

Rastreamento de Componentes Conexas em Vídeo 3D para Obtenção de Estruturas Tridimensionais

Manuscrito número 21829

Resumo

Este artigo apresenta um sistema que objetiva a geração de estruturas tridimensionais a partir de vídeo 3D. O sistema existente obtém, em tempo real, usando luz estruturada, tanto o modelo $2\frac{1}{2}D$ dos objetos quanto a textura dos mesmos. É descrita parte do trabalho já desenvolvido, consistindo de duas aplicações: (1) um método para detecção, rastreamento e casamento espacial de componentes conexas presentes em um vídeo 3D e (2) a aplicação de uma malha dinâmica adaptativa baseada em operadores estelares. Na primeira aplicação, a informação de imagem é combinada com posições tridimensionais a fim de alinhar partes de superfícies que são vistas em quadros subseqüentes. A abordagem que adotamos consiste na detecção de características salientes, tanto no espaço da imagem quanto no do mundo, provendo alinhamentos de textura e de geometria mais completos. Já a segunda aplicação envolve a construção de malhas que podem adaptar-se à geometria da superfície conforme esta evolui.

1. Introdução

Neste artigo, apresentamos duas aplicações desenvolvidas com vídeo 3D, além de um módulo de visualização do mesmo. As aplicações são (1) detecção, rastreamento e casamento de componentes conexas (CCs) e (2) aplicação de uma malha dinâmica adaptativa às componentes detectadas.

Para realizar a integração dos dados geométricos, primeiramente é preciso detectar as componentes conexas presentes em uma cena. Tal necessidade deve-se ao fato de que o casamento entre todas as componentes conexas (casamento global) é melhor feito quando cada uma delas é registrada individualmente, uma vez que é eliminada a influência do movimento relativo entre elas.

Dado que sistemas de vídeo 3D geralmente adquirem a textura da cena com uma resolução maior que a da geometria da mesma, uma abordagem razoável é uti-

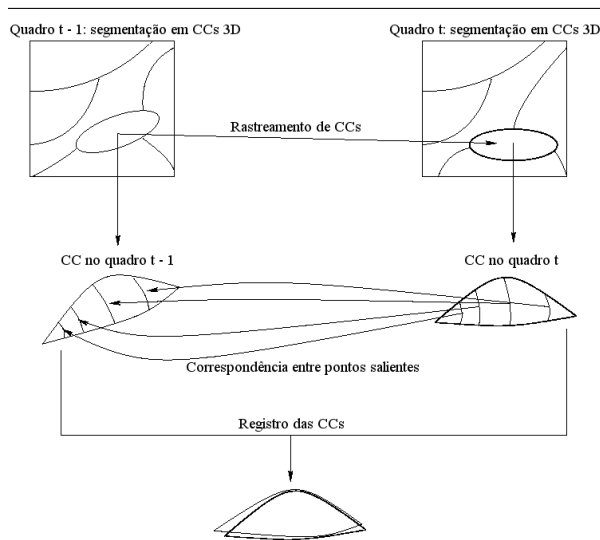


Figura 1. Visão geral do esquema proposto para rastreamento e casamento de componentes conexas em um vídeo 3D.

lizar a textura para melhorar o casamento da geometria. Trata-se de um passo intermediário importante e este documento contém o estudo que já foi desenvolvido neste sentido.

Os primeiros passos dados com o intuito de realizar a integração dos dados podem ser discriminados em três etapas: (1) rastreamento das componentes conexas; (2) identificação de pontos salientes correspondentes; (3) estimação de uma transformação geométrica para o registro tridimensional das componentes conexas. Esta seqüência de passos é esquematizada na Figura 1.

Também é descrita a geração de malhas para os modelos tridimensionais das CCs identificadas. De posse de tais malhas, podemos por fim fazer a integração das mesmas, tomando-as duas a duas e gerando uma malha final que acomode melhor as componentes conexas casadas. Este é um tópico que pretendemos abordar na continuação deste trabalho.

2. Vídeo 3D

A aquisição e manipulação de informação de vídeo 3D é um tópico chave da pesquisa recente em Visão Computacional e Computação Gráfica [6; 15]. Há vários problemas importantes em aberto na área e que a pesquisa no estado da arte começou a tratar nos últimos tempos.

