

Implementação de jogo da velha em realidade virtual

Cristóvão Bartholo
140135081

Universidade de Brasília (UnB)
cristovao@live.com

Resumo—Esse documento relata o procedimento de implementação do trabalho final da matéria de introdução ao processamento de imagem. Esse trabalho consistiu em realizar a implementação do artigo "Augmented Reality Tic-Tac-Toe" de Joe Maguire e David Saltzman da universidade de Stanford.

Index Terms—IPI, processamento, realidade-virtual, imagem, opencv, jogo-da-velha.

I. INTRODUÇÃO

O objetivo desse trabalho consistiu em aplicar os conhecimentos adquiridos em aula do conteúdo da matéria de Introdução ao Processamento de Imagens.

Este trabalho consiste em implementar um jogo da velha em python utilizando a biblioteca OpenCV.

A. Homografia

Homografia no campo da visão computacional refere-se a ideia de que dois retratos de uma mesma imagem planar estão relacionadas por homografia, assumindo o modelo pinhole. Ou seja, através de um conjunto de operações matriciais pode-se obter uma imagem através de outra.

$$H = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \quad (1)$$

$$H \begin{bmatrix} uw \\ vw \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(2)

Pode-se obter a matriz H resolvendo-se o sistema linear, utilizando três pontos de cada imagem.

B. Características da Imagem

Características em visão computacional refere-se identificação de propriedades regionais de uma imagem. Dentre os diversos métodos de extração de características tem-se o método de detecção de característica invariante à escala (SIFT).

Este método pode ser utilizado, por exemplo, para obter-se os pontos em duas imagens homomórficas e calcular a matriz homomórfica H, como pode ser visto na figura 1.

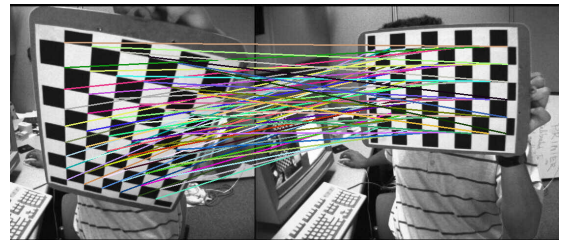


Figura 1. exemplo de extração de características em duas imagens homomórficas.

II. METODOLOGIA

O procedimento do jogo consiste em tirar uma foto de referência perpendicular (fig. 2) à figura básica e a partir disso projetar os devidos símbolos ao longo do jogo.

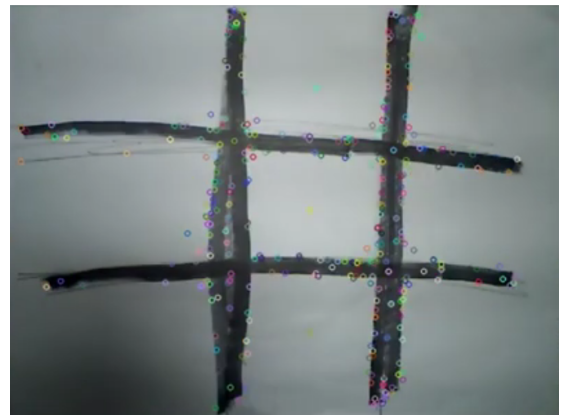


Figura 2. Exemplo de uma imagem de referência usando o SIFT.

A implementação do artigo se deu utilizando a biblioteca OpenCV, com sua versão para python, para a identificação de características da imagem foi utilizado o algoritmo SIFT provido pela biblioteca de visão computacional. Esse algoritmo foi usado para identificar os pontos característicos na imagem de referência e da imagem provida pela câmera ao longo

do jogo, para a partir dos pontos identificados processar um casamento dos pontos para então obter a matriz homomórfica.

A matriz homomórfica inversa então é utilizada para projetar os símbolos desenhados sobre a imagem de referência e transformar para o plano da imagem obtida pela câmera em um dado quadro.

o método utilizado para seleção dos pontos foi o *RANSAC* (RANdom SAmple Consensus) provido pela biblioteca do OpenCV.

O código foi separado em dois arquivos, um contendo a parte de visão computacional e o outro contendo a lógica do jogo da velha, que foi utilizado como uma biblioteca pelo programa principal.

A. Parte de visão computacional

Para testes foi utilizado um papel A4 em que foi desenhado a base do jogo da velha e para facilitar o desenvolvimento, recortes da letra "X" para serem sobrepostos sobre o "tabuleiro".

Para identificação do "X" na câmera o quadro é passado para o plano da imagem de referência utilizando a matriz *H*, então é utilizado o método "matchTemplate" no modo "TM_CCOEFF_NORMED" para identificar a figura na imagem. O template também é rotacionado 7° e 90° anti-horário para calcular o casamento. O template da figura utilizada para a identificação está apresentada na imagem 3.



Figura 3. Imagem do template do X

B. Parte da lógica do jogo

A lógica do jogo consistiu em seguir uma sequência de estratégias priorizadas em ordem decrescentes para garantir sempre uma boa jogada, de modo que não há como o oponente ganhar.

Na primeira etapa a IA tenta finalizar o jogo, caso haja a possibilidade. Seguindo disso, ela verifica se há alguma posição em que o jogador está prestes a finalizar o jogo criando uma reta de "X". caso não haja, ela verifica se a posição do meio está vaga, e se estiver a preencher.

Caso nenhuma das ações acima seja satisfeita, a IA observa o símbolo do meio. Caso o espaço do meio esteja ocupado pelo símbolo do oponente, a IA irá tentar preencher os quadrantes laterais e superiores, caso a casa central esteja ocupada pelo símbolo da IA, ela tentará preencher uma das casas da quina do jogo.

Finalmente, caso nenhum desses eventos ocorra, ela preencherá a primeira casa vazia que estiver disponível a ela.

Então a partir dos pontos obtidos, utiliza-se um threshold de 0.45 para determinar a posição dos pixels da imagem casamentada com o template. Utilizando-se essas posições pode-se determinar qual o quadrante em que o "X" foi inserido

e passar para a lógica do jogo para determinar a próxima jogada.

III. RESULTADOS

Um exemplo do resultado pode ser assistido através do link <https://youtu.be/OyUPyRpV3JU>.

IV. CONCLUSÕES

Pôde-se concluir que os resultados obtidos são satisfatórios, apesar do *frame rate* ser de em torno de dois quadros por segundo, isso pode ser otimizado no código para que essa taxa aumente.

Além disso, pode-se elaborar mais o jogo e permitir que o jogador selecione o símbolo com o qual deseja jogar, dentre outros detalhes que se pode acrescentar.

O método SIFT se mostrou robusto o suficiente para solucionar este problema, apesar de que com o posicionamento dos cartões que obstruíam partes do tabuleiro tornava a projeção ligeiramente mais instável.