Reconstrução Estéreo com Imagens de Alta Definição

Sandro B. N. Lopes, Bruno Marques Ferreira da Silva e Luiz M. G. Gonçalves
Universidade Federal do Rio Grande do Norte
Natal-RN, 59078-970
Email: {sandro, brunomfs, lmarcos}@dca.ufrn.br

***Abstract*—Este trabalho consiste no desenvolvimento de uma técnica para reconstrução tridimensional de cenas a partir de um par de imagens *full HD* (alta definição máxima - 1920x1080 pixels), adquiridas por uma câmera digital estéreo profissional. Para tal, foi desenvolvido um sistema de aquisição de imagens da câmera com o uso de uma placa de captura, e posteriormente, um programa de visão computacional baseado na biblioteca OpenCV que realiza as operações de calibração das câmeras isoladamente, calibração e retificação estéreo, cálculo do mapa de disparidade e geração de um mapa tridimensional. Foi realizada uma análise dos resultados obtidos para dois algoritmos de geração de mapas de disparidade disponibilizados pela biblioteca. Um deles é o método de correspondência de blocos com soma de diferenças absolutas (SAD), e o outro é o método de particionamento de grafos de fluxo (ou *GraphCut*). Além disto, foi feito um estudo sobre os procedimentos de calibração individual das câmeras, calibração e retificação estéreo.**

I. Introdução

O sistema de visão estéreo é o sensor de profundidade mais eficiente e requisitado nos seres humanos. A velocidade como este processo é feito e a qualidade dos dados obtidos permitem que uma pessoa possa reagir rápida e precisamente a algum tipo de evento no meio. E baseado na capacidade informacional dos sistemas de visão biológicos, pesquisas em processamento de imagens e visão computacional têm sido conduzidas para permitir extração de dados tridimensionais por meio de duas (ou mais) imagens digitais com uso de computadores.

O emprego das técnicas de visão estéreo é vasto, abrangendo campos de aplicação como sistemas autônomos de navegação, sistemas de vigilância, fotogrametria e rastreamento de pessoas e objetos [1]. Além disto, pode ser utilizado em projetos de realidade aumentada, como em [2], bem como 3DTV, videoconferência e entretenimento [3]. De acordo com o tipo de aplicação, estes processos podem ser feitos de forma *on-line*, ou seja, o sistema captura e processa os dados obtidos ao mesmo tempo, ou *off-line*, quando o processamento é feito depois da aquisição.

Nos últimos anos, uma nova questão tem surgido na área de visão estéreo. Trata-se do rápido desenvolvimento das tecnologias digitais de alta resolução, motivadas pelo interesse da indústria de telecomunicações em desenvolver sistemas de transmissão áudio-visuais mais rápidos e com maior poder de imersão ao espectador. Neste trabalho, um sistema de visão computacional foi desenvolvido para imagens em alta definição máxima (conhecida pela abreviatura em inglês *full HD*, com 1920x1080 pixels) obtidas de uma câmera digital estéreo profissional, associada a uma placa de captura da Blackmagic, modelo DeckLink HD Extreme 3D+. Foram utilizados algoritmos da biblioteca de visão computacional OpenCV, para a calibração de uma câmera, para a calibração e retificação estéreo, para o cálculo do mapa de disparidade e para a geração do mapa de profundidade propriamente dita.

Algoritmos tradicionalmente utilizados para a geração de mapas de disparidade, como o *GraphCut*, que é um algoritmo de particionamento de grafos de fluxo, e o SAD (do inglês *sum of absolute differences*), que é um método de correspondência em área, foram avaliados. Os resultados obtidos estão relacionados a projetos de pesquisa maiores, como o trabalho de fusão de dados tridimensionais de vários dispositivos e o desenvolvimento de um sistema de visão computacional para a estação de base de um veículo aéreo não-tripulado. Além disto, o programa de acesso a câmera estéreo estará disponível para usos futuros e poderá, inclusive, ser melhorado para abranger outros equipamentos e formatos de vídeo.

Este trabalho está dividido da seguinte forma: na sessão 2, será abordado o embasamento teórico necessário para compreender o problema, que será relatado na sessão 4. Na sessão 3, estão listados alguns trabalhos relacionados. Na sessão 5 é descrita a implementação do programa e na sessão 6, os experimentos realizados. Na última sessão comenta-se os resultados obtidos e os trabalhos futuros.

II. Base teórica

A. Geometria da cena

Na reconstrução tridimensional, o principal interesse é definir a disposição dos objeto com relação a câmera e a referência do mundo. Devido a simplicidade, geralmente utiliza-se o modelo de *perspectiva* ou *pin-hole*. Neste modelo, o **plano da imagem** π é perpendicular ao **eixo óptico** no ponto C, denominado de **ponto principal**. Todos os raios que partem dos objetos da cena e projetam-os no plano da imagem convergem para um único ponto O, denominado de **centro da projeção**. A distância f entre o centro de projeção e o plano da imagem é denomindada de **distância focal**.

O objetivo deste modelo é encontrar uma relação entre o ponto $P = [x_P \ y_P \ z_P]^T$ no ambiente e sua projeção $p = [x_p \ y_p]^T$, descrita pelas equações 1 e 2.

$$x_p = x_P * \frac{f}{z_P} \tag{1}$$

$$y_p = y_P * \frac{f}{z_P} \tag{2}$$

B. Calibração de câmeras

O processo de calibração de uma câmera tem como principal objetivo obter uma correspondência metamática entre os objetos de uma cena e a imagem gerada, por meio de uma transformação. Além disto, também permite extrair coeficientes de correção das principais distorções existentes em uma câmera.

