Relatório Parcial

Processos atencionais e aprendizado de máquina para sistemas robóticos

Aluno: Erik de Godoy Perillo
Orientadora: Profa. Dra. Esther Luna Colombini

Instituto de Computação
Universidade Estadual de Campinas

9 de dezembro de 2016

1 Introdução

A capacidade de percepção e construção de um modelo da realidade ao seu redor é fundamental para que sistemas robóticos interajam com o ambiente e executem tarefas diversas e complexas que podem ter as mais variadas utilidades para os humanos. Um componente fundamental para isso é a habilidade de dar foco apenas ao relevante, evitando assim o processamento desnecessário de enormes quantias de dados.

A atenção é um processo que faz parte do dia a dia de diversos seres vivos em diversas maneiras e é razoável inspirar-se nela para a construção de mecanismos semelhantes para a construção de sistemas de inteligência artificial em máquinas. Tal área tem sido foco de estudo há anos, resultando em diversas teorias em psicologia sobre a atenção humana que inspiraram a implementação de modelos computacionais bem sucedidos.

Neste trabalho, objetivamos construir um modelo atencional eficiente. Com base no modelo, implementaremos um *framework* atencional para robôs móveis que permita o uso da seleção em tempo real dos estímulos mais relevantes para as mais diversas tarefas que o robô possa executar. No trabalho atual focamos na atenção visual, mas o objetivo final do sistema é que ele funcione para outros sensores. O *framework* também contará com um módulo de reconhecimento de objetos que poderá ser substituído.

1.1 Objetivos da primeira parte do projeto

Os objetivos principais para o primeiro semestre do trabalho eram:

- Revisão bibliográfica sobre teorias sobre a atenção e diversos modelos.
- Escolha das técnicas mais adequadas para o processo atencional e o reconhecimento de objetos.
- Implementação de um modelo atencional.

Há uma quantidade surpreendente de avanços recentes na área de modelos de saliência visual. Entender os avanços mais relevantes é importante para a obtenção de um sistema atencional eficiente, então foi requerido mais tempo que o previsto para essa parte. Assim, todas as etapas previstas tiveram avanço, com exceção da parte de reconhecimento de objetos, a qual optamos por deixar para mais tarde pois a mesma serve como um complemento para nosso trabalho e é de menor relevância que o componente atencional. As atividades desenvolvidas são mais detalhadas a seguir.

2 Resumo das atividades

2.1 Revisão Bibliográfica

Dois dos conceitos importantes para o entendimento da literatura do meio são:

- *features*: Características básicas que formam entidades visuais, como cor (verde, azul), orientação (horizontal, vertical), contraste, tamanho.
- *Bottom-up vs. Top-down*: Por componente *bottom-up* de atenção entende-se saliências instintivas percebidas por mudanças e/ou contrastes muito grandes em uma cena. O componente *top-down* é aquele que dá saliência variável às *features* de acordo com a meta do agente do momento.

A maioria dos modelos computacionais baseia-se em teorias formadas na psicologia. Duas das mais famosas são a *Filter Integration Theory* (FIT) e a *Guided Search*. Ambas provêm contribuições importantes para o entendimento dos processos de saliência visual. Diversos modelos computacionais foram criados baseando-se em ideias delas. Começou-se então por elas.

A FIT indica basicamente que se a busca de um objeto de interesse em uma cena for por apenas uma *feature*, a localização é feita em tempo instantâneo. Entretanto, se o objeto de interesse for composto por múltiplas *features* a serem buscadas (e.g. uma linha horizontal verde), a localização do objeto é feita em tempo linear.

Já Guided Search diz que buscas por conjunções de *features* são na verdade mais rápidas pois a combinação das features gera um sinal de saliência mais forte no campo visual humano.

O VOCUS é um modelo atencional computacional para a detecção de saliências visuais. A maioria dos seus componentes é feita com base nas ideias da FIT. Nesta primeira etapa, exploramos seu componente *bottom-up*. Ele lida com as *features*: cor, intensidade, orientação. Seus mapas de saliência são calculados com base nessas *features* e em diversas dimensões da imagem.

2.2 Modelo atencional

A carga teórica adquirida foi útil para a concepção do nosso modelo, chamado de *att*. Muitos mecanismos foram inspirados no VOCUS, lidando com as mesmas *features* e em múltiplas dimensões da imagem.

2.2.1 Extração de *features*

O modelo extrai os seguintes mapas de uma certa imagem: luminância, luminância invertida, vermelho, verde, amarelo, azul, orientações vertical, horizontal, 45°e 135°. Os de luminância e cor podem ser extraídos pela conversão da imagem para o espaço de cor LAB e as orientações são extraídas usando-se filtros de Gabor.

