, mostrada na figura 6. Sem qualquer calibração ou ajuste, obteve-se um resultado bastante bom, particularmente considerando-se que a ocorrência de segmentos longos, nesta imagem, é muito menor.

A figura 7 exhibe os percursos do foco de atenção para as imagens das figuras 4 e 6. É interessante que, apesar de terem sido percorridos 200.000 *pixels* (equivalentes a 81% da imagem), não há quase passagem por áreas de intensidade uniforme, indicando que o foco de atenção concentra-se nas regiões de arestas e de alta variação.

6 Futuros Desenvolvimentos

O foco de atenção móvel necessita ainda ser testado em uma maior variedade de condições, a fim de melhor estimar a robustez e a capacidade de detecção do sistema. Em particular, experiências com imagens deterioradas devem ser feitas, bem como examinar sua sensibilidade a imagens de diferentes tipos de objetos e situações.

A questão de eficiência também precisa ser atacada, em especial, a dos algoritmos internos do simulador. Atualmente as mensagens são roteadas explicitamente, através de uma estrutura de dados relativamente complexa; melhoras substanciais para alguns casos particulares de redes já são possíveis, embora o caso geral ainda não esteja resolvido.

A maior questão em aberto é, no entanto, se *subsumption architectures* podem ser usadas para problemas mais complexos de reconhecimento, ou, ao menos, se podem ser conectadas a outros tipos de representação. Duas direções, portanto, se apresentam, a primeira consistindo em adicionar comportamentos cada vez mais complexos a rede de *AFSM's*, sem, contudo, alterar a metodologia. A segunda possível direção aponta no sentido de utilizar *subsumption architectures* em visão de baixo nível, gerando primitivas simples como segmentos de reta, contornos e agrupamentos.

Subsumption architectures é um campo ainda relativamente novo e inexplorado, embora já seja pos-

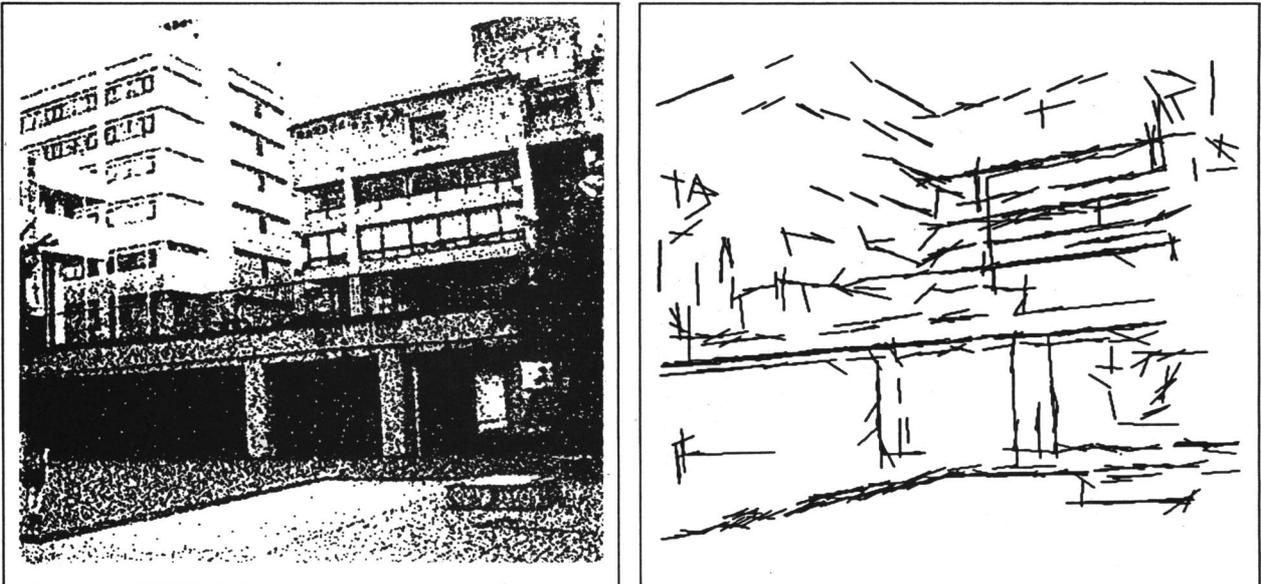


Figura 4: Imagem de um conjunto de prédios (512 x 480, com 8 bits por *pixel*, binarizada) e as linhas detectadas pelo foco de atenção móvel, para 200.000 unidades de tempo.