1) Correspondência de pontos: Dado um ponto $P = [x_P\ y_P\ z_P]^T$ no ambiente e seu correspondente no plano da imagem $\hat{p} = [x_{\hat{p}}\ y_{\hat{p}}]^T$ em pixels (ou seja, em coordenadas de imagem), a transformação que relaciona os dois pontos pode ser descrita pela equação 3:

$$\begin{bmatrix} \hat{p} \\ \cdots \\ 1 \end{bmatrix} = \frac{1}{z_p} \underbrace{\begin{bmatrix} -f_x & 0 & x_c \\ 0 & -f_y & y_c \\ 0 & 0 & 1 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} R & \vdots & T \end{bmatrix}}_{W} \begin{bmatrix} P \\ \cdots \\ 1 \end{bmatrix} \tag{3}$$

Onde R, a matriz de rotação e T, o vetor de translação, são os parâmetros extrinsecos da câmera. Os valores f_x e f_y, a distância focal medido em número de pixel nas coordenadas x e y, respectivamente, x_c e y_c, as coordenadas do ponto principal da imagem em pixels, e $\frac{1}{z_p}$, geralmente descrito como um fator de escalonamento s, são os parâmetros intrínsecos da câmera.

2) Distorções na câmera: Existem vários tipos de distorções descritas na literatura. O algoritmo de calibração da OpenCV trata duas delas: as distorções **radiais** ou **geométricas**, que causam o efeito 'barril' ou 'olho-de-peixe' sobre a imagem - as linhas mais afastadas do centro sofrem curvatura e as aresta sofrem atenuação , e as **tangenciais**, que causam a inclinação de linhas originalmente paralelas em determinada direção, dando a sensação de escalonamento maior em um dos lados.

A distorção radial ou geométrica pode ser modelada, aproximadamente, por meio das equações 4 e 5, enquanto que a distorção tangencial pode ser descrita peals equações 6 e 7.

$$x_{cor} = x_{ori}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{4}$$

$$y_{cor} = y_{ori}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{5}$$

$$x_{cor} = x_{ori} + [2p_1 y_{ori} + p_2(r^2 + 2x_{ori}^2)] \tag{6}$$

$$y_{cor} = y_{ori} + [p_1(r^2 + 2y_{ori}^2) + 2p_2 x_{ori}] \tag{7}$$

Os valores de k_1, k_2 e k_3 são os coeficientes de distorção radial e p_1 e p_2 são os coeficientes de distorção tangencial, relativos a transformação do pixel original $p = [x_{ori}\ y_{ori}]$ no correspondente corrigido $p_{cor} = [x_{cor}\ y_{cor}]$. O valor de r informa o raio de curvatura da lente.

C. Estereometria

O modelo de estereometria é baseado no processo de triangularização de pontos. Nele, assume-se que os planos de imagens são coplanares, apresentam os eixos ópticos paralelos entre si e estão perfeitamente alinhadas horizontalmente (ou seja, suas linhas são contínuas entre si). Além disto, os centros das duas câmeras devem coincidir, isto é, devem estar localizados no mesmo pixel em ambas as imagens. Com isto, a distância z_P entre o ponto P no ambiente e o eixo T, que liga os centros ópticos das câmeras, denominado de *linha de base*, pode ser definida através da equação 8, onde x_pE e x_pD são os valores da coordenada x para o ponto P na imagem esquerda e direita, respectivamente, e f é a distâcia focal, assumida igual para as duas câmeras:

$$z_P = \frac{fT}{x_pE - x_pD} \tag{8}$$

A partir dos parâmetros extrínsecos de cada câmera, obtidos na calibração individual (aqui descritas por R_E e T_E para a câmera esquerda, e R_D e T_D para a câmera direita), é possível estabelecer a matriz de rotação R_s e o vetor de translação $T_s = [x_{T_s}\ y_{T_s}\ z_{T_s}]$ que relacionam as projeções em coordenadas de imagem p_E e p_D:

$$R_s = R_D(R_E)^T \tag{9}$$

$$T_s = T_D - R_s^T T_E \tag{10}$$

Como a maioria das geometrias estéreo não obedecem os pré-requisitos estabelecidos para triangulação de pontos, é necessária a realização de um processo de calibração e retificação das imagens, feito com base no modelo geométrico **epipolar**. De acordo com este modelo, a relação entre a projeção de P no plano de imagem de uma das câmeras com a linha epipolar correspondente no plano da outra câmera é feita por meio de uma matriz 3x3 de posto 2, denominada de **matriz essencial**:

$$E = R_s S\ ,\ \ onde\ S = \begin{bmatrix} 0 & -z_{T_s} & y_{T_s} \\ z_{T_s} & 0 & -x_{T_s} \\ -y_{T_s} & x_{T_s} & 0 \end{bmatrix} \tag{11}$$

Usualmente, estebelece-se a relação entra as projeções da câmera esquerda e direita sob coordenadas de imagem. Neste caso, define-se uma nova matriz de correspondência denominada de **matriz fundamental**, e descrita como:

$$F = (M_D^{-1})^T E M_E^{-1} \tag{12}$$

Onde M_D e M_E são as matrizes de parâmetros intrísecos da câmera direita e esquerda, respectivamente.