A aquisição dos dados é realizada por um sistema proposto recentemente conhecido como *Vídeo 4D* [14], o qual provê ambas as informações de *geometria* e *textura* dinâmicas da cena em tempo real (taxa de amostragem de 30 quadros por segundo). Este sistema é parte de um projeto para a obtenção de *vídeo 4-dimensional*, que é um conceito recente. Um vídeo 4D pode ser entendido como *objetos 3D genéricos variando no decorrer do tempo*.

A idéia principal por trás do conceito de Vídeo 4D é a construção incremental da geometria estruturada da cena, ou seja, a geração de informação 3D completa *acumulada* com o passar do tempo ($4D = 3D + \text{tempo}$). Esta idéia representa um passo além do tradicional vídeo de profundidade (*range vídeo*) no qual cada quadro fornece informação de profundidade a partir do ponto de vista da câmera (ou seja, imagens *range*). O conceito de Vídeo 4D abre novas possibilidades interessantes e que foram muito pouco exploradas até agora. Até onde sabemos, ainda não há método algum que tenha resolvido completamente este problema.

A maior dificuldade é como integrar a informação $2\frac{1}{2}D$ de cada quadro ao longo do tempo. Um passo intermediário necessário é o rastreamento e o casamento da estrutura de objetos 3D presentes no decorrer do vídeo. O fato de os objetos poderem ser *deformáveis* (ou seja, *não-rígidos*) representa uma dificuldade adicional de todo o processo.

2.1. A Plataforma de Vídeo 3D

O sistema é uma plataforma completa para vídeo 3D, constituída por dispositivo de aquisição, processamento de dados, módulo de transmissão e módulo de visualização. Este sistema gera vídeo 3D em tempo real a partir de uma diversidade de cenas.

O dispositivo de aquisição emprega estéreo ativo, sendo composto por uma câmera de vídeo e um projetor, sincronizados e calibrados tanto geométrica quanto fotometricamente. O módulo de processamento de dados extrai informação sobre profundidade a partir de um código de luz estruturada. O módulo de transmissão trata da entrada e saída de dados, bem como da compressão e descompressão do fluxo dos mesmos. Finalmente, o

módulo de visualização renderiza vídeo 3D por meio da utilização de geometria dinâmica baseada em pontos.

O dispositivo de aquisição é construído com equipamento de vídeo NTSC facilmente encontrado no mercado. Isto traz diversas vantagens, tais como uma boa relação custo-benefício, facilidade de sincronização, compatibilidade e muitas opções de canais de distribuição.

2.2. Aquisição de Vídeo 3D em Tempo Real

Para detectar componentes geometricamente conexas em uma cena, o sistema de captura 3D deve fornecer imagens e geometria de alta qualidade em tempo real. Qualidade é crucial para a obtenção de análises e sínteses precisas. O tempo real é necessário para a exploração de coerência no tempo e para a captura de movimentos sutis de componentes conexas, bem como para a redução de problemas de casamento durante a análise espaço-temporal.

2.2.1. O Método (b, s)-BCSL A reconstrução da profundidade da cena é obtida por meio de triangulação das correspondências entre a câmera e o projetor. A chave de todo o processo de modelagem 3D do sistema é baseada em uma extensão do método (b, s)-BCSL de cores [10] para trabalhar com *cenar dinâmicas*, por meio da combinação com um vídeo sincronizado.

O código (b, s)-BCSL fornece um eficiente esquema de correspondência câmera/projetor. Trata-se de um código para fronteira de luz colorida estruturada, a qual consiste de uma seqüência de s padrões de faixas a serem projetados, cada um com b cores. O uso de padrões complementares é necessário para detectar, de forma robusta, transições de faixas e cores. O sistema utiliza 6 cores que podem ser detectadas, sem ambigüidades, por meio de cruzamentos com zero: vermelha, verde, azul, ciano, magenta e amarela. Este conjunto de cores é geralmente referenciado pela sigla RGBCMY, aludindo às iniciais dos nomes destas na língua inglesa.