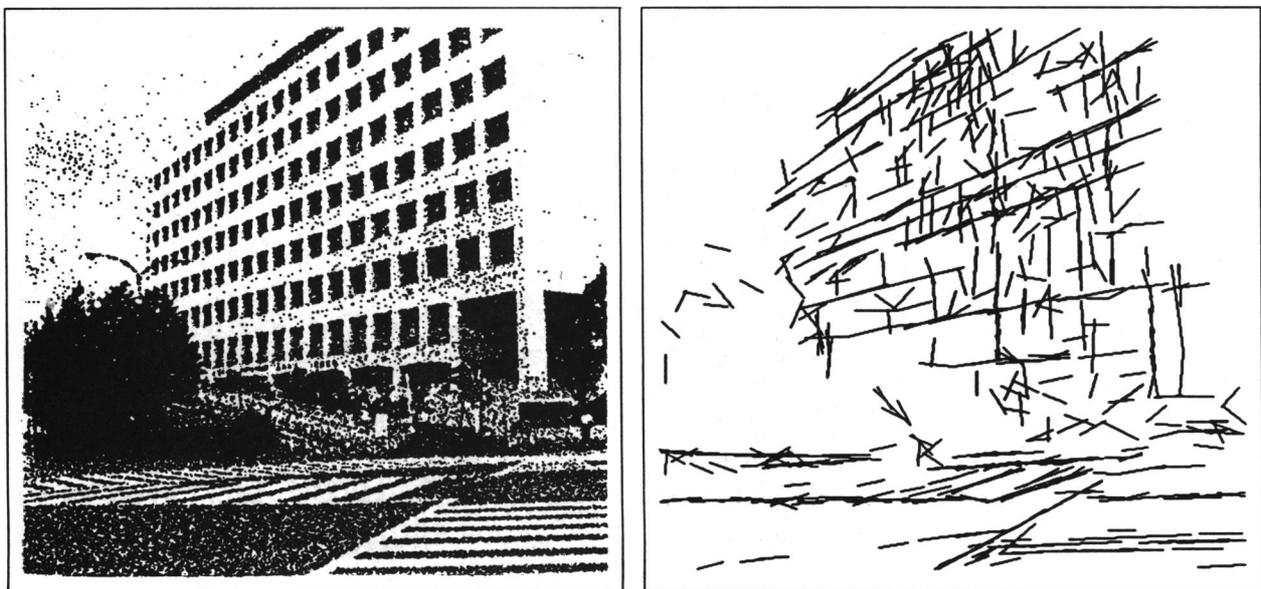


Figura 5: Imagem de um outro conjunto de prédios (512 x 480, com 8 bits por *pixel*, binarizada) e as linhas detectadas pelo foco de atenção móvel, para 200.000 unidades de tempo.