III. Trabalhos relacionados

A quantidade de trabalhos envolvidos com reconstrução tridimensional é vasta e multidisciplinar. Para relacionar este trabalho com a literatura existente, buscou-se por aplicações de reconstrução tridimensional que utilizem imagens *full HD* ou baseadas no modelo binocular estéreo desenvolvidas a partir de 2009, para encontrar projetos que desenvolveram algum sistema de visão com aparelhos semelhantes aos utilizados neste trabalho.

O trabalho de [4] constituiu-se no desenvolvimento um *pipeline* de operação sobre uma plataforma híbrida, para a construção de mapas de disparidade em tempo real. Nele,

foi utilizado uma CPU, uma GPU NVIDIA GTX480 e um FPGA modelo Stratix III EP3SL340 com duas memórias RAM DDR2 e interface PCIe 8x. O método de geração dos mapas de disparidade utilizado foi o *block matching* com ZSAD (soma das diferenças absolutas com média zero). As imagens de teste foram obtidas de uma câmera de vídeo na resolução de 1920×1080 pixels, a 30 quadros por segundo em modo progressivo.

No trabalho de [2], foi desenvolvido um sistema embarcado de captura de imagens e extração de mapas de disparidade para realidade aumentada, constituído de uma placa de visão modelo GigE com uma placa FPGA Altera Stratix e um *sistem-on-chip* modelo Acadia II, que substitui um sistema anterior composto de um processador Intel Core i7 a 2,2GHz, memória de 4GB modelo DDR3-1333 e uma GPU NVIDIA GTX 285M. O processo de geração do mapa de disparidade foi feito por meio de *block matching* em níveis de resolução diferentes, para gerar uma pirâmide Gaussiana de níveis.

No trabalho de [5] foi desenvolvido uma sistema em ASIC que permite ao usuário manipular a percepção de profundidade de um fluxo de 3DTV em alta definição, gerando novas imagens. A plataforma utilizada na implementação foi um processador TSMC 65nm 1P8C. O tamanho total do circuito é de 6,9mm $\times$ 6,9 mm e o consumo foi de 1,2W. O algoritmo utilizado para a geração dos mapas de disparidade foi o *block matching* com SAD, com variação das janelas de busca de forma adaptativa, de acordo com textura das imagens, sendo realizada uma análise de oclusões por meio de um algoritmo de verificação cruzada. As imagens foram extraídas a uma taxa de 60 quadros por segundo.

No trabalho de [6], foram utilizadas imagens em UHD (abreviação de *ultra high definition* - ultra alta definição) para a reconstrução tridimensional de cenas, por meio de visão estéreo para múltiplas imagens utilizando um conjunto de descritores. Objetiva-se neste trabalho reduzir o tempo e o consumo computacional do sistema, ao mesmo tempo que se busca obter melhores nuvens de pontos. O algoritmo desenvolvido foi executado em uma máquina Intel Xeon 2,5GHz Quad Core, com 12GB de memória, e os resultados foram comparados com algoritmos de correlação cruzada normalizada desenvolvidos.

A partir dos trabalhos citados, é possível observar dois diferenciais neste projeto. Primeiramente, o uso de uma câmera digital estéreo profissional para a aquisição de imagens, que garante a geração de um par de imagens em alta resolução aproximada do modelo admitido para a triangulação de pontos e em tempo real. O segundo caso é o uso da técnica de *GraphCut* em imagens de alta resolução que, devido à requisição de processamento e tempo, torna-se muito custoso para a uso, principalmente em tempo real, mas que pode ser uma ótima opção para aplicações que requisitem precisão. A tabela I mostra a classifcação dos trabalhos mencionados.

Na tabela I, para cada referência é estabelecida a sua resolução de trabalho, em pixels (campo *Resolução*), a técnica utilizada para a obtenção do mapa de disparidade (campo *disparidade*), onde a sigla **BM** indica técnicas de *block matching*,

Tabela I
CLASSIFICAÇÃO DOS TRABALHOS RELACIONADOS.

Referência	Resolução	Disparidade	Processamento
[4]	1920x1080	**BM**	hardware
[2]	1920x1080	**BM**	hardware
[5]	1920x1080	**BM**	software
[6]	3072x2048	**MR**	software
Este artigo	1920x1080	**BM**,**GC**	software

MR indica técnicas de multi-imagens e **GC** indica técnica de *GraphCut*, e o modo de processamento do sistema de geração do mapa de diparidade, se é feito diretamente em hardware ou em software (campo *processamento*).

IV. PROBLEMA

O problema abordado por este trabalho pode ser descrito como o desenvolvimento de um *software* para reconstrução tridimensional de ambientes utilizando um par de imagens obtidos de uma câmera estéreo de alta-definição (1920x1080 pixels) e a análise dos resultados obtidos com dois tipos diferentes de algoritmos para a geração de mapas de disparidade, que é parte importante do processo.

Os dois algoritmos analisados foram o *GraphCut*, um algorimo baseado em especialização de grafos, e o segundo é o algoritmo SAD (abreviação do inglês *sum of absolute differences* - soma de diferenças absolutas), que é baseado no modelo de partição de imagens por blocos.

