

PUC-RIO / Departamento de Informática
Curso de Realidade Aumentada e Colaborativa (2005.2)
Trabalho 2

Desenvolvedores: Mauricio Ferreira, Diogo Carneiro e Carlos Eduardo Lara

Introdução

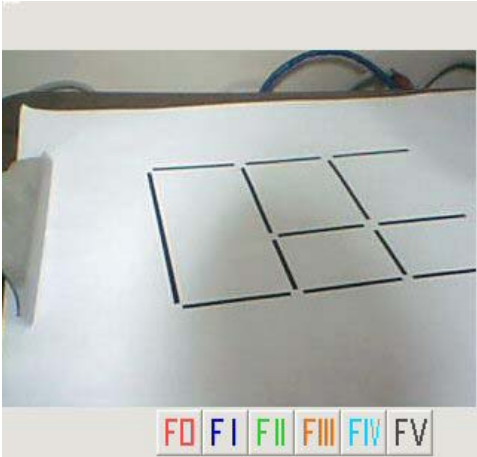
A Realidade Aumentada é um subconjunto da Realidade Virtual que realiza à partir do mundo real uma fusão com objetos virtuais.

O objetivo do projeto foi de implementar a técnica de Realidade Aumentada utilizando os conhecimentos adquiridos nas aulas, ministradas pelo professor Marcelo Gattass, para inserir corretamente objetos virtuais tri-dimensionais em imagens de uma cena real, capturadas por uma WebCam. O projeto foi construído com base nas bibliotecas do "Juiz Virtual" criadas na Tese de doutorado do Flávio Szenberg.

Processos de Desenvolvimento

Conforme dito acima o objetivo é poder desenhar um objeto virtual sob alguma referência do mundo real. Para isso é utilizado um desenho chamado Marcador. O modelo escolhido é baseado em linhas perpendiculares, é através dele que a webcam encontra os pontos de referência onde o software poderá projetar o desenho virtual. Para isso, foram necessários, basicamente, os seguintes passos, que serão detalhados nas próximas seções ao longo deste texto:

1. Capturar o marcador através da Webcam em tempo real

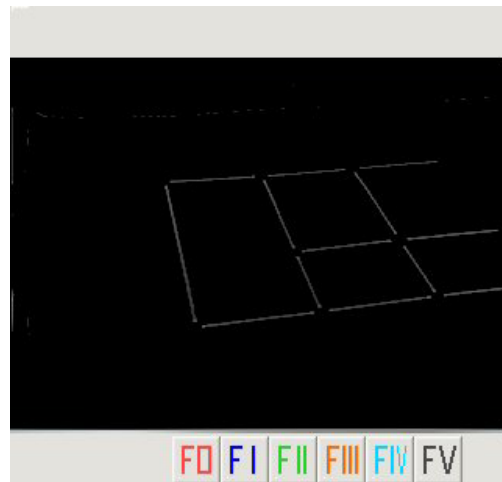
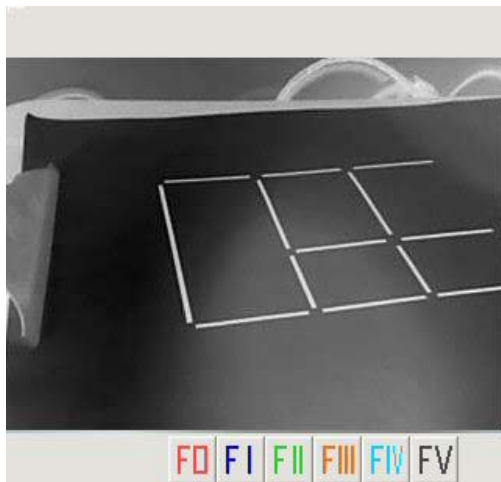


Essa fase é apenas a fase de captura da imagem que contenha o marcador.