2.2.2 Extração de saliência

Para cada mapa, a saliência é calculada usando-se o mecanismo de *center-surround*: uma operação que basicamente extrai contrastes fortes do mapa, dando intensidades de pixel altas para essas regiões.

Figura 1: À esquerda, a imagem original. No centro, seu mapa de oponência amarelo-azul. À direita, o resultado de *center-surround* no mapa de cor.

2.2.3 Mapas de saliência para cada instância de *feature*

Para cada *feature* (e.gvermelho) é calculado o *center-surround*. Isso é feito na imagem original e em diversas outras dimensões dela, calculando-se a pirâmide da imagem. Geralmente usamos quatro níveis. Isso é importante para capturar saliências nos mais diversos níveis de detalhe da imagem. Uma vez calculados, todos os mapas de uma certa *feature* são redimensionados para as dimensões originais e somados, formando assim um mapa de *feature*.

2.2.4 Normalização

Uma vez calculados os mapas para cada *feature* (vermelho, orientação horizontal etc), é preciso fazer uma normalização nos mesmos. Isso se deve ao fato que, se há grande frequência de picos de saliência no mapa de vermelho, por exemplo, este não é de muito valor, pois o que se quer identificar são

regiões salientes com relação à imagem como um todo. Assim, para cada mapa é calculado um peso de normalização.

São diversos os critérios desenvolvidos, como: número de máximos locais, densidade de máximos locais, espalhamento espacial dos máximos. Uma análise das alternativas não mostrou muitas diferenças no desempenho, então opta-se pelo método mais simples, por padrão, que é o número de máximos locais. Isso pode ser obtido por limiar *Otsu*, seguido de um algoritmo de componentes conexos para contar os máximos locais.

2.2.5 Mapas de saliência para cada *feature*

Uma combinação hierárquica dos mapas é feita após as normalizações. No final dessa operação, há três mapas: cor, contraste(luminância) e orientação. Eles são formados simplesmente somando e normalizando suas instâncias: O de cor, por exemplo, é formado somando-se os mapas de saliência de vermelho, verde, amarelo e azul.

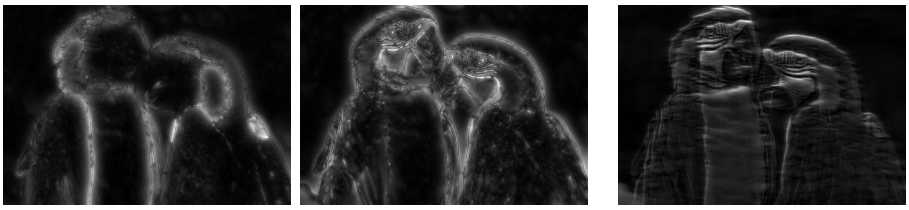

Figura 2: Mapas de saliência da figura original 1. Esquerda: mapa de cor. Centro: mapa de contraste. Direita: mapa de orientação.

2.2.6 Combinação final

É dado um peso para cada mapa de saliência (cor, contraste, orientação) e então eles são somados e normalizados. Nos testes feitos, obteve-se melhores resultados dando peso maior para a cor e menor para a orientação. Nos exemplos aqui, os pesos são 2, 1, 0.1 para cor, contraste e orientação, respectivamente.

Figura 3: Mapas de saliência final para uma figura. À esquerda, a imagem original. À direita, o mapa de saliência final.

2.2.7 Implementação

Todas as etapas do modelo descritas aqui foram implementadas. A linguagem utilizada foi Python, usando-se OpenCV e numpy. O código está disponível em (codigo aqui).

2.2.8 Considerações

O modelo implementado foi testado em diversas imagens e dá resultados satisfatórios na maioria dos casos: em regiões intuitivamente mais salientes, o mapa mostra a região como mais clara (como na figura 3). Há muitos hiperparâmetros para o modelo, como: níveis de pirâmide, valores dos filtros de Gabor, método de normalização, pesos para os mapas. Isso exige buscas exaustivas no espaço de alta dimensionalidade dos hiperparâmetros para achar bons valores, o que é muito custoso. Assim, embora o modelo atual dê bons resultados, um ponto negativo seu é a alta quantidade de hiperparâmetros.

2.2.9 Comparações

Métricas. Comparações com modelos no topo.

2.3 Modelos novos

Deep learning everywhere.

3 Produção Científica

Modelo Att. Notas: Estudo de métricas. Estudo de sistemas com Deep Learning.

4 Próximos passos

DeepFix. Vídeo.