A. Método de calibração individual das câmeras

O modelo de calibração utilizado é baseado no metodo de Zhang, realizado por meio de um tabuleiro de xadrez, que é um objeto plano, fotografado sob várias poses. Para uma imagem do padrão, objetiva-se estabelecer uma correspondência entre dois planos, processo denominado de **homografia**. Neste caso, assume-se que $z_P = 0$, eliminando o vetor r_3 da matriz de rotação e gerando a equação 13:

$$\tilde{p} = s \underbrace{M \begin{bmatrix} r_1 & r_2 & T \end{bmatrix}}_{H} \tilde{P} \tag{13}$$

Assumindo $H = [h_1\ h_2\ h_3]$, e aplicando as restrições de ortonormalidade da matriz de rotação, obtem-se o conjunto de restrições estabeleceido em 14:

$$\begin{cases} h_1^T M^{-T} M^{-1} h_2 = 0 \\ h_1^T M^{-T} M^{-1} h_1 = h_2^T M^{-T} M^{-1} h_2 \end{cases} \tag{14}$$

Fazendo $B = M^{-T} M^{-1}$, é possível concluir que:

$$B = \begin{bmatrix} \frac{1}{{f_x}^2} & 0 & \frac{-x_c}{{f_x}^2} \\ 0 & \frac{1}{{f_y}^2} & \frac{-y_c}{{f_y}^2} \\ \frac{-x_c}{{f_x}^2} & \frac{-y_c}{{f_y}^2} & \frac{-x_c}{{f_x}^2} + \frac{-y_c}{{f_y}^2} + 1 \end{bmatrix} \tag{15}$$

O conjunto de restrições definidos em 14 pode ser reescrito sob a forma de $v_{ij}^T b$, onde:

$$b = \begin{bmatrix} b_{11} \\ b_{12} \\ b_{22} \\ b_{13} \\ b_{23} \\ b_{33} \end{bmatrix} \quad e \quad v_{ij} = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{j1}h_{i2} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i2}h_{j3} + h_{i3}h_{j2} \\ h_{i3}h_{j3} \end{bmatrix} \tag{16}$$

Da seguinte forma:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - V_{22})^T \end{bmatrix} b = 0 \tag{17}$$

Dado um conjunto de n imagens do padrão em poses diferentes, é possivel extender a equação acima para uma equação linear do tipo $Vb = 0$, onde V é uma matriz $2n\,x\,6$. Para valores de $n \geq 2$, o vetor b pode ser definido. Por ele, é possível definir os parâmetros de calibração através do conjunto de fórmulas descrito em 18:

$$\begin{aligned} y_c &= \frac{b_{12}b_{13} - b_{11}b_{23}}{b_{11}b_{22} - b_{12}^2} \\ f_x &= \sqrt{\frac{\gamma}{b_{11}}} \\ fy &= \frac{\gamma b_{11}}{b_{11}b_{22} - b_{12}^2} \\ x_c &= -\frac{b_{13}f_x^2}{\gamma} \\ r_1 &= \frac{1}{2}M^{-1}h_1 \\ r_2 &= \frac{1}{2}M^{-1}h_2 \\ r_3 &= r_1 \times r_2 \\ T &= \frac{1}{2}M^{-1}h_3 \end{aligned} \tag{18}$$

Onde:

$$\gamma = b_{33} - \frac{b_{13}^2 + y_c(b_{12}b_{13} - b_{11}b_{23})}{b_{11}} \tag{19}$$

O valor do fator de escala $s = \|M^{-1}h_1\|$ é extraído da condição de ortonormalidade da matriz de rotação.

B. Retificação estéreo

O modelo de retifição estéreo utilizado é baseado no método desenvolvido por Bouguet, e parte do princípio que a calibração individual das câmeras foi realizada previamente. Além disto, assume-se que o ponto principal na imagem é a origem do sistema de coordenadas da imagem e que as distâncias focais são iguais a f. Dada a matriz de rotação R_s e vetor de translação $T_s = [x_{T_s}\ y_{T_s}\ z_{T_s}]$ que relacionam o sistema de coordenadas das duas câmeras, inicialmente assume-se que existem duas matrizes $\hat{R}_E$ e $\hat{R}_D$, para os planos de imagem da esquerda e da direita respectivamente, que realizam metade do movimento de rotação de R_s.

A matriz R_{rect}, que realiza o alinhamento dos epipolos da câmera esquerda horizontalmente e apontá-los para o infinito, é definida em 20:

$$R_{rect} = \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix} \tag{20}$$

Onde:

$$\begin{cases} e_1 = \dfrac{T_s}{\|T_s\|} \\ e_2 = \dfrac{1}{\sqrt{x_{T_s}^2 + y_{T_s}^2}} \begin{bmatrix} -y_{T_s} & x_{T_s} & 0 \end{bmatrix}^T \\ e_3 = e_1 \times e_2 \end{cases} \tag{21}$$

As matrizes R_{sE} e R_{sD} que realizam o alinhamento coplanar e horizontal dos planos de imagem das câmeras podem então ser definidas por:

$$R_{sE} = R_{rect}\hat{R}_E \tag{22}$$

$$R_{sD} = R_{rect}\hat{R}_D \tag{23}$$

Os parâmetros intrínsecos de cada câmera passam a ser descritos por meio de matrizes de projeção V_E e V_D, dadas pelo conjunto de equações 24, onde x_{C_E} e y_{C_E} são as coordenadas no plano da imagem, em x e y, do centro da imagem esquerda:

$$V_E = \begin{bmatrix} f & 0 & x_{C_E} & 0 \\ 0 & f & y_{C_E} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{24}$$

$$V_D = \begin{bmatrix} f & 0 & x_{C_E} & x_{T_s}f \\ 0 & f & y_{C_E} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{25}$$