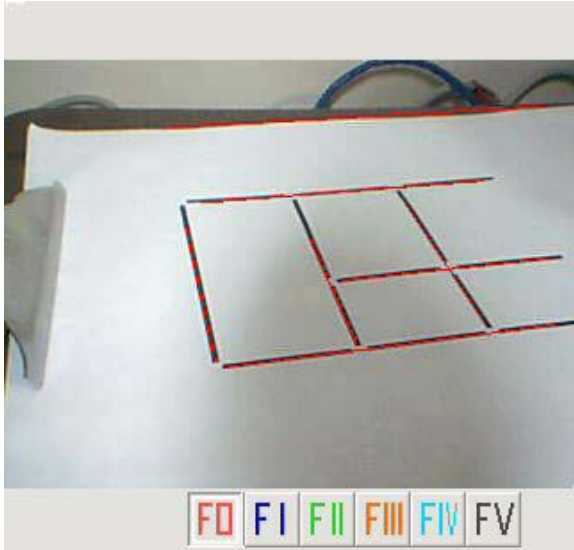
2. Filtragem para realce de linhas

Essa fase tem o objetivo de detectar pontos candidatos a estarem sobre um segmento de reta presente na imagem, é empregado à imagem o filtro Laplaciano da Gaussiana (LoG), baseado na luminância. Este filtro aplica-se a imagens monocromáticas em tons de cinza, portanto foi necessário converter a imagem em negativo através da transformação negativa quando os pontos que desejamos realçar são mais claros que seus vizinhos.

Para exemplificar os resultados do filtro as imagens abaixo representam primeiramente à esquerda a imagem de entrada em tons de cinza negativos (que é a imagem de entrada do filtro) e a imagem à direita o resultado do filtro LoG baseado em uma segmentação (threshold) feita na imagem filtrada. Todo esse processo da segunda etapa faz referência a função `jvLoGFilter` da biblioteca do Juiz Virtual.



3. Extração de segmentos de retas longos



Essa fase tem o objetivo é localizar segmentos de retas longos candidatos a serem linhas da imagem do modelo.

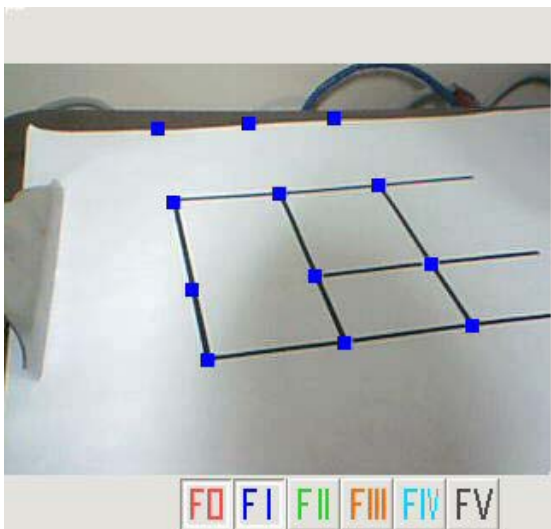
A partir da imagem gerada pelo LoG é possível filtrar melhor os segmentos de reta baseado em uma segmentação (threshold) feita na imagem filtrada. Esse processo de segmentação foi feito através da utilização de um limiar de corte sem um método automático para determinar um valor ótimo para esse corte. Esse valor é determinado em tempo real pelo o usuário.

O procedimento para extrair os segmentos de reta é dividido em dois passos:

1. Eliminação de pontos que não estão sobre nenhum segmento de reta.
2. Determinação de segmentos de retas.

Assim é possível gerar as retas vermelhas encontradas na imagem acima. Todo esse processo da segunda etapa faz referência a função `fvEdgeDetect` da biblioteca do Juiz Virtual.