Em nossos experimentos, foi adotado um código (6, 2)-BCSL, ou seja, 2 padrões compostos pelas 6 cores citadas, o que caracteriza mais de 900 combinações de faixas possíveis. Dois padrões é o mínimo permitido. A configuração escolhida fornece a melhor razão entre a coerência espaço-temporal e o comprimento do código.

O que o sistema emprega, na realidade, é uma seqüência de pares de padrões complementares: $S_1\bar{S}_1S_2\bar{S}_2$. Toda faixa colorida é projetada e seguida pela projeção de uma outra faixa que possui uma cor complementar, a fim de facilitar a detecção, de forma robusta, das fronteiras das faixas. Este cálculo

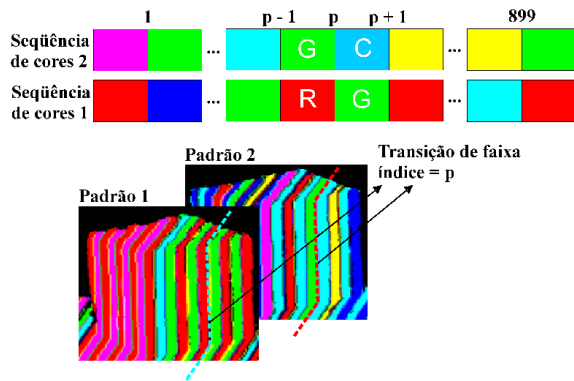


Figura 2. Decodificação das transições de faixas.

é feito a partir da diferença das duas imagens resultantes. As fronteiras das faixas tornam-se cruzamentos com zero nas imagens consecutivas e podem ser detectadas com precisão sub-pixel.

Um padrão S_1 é uma imagem composta por diversas faixas verticais coloridas, de forma que o padrão \bar{S}_1 é uma outra imagem, também composta por faixas verticais de mesma largura e posição que as faixas do padrão S_1 , só que com cores complementares. Este esquema faz com que a identificação do código seja muito robusta e a localização das fronteiras mais precisa. A reconstrução completa de uma geometria é obtida após a projeção de um padrão S_1 e seu complemento \bar{S}_1 seguida pela projeção de um padrão S_2 e seu complemento \bar{S}_2 .

Há uma outra vantagem na utilização de padrões complementares: foi verificado, empiricamente, que pares de padrões de cores complementares, quando projetados a 120 Hz, são percebidos como *luz branca constante*. Assim, o código (b, s)-BCSL complementado torna possível realizar a captura simultânea tanto da geometria quanto da textura da cena 3D. Ele também faz com que a construção de máscaras de sombra seja uma tarefa trivial.

Para a construção da correspondência câmera/projetor, projeta-se uma subsequência de dois padrões na cena e detectam-se as cores e as fronteiras das faixas projetadas a partir da imagem obtida por uma câmera de alta velocidade. As quatro cores projetadas, duas para cada padrão, detectadas próximas a qualquer fronteira, são unicamente decodificadas para a faixa projetada de índice p (feffig:decode). A coluna correspondente no espaço do projetor é detectada em $O(1)$ utilizando o processo de decodificação (6, 2)-BCSL. A profundidade é então calculada pelos parâmetros intrínsecos do par câmera/projetor e pela transformação rígida entre seus

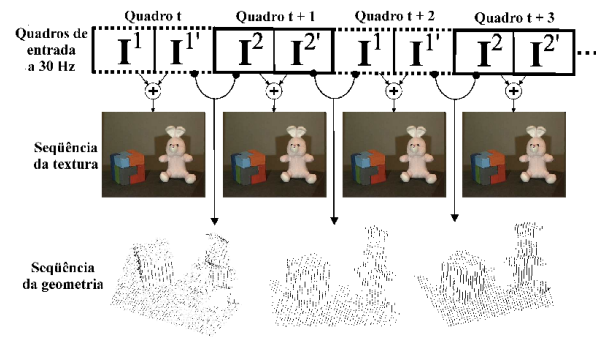


Figura 3. Esquema de captura envolvendo quadros do vídeo de entrada e a sequência de saída da textura e da geometria.

sistemas de referência.