Desta forma, um ponto tridimensional passa a ser mapeado em um ponto bidimensional na imagem em coordenadas homogênes. A matriz Q que realiza a projeção inversa, de um ponto bidimensional na imagem em um ponto tridimensional, é dada por:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -x_{C_E} \\ 0 & 1 & 0 & -y_{C_E} \\ 0 & 0 & 0 & f \\ 0 & 0 & -\left(\dfrac{1}{x_{T_s}}\right) & \dfrac{(x_{C_E} - x_{C_D})}{x_{T_s}} \end{bmatrix} \tag{26}$$

C. Método GraphCut - Correspondência global com particionamento de grafos

A partição de grafos de fluxo (ou *GraphCut*) é um método geralmente utilizado para rotulação de pixels em uma imagem, cujo objetivo é minimizar uma função custo de energia estabelecida. No cálculo de disparidade, seu uso permite estabelecer a melhor associação entre os pixels das duas imagens, através da minimização de uma função de fluxo.

Inicialmente é estabelecido um grafo completo, cujos vértices são os pixels (ou elementos) de cada imagem e as arestas são definidos entre cada par do conjunto de vértices

disponíveis, nos dois sentidos (de uma imagem para outra). As ligações entre as imagens são representadas pelo produto interno de ambas, e as ligações entre pixels de uma mesma imagem pelos seu auto-produto (produto interno de um vetor com ele mesmo). A ponderação das arestas do grafo gerado são então estabelecidas de forma a penalizar (ou seja, atribuir maior valor) às conexões cujo par apresente grande diferença de intensidade.

O objetivo é encontrar um subgrafo que indique a melhor semelhança entre cada pixel destas imagens, estabelecendo a correspondência entre todos os elementos (desde que haja correspondência), atendidas as restrições de vizinhança (melhores ligações entre pixels da mesma imagem). O subgrafo obtido é denominado **clique maximal** ou **conjunto maximal conectado**.

D. Método SAD - Correspondência em blocos com soma de diferenças absolutas

O objetivo deste método é estabelecer, de forma local, a disparidade $d(x_p, y_p)$ para cada pixel (x_p, y_p) referente a uma das imagens. Apresenta maior simplicidade de implementação e custo computacional baixo, inclusive servindo para aplicações em tempo real; no entanto, os mapas de disparidade gerados são pouco precisos.

Neste processo, primeiramente é estabelecida uma medida da dissimilaridade ou **custo de correspondência** entre os pixels correspondentes nas duas imagens e que será diretamente proporcional a diferença de intensidade entre os valores dos pixels analisados. Esta função é identificada por $C(x_p, y_p, d)$. Estabelece-se, então, uma janela $W(x, y)$, de tamanho $m \times n$ e centrada no pixel (x, y) de interesse, em uma das imagens. Posteriormente, define-se uma região de busca na outra imagem onde o ponto de interesse possui maior probabilidade de se localizar. Dentro de W, é calculada a função de dissimilaridade correspondente que, neste caso, é dado por:

$$C(x_p, y_p, d) = \sum_{u,v \,\in\, W(x,y)} |I_E(u, v) - I_D(u - d, v)| \quad (27)$$

Onde os valores de $I_E(x, y)$ e $I_D(x, y)$ correspondem aos valores de intensidades luminosas no pixel (x, y) para as imagens esquerda e direita, respectivamente. A disparidade relativa a um determinado pixel é obtida através de minimizaçao da função de dissimilaridade, de forma que:

$$d(x_p, y_p) = arg\ min_k\ C(x_p, y_p, k) \quad (28)$$

V. Implementação

Para a resolução do problema proposto, foi desenvolvido um *software* que realiza a captura de imagens da câmera para um computador, por meio de uma placa de captura, e posteriormente realiza a calibração e o processamento estéreo para a obtenção de mapas tridimensionais, fazendo uso da biblioteca de visão computacional OpenCV.

O diagrama 1 descreve a arquitetura da implementação.

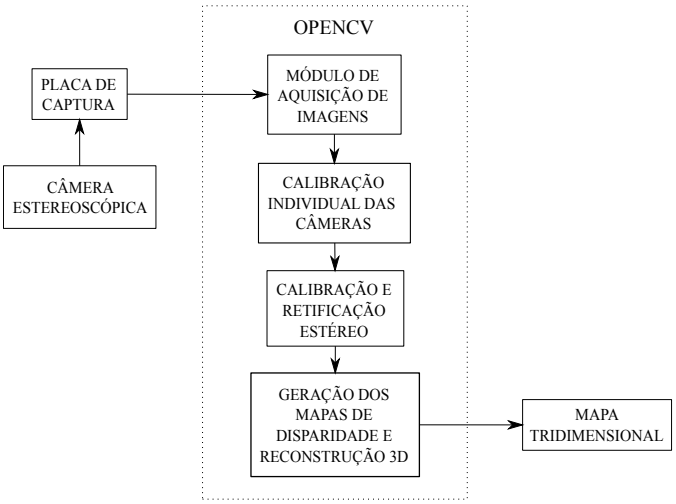


Fig. 1. Diagrama de arquitetura do *software* implementado.