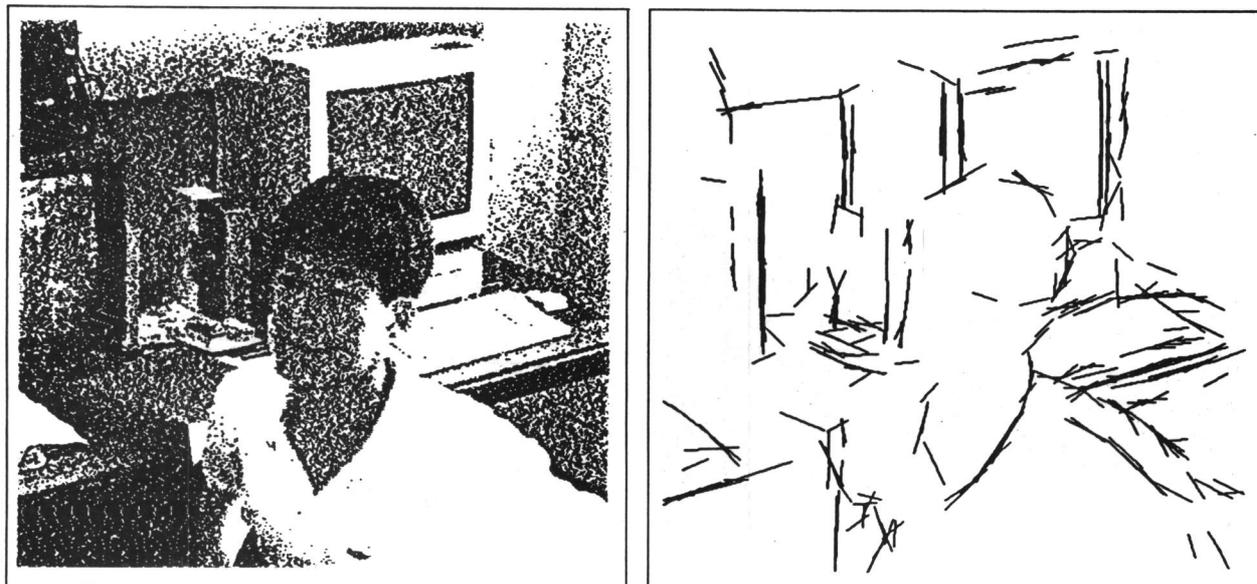


Figura 6: Imagem de um estudante trabalhando em frente a *workstations* (512×480 , com 8 bits por *pixel*, binarizada) e as linhas detectadas pelo foco de atenção móvel, para 200.000 unidades de tempo.

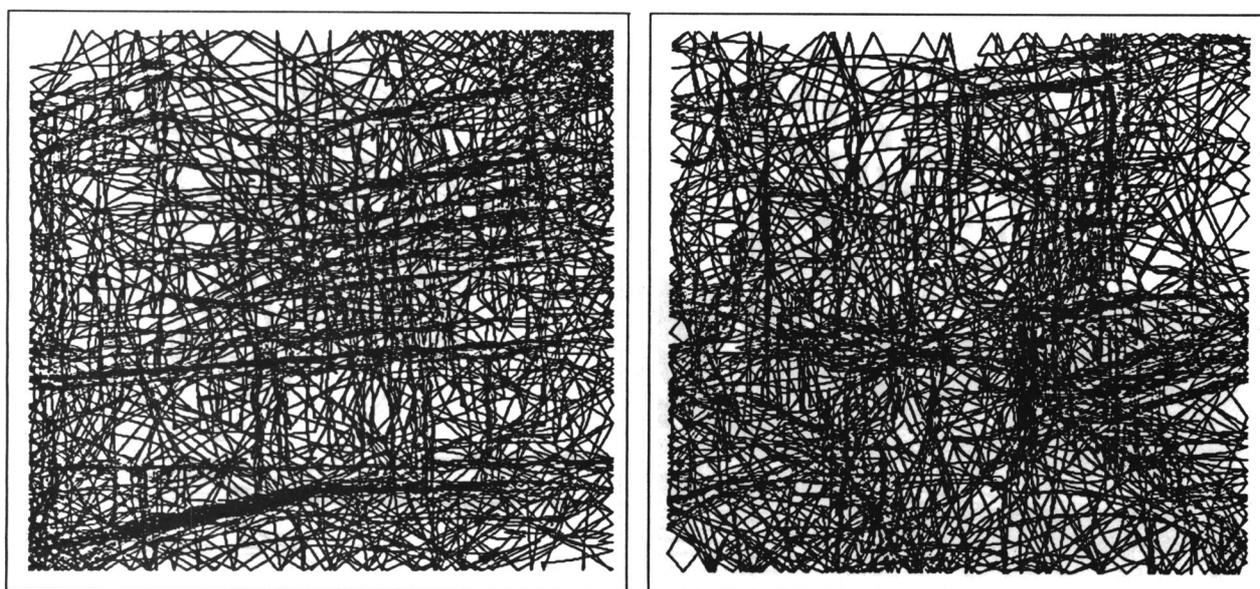


Figura 7: Traçado do percurso do foco de atenção móvel para as imagens da figura 4 (esq.) e da figura 6 (dir.), para 200.000 unidades de tempo.

sível encontrar na literatura sugestões para soluções de problemas consideravelmente mais difíceis como planejamento, aprendizado e representação do mundo. Este sistema de foco de atenção também pode ser usado como bancada de teste e desenvolvimento de tais métodos.

Finalmente, gostaria de dizer que os resultados obtidos até agora foram muito superiores às expectativas iniciais do autor, tendo em vista as limitações de sensoriamento e a simplicidade da rede de *AFSM*'s implementada. Do início da pesquisa até agora, o foco de atenção apresentou progressos notáveis, menos por esforços de calibração e ajuste do que por mudanças na configuração geral do sistema, o que considero um fator bastante positivo a favor do uso de *subsumption architectures*.