O código (6, 2)-BCSL pode ser facilmente combinado com vídeo. Cada quadro de vídeo com resolução 640×480 no padrão NTSC é composto por dois campos 640×240 entrelaçados. Cada campo é exposto/capturado em $1/60$ de segundo. A câmera e o projetor são sincronizados utilizando-se *genlock* (do inglês *generator lock*), que se trata de uma técnica comumente usada para realizar a sincronização da saída de vídeo de um equipamento com a captura de vídeo de outro. Para a projeção, é gerada uma seqüência de quadros intercambiando os dois padrões que são codificados com seus correspondentes complementos com campos em um único quadro. Este sinal de vídeo é enviado ao projetor e conectado à entrada *genlock* da câmera. A soma dos dois campos fornece uma imagem de textura e sua diferença resulta nas cores e fronteiras das faixas projetadas. A aquisição completa da geometria e da textura é ilustrada na Figura 3.

Este sistema é adequado para o rastreamento de componentes porque mantém um bom balanço entre a detecção de textura, geometria e movimento. Nossos vídeos foram obtidos por meio da projeção de 70–90 faixas sobre as cenas, com diferentes escalas. São utilizados uma câmera Sony HyperHAD e um projetor Infocus LP-70.

2.3. Pipeline do Sistema

A Figura 4 nos dá uma visão geral do *pipeline* do processamento 3D. A seqüência $S_1\bar{S}_1S_2\bar{S}_2$ de padrões de faixas é projetada sobre a cena (Figura 4(a)). Para cada par $S_k\bar{S}_k$, são inferidas as cores das faixas (Figura 4(b)) e as respectivas fronteiras (Figura 4(c)). Na seqüência, códigos de fronteira são derivados a partir de padrões

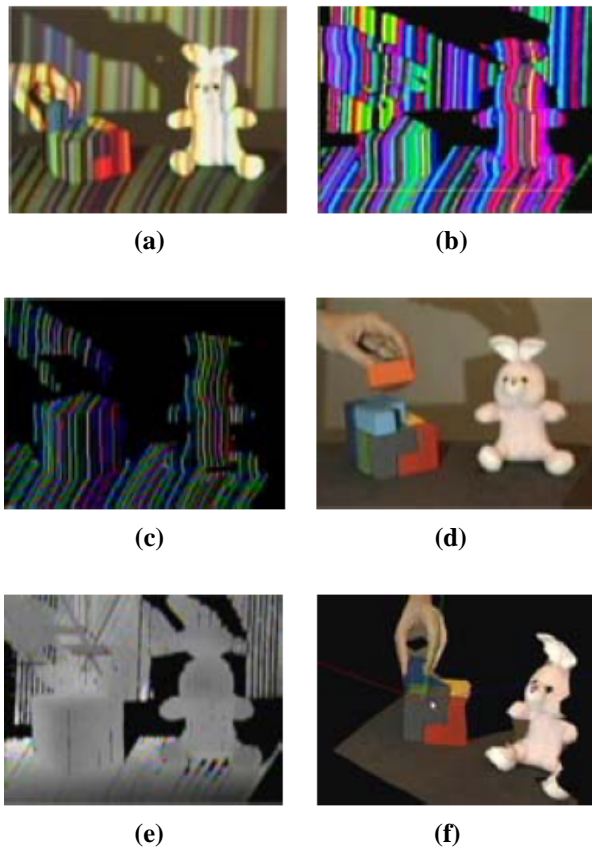


Figura 4. Seqüência de passos do sistema de vídeo 3D.

S_{ij} que se casam. Textura e máscara de sombra são obtidas pela combinação das imagens $S_k \overline{S_k}$ (Figura 4(d)). A profundidade é calculada por meio da triangulação das correspondências de fronteira na câmara e no projetor, seguida por interpolação (Figura 4(e)). Finalmente, o vídeo 3D é montado, transmitido e visualizado (Figura 4(f)).

A observação chave que explica porque a codificação (b, s)-BCSL funciona bem para cenas dinâmicas é que as fronteiras das faixas projetadas alteram muito pouco de um quadro para outro, mesmo quando objetos da cena movem-se rapidamente. A obstrução de fronteiras ocorre somente devido ao movimento dos objetos na frente da câmara.