A. Câmera estéreo

A diferença mais notória entre uma câmera usual e uma câmera estéreo é, esencialmente, a quantidade de matrizes de fotossensores presentes no equipamento; enquanto que no primeiro caso existe apenas uma matriz, no segundo caso há um par destes. Isto está ligado ao objetivo principal de uma câmera estéreo, que é a capacidade de gerar vídeos em modo tridimensional.

No trabalho, foi utilizada uma câmera estéreo modelo AG-3DA1P, da Panasonic. Este modelo caracteriza-se por capturar imagens em formatos de 1920x1080 e 1080x720 pixels, com taxa de transmissão de 50 ou 60 fps (quadros por segundo). Permite conexão HDMI ou SDI (dois canais) com um dispositivo de saída, além de um par de entradas para a inserção de cartões de memória SD-card, conexão para captura de áudio diretamente do dispositivo. A figura 2, disponível em [7] mostra o equipamento.

Fig. 2. Câmera estéreo utilizada no trabalho.

B. Placa de captura

Uma placa de captura é um dispositivo que permite conectar um ou mais dispositivos de vídeo e áudio - como câmeras, monitores, microfones e aparelhos de som - a um computador (ou qualquer outro elemento de processamento especializado, como um *video-game*), por meio de uma tecnologia de transmissão específica. Este tipo de equipamento pode ser conectado ao computador de forma externa, por meio de um cabo, ou internamente, através de uma conexão direta com a placa-mãe.

A placa de captura utilizada foi a DeckLink HD Extreme 3D, desenvolvida pela Blackmagic, e que apresenta como principais características a transmissão de dados via HDMI (incluído o suporte à versão 1.4 da tecnologia), além de permitir transmissão via SDI a 3 Gb/s e a transmissão analógica tradicional, via *Component Video*. Além disto, apresenta suporte a transmissão de formatos analógicos de vídeo, como PAL e NTSC, e a formatos digitais de 1080x720, 1920x1080 e 2440x1550 pixels. Possui drivers para Windows, Linux e MacOS, e pode ser utilizado junto com outros programas de edição de vídeo conhecidos, além de possuir um programa próprio para gerenciamento dos dados obtidos - o **Media Express** - e uma API para programação.

C. OpenCV

A OpenCV é uma biblioteca para a construção e manipulação de algoritmos de visão computacional e processamento de imagens desenvolvida pela Intel e atualmente administrada pela WillowGarage. Possui mais de quinhentos algoritmos para tratamento de imagens, segmentação, extração de contornos, reconhecimento e rastreamento de padrões e estereovisão, implementados em C, C++, Python e Java, além de apresentar versões para os sistemas operacionais Windows, Linux, MacOS, Solaris, Android e IOS. Apresenta suporte a manipulação de imagens e vídeos em arquivo no disco ou diretamente de um dispositivo de vídeo, e é desenvolvida sobre licença BSD, que permite tanto uso acadêmico quanto comercial.

A versão utilizada neste projeto é a 2.4.4, compilada e executada no sistema operacional Ubuntu 12.04.

D. Módulo de interface câmera-computador

Apesar da OpenCV possuir suporte para captura de imagens por meio de uma câmera, não foi possível obter diretamente os pares de imagem da câmera estéreo por meio da sintaxe nativa da biblioteca. Além disto, o computador utilizado para o processamento não apresentava nenhuma conexão que suportasse os dados transmitidos pela câmera. Desta forma, se tornou necessário a utilização da placa de captura para realizar a leitura de um par de imagem em tempo real, e o desenvolvimento de um módulo que fosse capaz de converter os dados obtidos pela placa de captura para um formato que a OpenCV pudesse reconhecer.

A placa de captura utilizada possui uma API própria para manipulação dos dados obtidos, cuja versão utilizada foi a 9.7, e que possui um conjunto de interfaces, ou classes abstratas, para realizar os diversos procedimentos sobre os dados obtidos pela placa. O módulo desenvolvido limita as funcionalidades àquelas compatíveis com as próprias características da câmera envolvida e os objetos de estudo do trabalho, que são as imagens de alta resolução.

E. Calibração e retificação - individual e estéreo - das câmeras

Como a calibração de uma câmera é baseada em um objeto padrão de referência, que pode ser bi ou tridimensional, é necessário que ele possa ser reconhecido na imagem. No projeto, o padrão utilizado foi um tabuleiro de xadrez, cujo uso é facilitado pela OpenCV, já que possui um método que permite extrair as coordenadas das quinas internas existentes na imagem. Em um tabuleiro de xadrez com m linhas e n colunas de células, o número de quinas internas em uma linha é $m-1$, enquanto que em uma coluna é $n-1$.Com os valores do padrão detectados na imagem, o próximo passo é definir quais as coordenadas no ambiente dos pontos detectados na imagem, definidos de forma arbritária neste projeto.

Posteriormente, realiza-se o processo de calibração e retificação das imagens individualmente. Os valores obtidos para os valores extrínsecos (as matrizes de rotação e translação da câmera) e intrísecos (a matriz da câmera e os coeficientes de distorção) para cada câmera do equipamento estéreo são armazenados nos arquivos *Calibração esquerda.yml* e *Calibração direita.yml*, com o objetivo de disponibilizá-los para aplicações futuras. As imagens retificadas obtidas são armazenadas nos arquivos de imagem *Imagem_direita-corrigida.png* e *Imagem_esquerda-corrigida.png*.