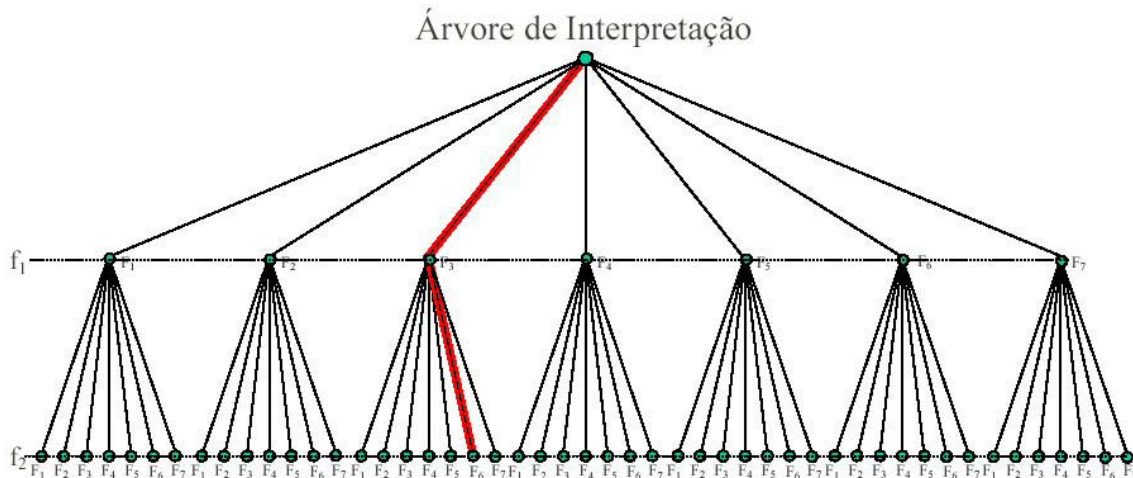
4. Reconhecimento dos segmentos



Agora com alguns segmentos de reta conhecidos na mão é possível encontrar informações sobre a imagem em relação ao marcador. A figura ao lado é uma tentativa de encontrar os vértices do marcador. Os pontos azuis são as interseções dos segmentos de retas encontrados, como foi reconhecido também retas fora do marcador, foram encontrado aqueles três pontos fora do marcador.

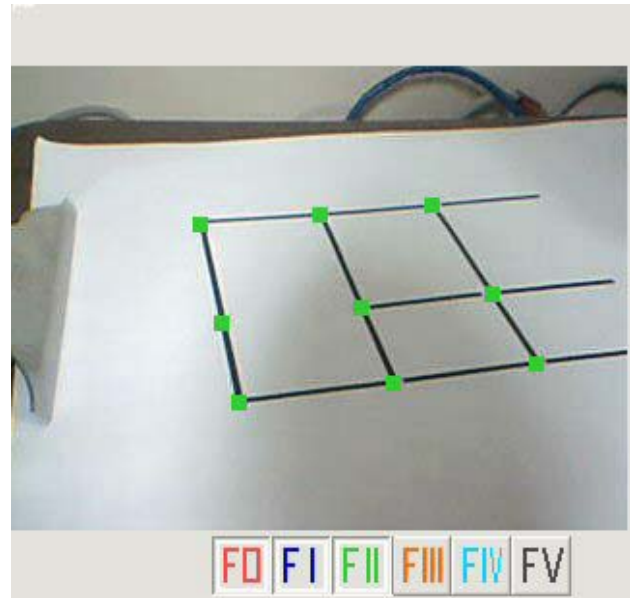
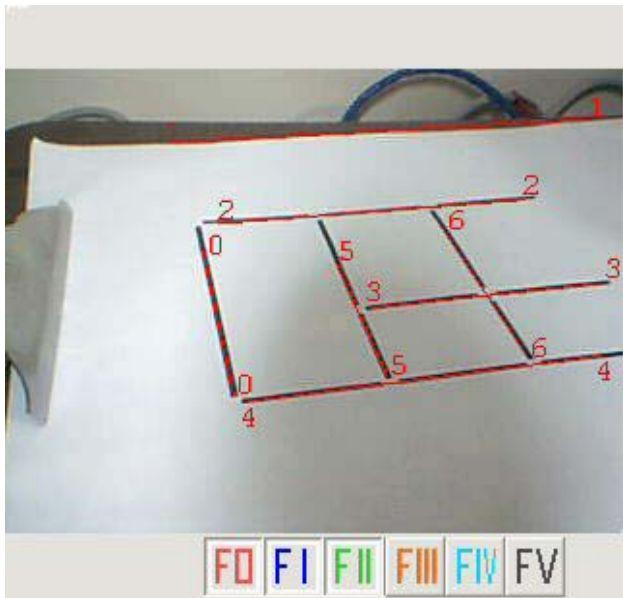
Numa tentativa de encontrar somente os vértices do marcador foi preciso interpretar os segmentos de retas

encontrados e verificar se o dado conjunto de retas se "encaixa" com as do modelo. Para isso foi utilizada a função de reconhecimento de correspondências de linhas baseada numa árvore de interpretação também da biblioteca do Juiz Virtual: `jvInterpret`