3. Rastreamento e Registro de Componentes Conexas 3D

Um dos problemas importantes em aberto nas áreas de Visão Computacional e Computação Gráfica é o rastreamento e o casamento de *componentes conexas* geo-

metricamente em um vídeo 3D. A presente seção descreve uma nova solução para este problema, a qual é obtida por meio da integração das informações de *geometria* e de *textura* fornecidas pelo vídeo 3D.

Para cuidar deste problema, nossa abordagem inicia com a identificação de componentes conexas tridimensionais, em cada quadro, que representem objetos completos ou partes de objetos. Os próximos passos necessários são:

- ① rastrear as componentes conexas;
- ② identificar pontos salientes correspondentes;
- ③ estimar a transformação geométrica para registro 3D das componentes conexas.

3.1. Visão Geral da Abordagem Proposta

A Figura 1 ilustra nossa abordagem para o rastreamento de componentes conexas 3D. Dois quadros subsequentes são esquematicamente exibidos (denotados como quadro $t-1$ e quadro t). Primeiramente, as componentes conexas são detectadas para cada quadro. O procedimento de rastreamento identifica qual componente conexa no quadro $t-1$ corresponde a qual no quadro t .

Uma vez que as CCs tenham sido rastreadas ao longo da seqüência de vídeo, o próximo passo é casar cada par de CC nos quadros subsequentes. Com o objetivo de casar as CCs de uma maneira eficiente, três passos são seguidos:

- ① Pontos salientes são detectados no quadro t ;
- ② Os pontos correspondentes são identificados no quadro $t-1$;
- ③ Um alinhamento entre os pontos correspondentes nos quadros $t-1$ e t é executado.

A resolução da textura é maior que a da geometria porque foi adotado *estéreo ativo* (luz estruturada) para obter informação de geometria 3D. Baseado no conceito da taxa de Nyquist, a resolução da textura precisa ser pelo menos duas vezes a da geometria. Em nossos experimentos, ela é mais que o dobro. A imagem da textura provê características visuais a serem exploradas enquanto a geometria fornece características geométricas (forma) dos objetos na cena. Ambas são exploradas de uma forma complementar.

O sistema de captura de vídeo 3D envolve três sistemas de coordenadas diferentes:

Sistema de coordenadas 2D: é o domínio paramétrico da imagem da textura, assim como para a imagem de profundidade (isto é, geometria). Para cada ponto (u, v) na imagem é associada uma função de cor $c(u, v) = (R(u, v), G(u, v), B(u, v))$ assim

como uma informação de profundidade $d(u, v)$. Conforme explicado na Seção 2.2, a informação de profundidade não é calculada pelo sistema para todos os pontos (u, v) , mas ela pode ser obtida por interpolação dos pontos de amostragem. Usamos a notação $c(u, v, t)$ e $d(u, v, t)$ para denotar a textura e a imagem de profundidade do quadro no tempo t , respectivamente.

Sistema de coordenadas da câmera: este sistema de coordenadas é dado pelos parâmetros intrínsecos (por exemplo, distância focal, taxa de aspecto e centro de projeção) e extrínsecos (translação e rotação) da câmera. Baseando-se na parametrização (u, v) da imagem, é possível calcular a posição de um ponto da cena visto pela câmera: consideremos um ponto P amostrado na imagem de profundidade associado ao pixel $p = (u, v)$ e com profundidade $w_P = d(u, v)$. Podemos calcular o vetor direção de P como $\mathbf{v} = (P - O) / \|P - O\|$, em que O é a origem do centro de projeção. Todos os pontos pertencentes ao raio de visão determinado por O e P são da forma $\mathbf{r}(w) = O + w\mathbf{v}$, em que o parâmetro w indica a distância até a origem O na direção de \mathbf{v} . Então, a coordenada do ponto P da superfície é dada, no sistema de coordenadas da câmera, por $P = \mathbf{r}(w_P) = O + w_P\mathbf{v}$;

Sistema de coordenadas do mundo: é o sistema de coordenadas global da cena, no qual elementos da câmera e da cena estão embutidos. Uma vez que tanto os elementos da câmera quanto da cena podem se mover, suas posições e orientações são dependentes do tempo.

3.2. Detecção de Componentes Conexas

O primeiro passo para a detecção de componentes conexas é a construção de um grafo de conexão de pontos 3D. De modo simples, pontos cuja proximidade seja menor que um determinado limiar l são ditos *conexos*. Uma componente conexa é um grafo isolado de pontos conexas. Abordagens ingênuas para a construção de tais grafos podem possuir um tempo de complexidade muito alto.