No processo de calibração estéreo, que visa obter parâmetros que relacionam as duas câmeras, as matrizes de translaçao e rotação entre as câmeras e as matrizes fundamental e essencial obtidas no processo de calibração estéreo são disponibilizados no arquivo *Parâmetros estereoscópicos.yml*, enquanto que os novos valores obtidos para as matrizes de rotação e de projeção das duas câmera e a matriz de mapeamento da disparidade em profundidade, resultantes da retificação estéreo, são armazenados no arquivo *Parâmetros retificação estéreo.yml*.

F. Construção dos mapas de disparidade e reconstrução tridimensional

Foram utilizados dois algoritmos distintos para a geraçao dos mapas de disparidade. Um implementa o método *GraphCut* e o outro, o SAD. Os mapas de disparidade obtidos são salvos em um arquivo plano e uma imagem. Para o método SAD, os arquivos são denominados *Disparidade.yml* e *Disparidade.png*, enquanto que para o *GraphCut*, os dados estão nos arquivos *Disparidade esquerda.yml* e *Disparidade direita.yml*, e as imagens foram disponibilizadas no arquivos *Disparidade esquerda.png* e *Disparidade direita.png*. Com relação ao método *GraphCut*, também são armazenadas imagens do valor absoluto das disparidade, já que os valores obtidos para o referêncial esquerdo são negativos. Esta ima-

gens são *Disparidade esquerda realcada.png* e *Disparidade direita realcada.png*.

Com as disparidades já disponibilizadas, parte-se para a última parte do processo, que é a reconstrução propriamente dita. O resultado da operação de reconstrução tridimensional é armazenado em uma matriz de valores em ponto flutuante de 16 bits e salvo no arquivo *Mapa3D.png*.

VI. Experimentos e resultados

Para a análise dos resultados, foram utilizadas imagens estáticas obtidas diretamente da câmera em 1920×1080 pixels em modo progressivo a uma taxa de 59,94 quadros por segundo. O algoritmo desenvolvido para este trabalho realiza a calibração individual e estéreo das câmeras, mas a medição do mapa de disparidade foi realizada com as imagens originais, e não com as imagens retificadas. Isto ocorre porque as imagens retificadas apresentavam um alto grau de distorção, prejudicando a identificação dos objetos na cena. Além disto, a calibração das câmeras foi feita com apenas uma imagem do padrão de xadrez, que é inserido dentro da cena a ser retificada.

A execução deste programa foi feita em um computador Intel Xeon CPU 5405 2,0GHz x4, memória de 5,9Gb e sistema operacional Ubuntu 12.04, e os experimentos foram realizados com três conjuntos de imagens. Os mapas de disparidade obtidos são mostrados nas figuras 5 e 6.

VII. Conclusão

Neste trabalho foi desenvolvido uma aplicação para realizar a reconstrução estéreo a partir de um conjunto de imagens obtidas de uma câmera digital estéreo. Junto com a reconstrução, também houve a necessidade de se construir um módulo para a aquisição de imagens da câmera. Além disto, observou-se os aspectos computacionais do processo em geral, como a calibração individual e estéreo das câmeras, além da retificação estéreo.

A primeira observação dos resultados obtidos é a notória deficiência que os dois algoritmos de mapa de disparidade apresentaram, mesmo assumindo que as imagens *full HD* apresentem maior detalhamento da cena que as imagens convencionalmente utilizadas. No entanto, é importante observar que estes resultados não podem ser conclusivos, visto que o mapa de disparidade foi gerado com o par original de imagens, e não com o conjunto retificado. Além disto, o método de calibração utilizou apenas uma imagem na operação (a própria imagem a ser analisada), o que tornam imprecisos os resultados obtidos para os parâmetros extrínsecos e intrísecos de cada câmera.

Com relação ao desempenho dos algoritmos, é possível perceber a melhor qualidade dos mapas de disparidade gerados pelo algoritmo *GraphCut* com relação ao algoritmo SAD. No entanto, a análise de tempo de execução mostrou que o *GraphCut* apresentou tempo excessivamente alto, tornando seu uso proibitivo para aplicações em tempo real.

Como trabalhos futuros, espera-se aprimorar o método de calibração individual e estéreo, para permitir que os parâmetros obtidos apresentem valores mais precisos e, consequentemente, possam realizar uma reconstrução mais aprimorada e resultados mais confiáves, bem como desenvolver um módulo de calibração em tempo real para permitir o programa possa ser adaptável as possíveis alterações dos parâmetros da câmera, como a distância focal e a posíção da câmera com relação aos objetos. Também espera-se desenvolver um método de avaliação de desempenho mais robusto e obter um conjunto de imagens de teste mais abrangente, que permita realizar uma avaliação mais detalhada dos algoritmos, por efeitos como textura e iluminação. Por fim, espera-se estender este estudo a outras técnicas, visando futuros aprimoramentos.

Agradecimentos

O autor agradece ao professor doutor Luiz Marcos Garcia Gonçalves pela orientação, ao professor doutor Rafael Vidal Aroca e ao mestre Bruno Marques Ferreira da Silva pelas importantes contribuições, ao grupo de pesquisadores do laboratório NatalNet, especialmente o grupo de trabalho do projeto N-VANT, e a CAPES e o CNPq, pelo apoio financeiro.