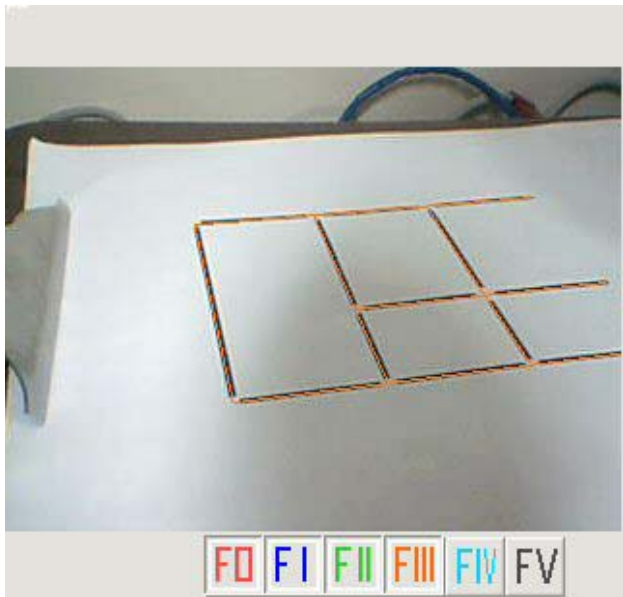


Essa estrutura de árvore chamada de “árvore de interpretação”, na qual cada folha representa uma possível solução. Em cada nó da árvore é feita uma interpretação para verificar a validade da possível solução do reconhecimento. A profundidade da árvore depende do tamanho do modelo – ela é igual ao número de linhas do modelo real. A figura acima ilustra os dois primeiros níveis de uma árvore de interpretação.

Assim é possível identificar as retas que são ou não pertinentes as retas do modelo. Por exemplo, na primeira figura abaixo o interpretador identificou que a reta número 1 não pertence ao modelo e é então descartada. Agora com posse das retas pertencentes ao modelo é possível gerar as interseções das retas e pegar os vértices corretos. E através de uma transformação projetiva planar H (homografia) correspondente às linhas reconhecidas é encontrada (usando pontos de interseção da homografia buscar vértices que não puderam ser capturados pela webCam. O resultado desse processo é a imagem abaixo à direita.



5. Reconstrução do Modelo

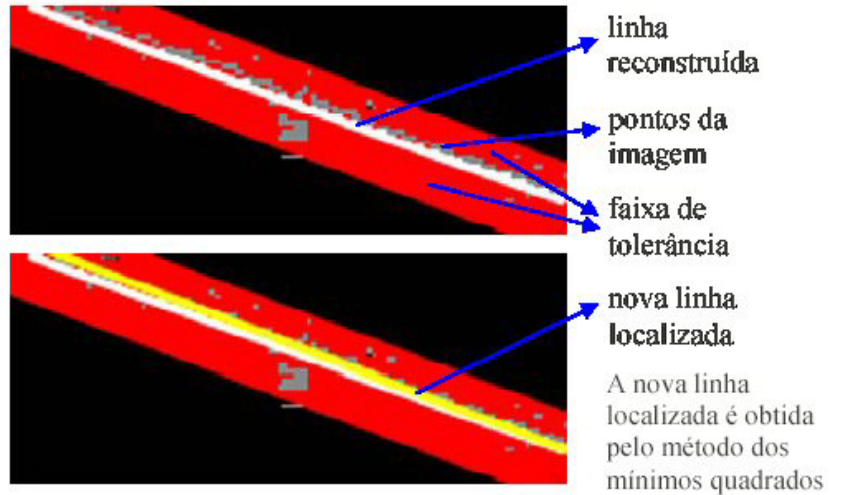
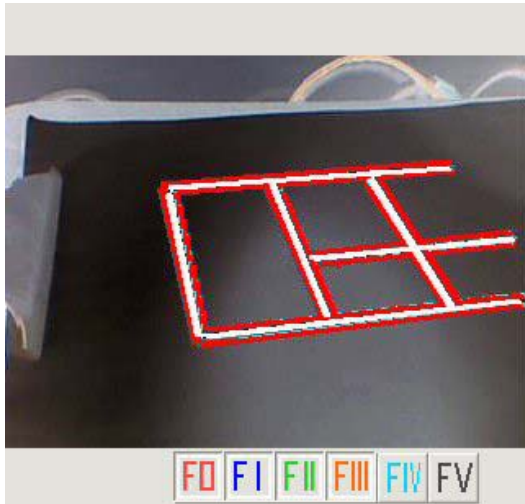