Agradecimentos

A orientação do Prof. Yoshiaki Shirai, da Universidade de Osaka, foi fundamental na escolha do modelo sobre o qual foi testado o foco de atenção móvel. Também gostaria de agradecer aos membros de sua equipe, em particular pelo apoio no processo de digitalização das imagens utilizadas neste trabalho, e ao Prof. Junior Barreira, do IME-USP, pelas sugestões relativas a este artigo.

Referências

- [Aloimonos 90] Aloimonos, J. *Purposive and Qualitative Active Vision*. In: *Proc. of the 1990 DARPA Image Understanding Workshop*. Pittsburgh, USA. p. 816-828. September, 11-13, 1990.
- [Aloimonos et al. 87] Aloimonos, J.; Weiss, I. & Bandyopadhyay, A. *Active Vision*. In: *Proc. of the First International Conference in Computer Vision (ICCV)*. London, UK. p. 35-54. June, 1987.
- [Angle-Brooks 90] Angle, C. & Brooks, R. *Small Planetary Rovers*. In: *Proc. of IEEE Int. Workshop on Intelligent Robots and Systems - IROS'90*. Tsuchiura, Ibaraki, Japan. p. 383-388. July, 3-6, 1990.
- [Ballard 89] Ballard, D. *Reference Frames for Animate Vision*. In: *Proc. of 11th IJCAI*. Detroit, USA. Vol. II. p. 1635-1641. August, 20-25, 1989.
- [Ballard-Ozcandarli 88] Ballard, D. & Ozcandarli, A. *Eye Fixation and Early Vision: Kinetic Depth*. In: *Proc. of the Second ICCV*. Florida, USA. p. 524-531. December, 5-8, 1988.
- [Brillault-O'Mahony 91] Brillault-O'Mahony, B. *New Method for Vanishing Point Detection*. In: *Computer Graphics, Vision and Image Processing (CGVIP): Image Understanding*, 54(2). September, 1991.
- [Brooks 86] Brooks, Rodney. *A Robust Layered Control System for a Mobile Robot*. In: *IEEE Journal of Robotics and Automation*, RA-2(1):14-23. March, 1986.
- [Brooks 89] Brooks, Rodney. *A Robot that Walks: Emergent Behaviour from a Carefully Evolved Network*. In: *Neural Computation*, 1(2):253-262. Summer, 1989.
- [Brooks 91] Brooks, Rodney. *New Approaches to Robotics*. In: *Science*, (253):1227-1232. September, 1991.
- [Brooks-Flynn 89] Brooks, R. & Flynn, A. *Robot Beings*. In: *Proc. of IEEE/RSJ Int. Workshop on Intelligent Robots and Systems - IROS'89*. Tsukuba, Japan. p. 2-10. September, 4-6, 1989.
- [Burt 88] Burt, P. *Algorithms and architectures for smart sensing*. In: *Proc. of the 1988 DARPA Image Understanding Workshop*. p. 139-153. 1988.
- [Clark-Ferrier 88] Clark, J.J. & Ferrier, N.J. *Modal Control of an Active Vision System*. In: *Proc. of the Second ICCV*. Florida, USA. p. 514-523. December, 5-8, 1988.
- [Chapnick 90] Chapnick, Philip. *Herbert, Murphy and Periplaneta Computatrix*. In: *AI Expert*, (12/90):23-25. December, 1990.
- [Mae 89] Maes, Pattie. *How to do the Right Thing*. In: *Connection Science*, 1(3):291-323. 1989.
- [Minsky 86] Minsky, Marvin. *The Society of Mind*. Simon & Schuster. New York, USA. 1986.
- [Nakatani et al. 80] Nakatani, H.; Kimura, S.; Saito, O.; Kitahashi, J. *Extraction of Vanishing Point and Its Application to Scene Analysis Based on Image Sequence*. In: *Proc. of Fifth International Conference on Pattern Recognition*. Miami Beach, USA. p. 370-372. December, 1-4, 1980.
- [Noton-Stark 71] Noton, D. & Stark, L. *Scanpaths in Eye Movements during Pattern Perception*. In: *Science*, (171):308-311. 1971.
- [Pinhanez 92] Pinhanez, Claudio S. *Um Simulador para Subsumption Architectures*. — a ser publicado — . 1992.
- [Sandini-Tagliasco 80] Sandini, G. & Tagliasco, V. *An Anthropomorphic Retina-like Structure for Scene Analysis*. In: *Computer Graphics, Vision and Image Processing (CGVIP)*, (14):365-372. 1980.