References

[1] P. Corsonello, P. Zicari, S. Perri, and G. Cocorullo, "Low-cost fpga stereo vision system for real time disparity maps calculation," *Microprocessors and Microsystems*, pp. 281 – 288, february 2012.

[2] E. Gudis, G. van der Wal, S. Kuthirummal, S. Chai, S. Samarasekera, R. Kumar, and V. Branzoi, "Stereo vision embedded system for augmented reality," SRI International, Tech. Rep., 2012.

[3] L. He-jian, T. Guo-wei, Z. Zhao-yang, A. Ping, M. Ran, W. Jian-wei, and W. Fu-qiong, "Hardware solution of real-time depth estimation based on stereo vision," Laboratory of Advanced Display and System Application Ministry of Education/Shanghai University, Xangai, China, Tech. Rep., 2012.

[4] P. Greisen, S. Heinzle, M. Gross, and A. P. Burg, "An fpga-based processing pipeline for high-definition stereo video," *EURASIP Journal on Image and Video Processing 2011*, 2011.

[5] J.-H. Yang, J. Im, K. Lim, and S.-J. Choi, "An asic design for 3d depth control of full hd resolution stereoscopic video," DTV SoC Division/SIC Center/LG Electronics Inc., Seul, Coréia do Sul, Tech. Rep., 2012.

[6] E. Tola, C. Strecha, and P. Fua, "Efficient large-scale multi-view stereo for ultra high-resolution image sets," *Machine Vision and Applications*, maio 2011.

[7] Panasonic. (2013) Panasonic-usa. [Online]. Available: http://www.panasonic.com/business/provideo/AG-3DA1.asp

[8] G. Bradsky and A. Kaehler, *Learning OpenCV*. Sebastopol, CA, Estados Unidos: O'rreily Media Inc., setembro 2008.

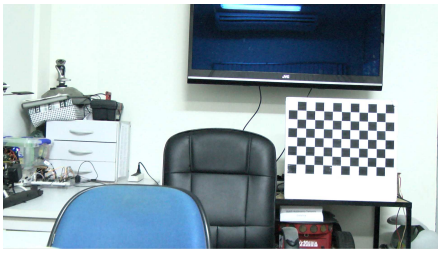
[9] R. Laganière, *OpenCV 2 Computer Vision Application Programming Cookbook*. Birminham, Reino Unido: Packt Publishing Ltd., maio 2011.

[10] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ, Estados Unidos: Prentice-Hall, 1982.

[11] *SDK-DeckLink*, december 2012.

[12] A. Olofsson, "Modern stereo correspondence algorithms: Investigation and evaluation," Master's thesis, Linköping University, Linköping, Sweden, june 2010.

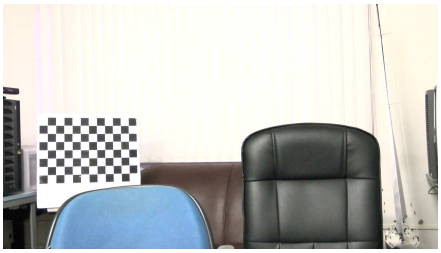
(a) Primeira imagem - direita



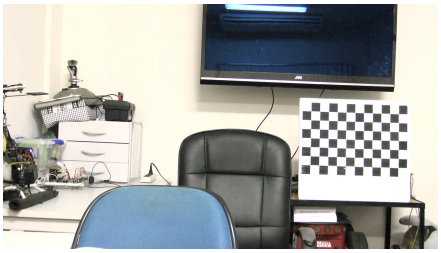
(b) Segunda imagem - direita

(c) Terceira imagem - direita

Fig. 3. Imagens originais - direita



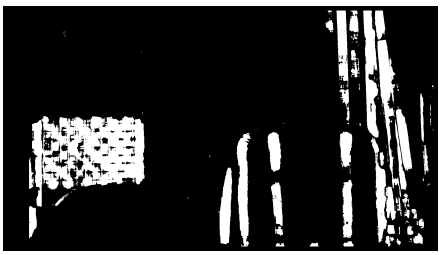
(a) Primeira imagem - esquerda



(b) Segunda imagem - esquerda

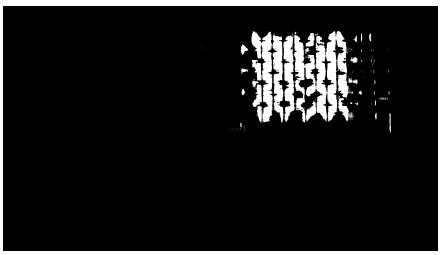
(c) Terceira imagem - esquerda

Fig. 4. Imagens originais - esquerda



(a) Primeiro par - SAD

(b) Segundo par - SAD

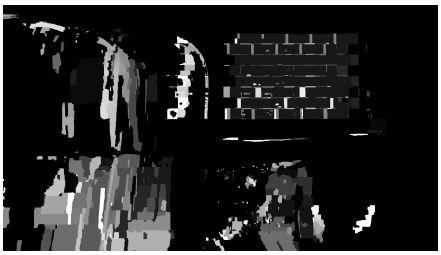


(c) Terceiro par - SAD

Fig. 5. Imagens de disparidade - SAD

(a) Primeiro par - *GraphCut*

(b) Segundo par - *GraphCut*



(c) Terceiro par - *GraphCut*

Fig. 6. Imagens de disparidade - *GraphCut*