A partir de agora basta apenas reconstruir o modelo com base nos vértices encontrados pela etapa anterior, a figura ao lado ilustra o modelo reconstruído. A partir daí como o algoritmo de árvore de Interpretação é muito "pesado", portanto se ficarmos fazendo essa interpretação a todo frame, não iremos obter uma aplicação em tempo real nas máquinas atuais. Para contornar este problema é feito entre um frame e outro um ajuste de linhas de acordo com as linhas do frame anterior, dada uma faixa de tolerância. Isto resolve o problema quando a câmera está parada, mas o problema permanece se a movimentação da câmera é um pouco

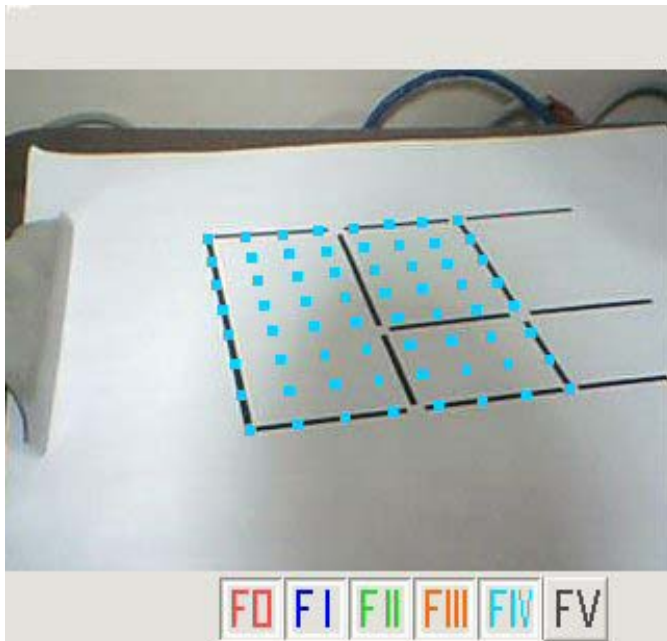
maior. Para resolver este problema faz-se uma predição do movimento segundo os 2 frames anteriores (isto está explicado em maiores detalhes na tese do Szenberg).

A figura abaixo à direita ilustra o esquema para reajuste das linhas na imagem com base nas faixas de tolerância. A linha branca é a reconstruída e a linha amarela é localizada a partir dos pontos cinza internos à faixa de tolerância,

desenhada na cor vermelho. A linha amarela é localizada através do método de mínimos quadrados (é a reta para a qual a soma dos quadrados das distâncias aos pontos considerados é mínima).



6. Calibração da Câmera



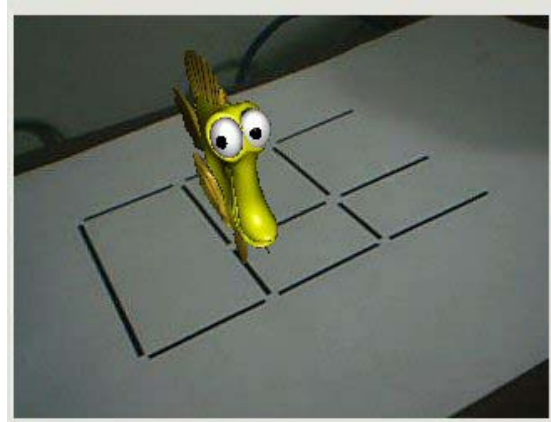
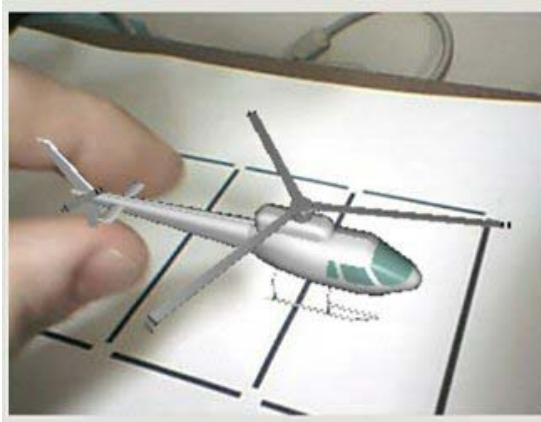
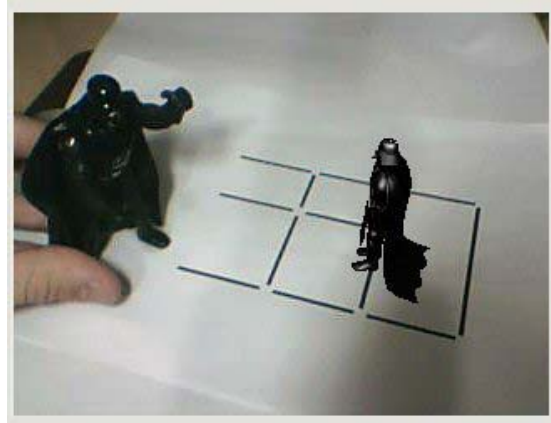
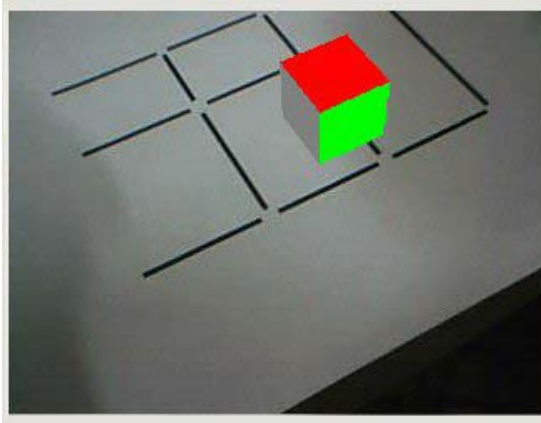
O algoritmo utilizado para a calibração da câmera foi o Tsai 2D. Tal algoritmo precisa de um input de no mínimo 8 pontos de referência. Neste ponto, o software já é capaz de fornecer tais pontos necessários para começar então o processo de calibração da câmera. A figura ao lado representa o resultado do método `tsaiBuildNewData` que reamostra os pares de pontos segundo pares de pontos originais.

A partir daí podemos encontrar a posição da câmera e a distância focal, e com base também nas coordenadas do centro da imagem é possível encontrar a matriz de

projeção e a matriz de modelview para o OpenGL. Agora sim já podemos projetar objetos virtuais sobre a superfície do marcador.

Resultados

As figuras abaixo representam alguns resultados finais encontrados:



Conclusões

O trabalho apresentou resultados satisfatórios, porém o sistema ainda tem muito que evoluir pois apresenta problemas quando a movimentação da câmera ocorre de maneira rápida e também quando ocorre variações grandes na luz ambiente. Essa técnica de Realidade aumentada é superior a de cadeia fechada no sentido da oclusão, pois mesmo com oclusão de algumas retas do marcador, mesmo assim ainda é possível reconhecer o modelo. Resumidamente, estes são os passos executados pelo programa:

- Filtro Laplaciano da Gaussiana (LoG)
- Detecção de segmentos de retas por divisão da imagem em células.
- Interpretação dos segmentos de retas segundo o modelo fornecido.
- Extração dos pontos relevantes do modelo (por interseção das linhas reconhecidas)
- Calibração de câmera (Método Tsai).

Caso o sistema não se perca, apenas ficará executando os seguintes passos:

- Filtro Laplaciano da Gaussiana (LoG)
- Ajuste das linhas de acordo com as 2 homografias anteriores.
- Extração dos pontos relevantes do modelo (por interseção das linhas reconhecidas).
- Calibração de câmera (Método Tsai).

Referências bibliográficas

- E. Trucco, A. Verri, “ Introductory Techniques for 3-D Computer Vision ”, Prentice Hall, 1998.
- Marcelo Gattass, Realidade Aumentada e Cooperativa, PUC – Rio de Janeiro , 2005.2.
- [Tsai86] Tsai, Roger. “An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision” . Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, págs 364-374, 1986.
- [Tsai87] Tsai, Roger. “A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses” . IEEE Journal of Robotics and Automation 3(4): 323-344, Agosto de 1987.
- [Tsai86] Tsai, Roger. “An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision” . Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, págs 364-374, 1986.
- [Szemberg01] Szemberg, Flávio. “ACOMPANHAMENTO DE CENAS COM CALIBRAÇÃO AUTOMÁTICA DE CÂMERAS” . Dezembro de 2001.