Os métodos de decodificação usados em sistemas estéreo ativo fornecem automaticamente informação local que pode ser usada para a detecção de componentes conexas. As coordenadas j (linha no espaço da câmera) e p (número do plano projetado) de um ponto 3D definem uma topologia intrínseca.

Na verdade, pontos que estão próximos no espaço discreto (j, p) são fortes candidatos a estarem próximos

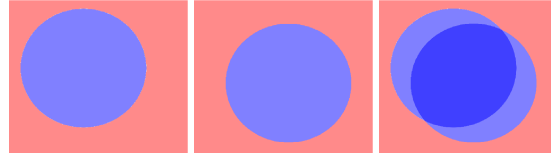


Figura 5. Área de sobreposição em quadros subsequentes.

no espaço 3D. Isto nem sempre é verdade devido a descontinuidades na profundidade da cena. Por outro lado, pontos distantes no espaço (j, p) também estão distantes no espaço 3D. Portanto, o uso de uma tabela (j, p) reduz significativamente o espaço de busca para o cálculo do grafo de conexão e dos diferentes componentes.

Neste esquema, dois pontos estão conectados se eles são k -vizinhos na tabela (j, p) e se a distância que os separa no espaço 3D é menor que o limiar l . É necessário ter cuidado porque a resolução de j é tipicamente muito maior que a resolução de p e, conseqüentemente, suas unidades são diferentes.

3.3. Rastreamento de Componentes Conexas

Como as CCs são segmentadas quadro a quadro independentemente, é necessário implementar um procedimento de rastreamento para identificar qual CC no quadro $t - 1$ corresponde a qual no quadro t . Diferentes eventos importantes precisam ser tratados pelo procedimento de rastreamento, tais como: CCs móveis que mudam de forma; novas CCs que aparecem na cena; componentes conexas antigas que desaparecem (por exemplo, movendo-se para fora da cena); CCs que se unem (por exemplo, dois objetos diferentes que se tocam em algum instante); CCs que se dividem (por exemplo, objetos que se tocavam e que separam-se movendo-se cada um para um lado).

O procedimento de rastreamento proposto considera quadros consecutivos de modo que possamos explorar o fato de que cada CC realiza pequenos movimentos entre quadros subsequentes devido à alta taxa de aquisição (30 quadros por segundo). Em tais casos, espera-se que a intersecção de uma CC c em um quadro t com sua correspondente CC no quadro $t - 1$ seja grande. A intersecção entre as CCs em quadros subsequentes é usada para rastrear as CCs (veja a Figura 5).

Seja $L = \{l_1, l_2, \dots, l_{|L|}\}$ um conjunto de rótulos de CCs em cada quadro. Rastrear uma componente conexa pode ser visto como a determinação de um mapeamento $M : L \rightarrow L$ tal que, para cada quadro no tempo t , se uma CC possui rótulo l_x em um quadro no tempo $t - 1$ e rótulo l_y no quadro no tempo t , então

$M(l_y) = l_x$. A imagem de profundidade $d(u, v, t)$ é usada para construir M . Para cada CC c no tempo t , a intersecção entre c e todas as CCs no tempo $t - 1$ é calculada. Somente as componentes conexas no tempo $t - 1$ com uma intersecção maior que um limiar definido a priori são consideradas como candidatas. Sejam c_1, c_2, \dots, c_n estas CCs candidatas no quadro $t - 1$. A diferença média entre as coordenadas de profundidade de c e c_1, c_2, \dots, c_n é calculada e a CC em $t - 1$ que minimiza esta diferença é escolhida como a candidata final:

$$m(c, c_i) = \frac{1}{|c \cap c_i|} \sum_{(u,v) \in c \cap c_i} |d(u, v, t) - d(u, v, t-1)|$$

$i = 1, 2, \dots, n$. Um limiar de distância máximo d_{\max} é determinado de modo que, se $\min_i m(c, c_i) > d_{\max}$, então nenhum mapeamento entre os quadros $t - 1$ e t é criado para c . Neste caso, c é considerada como sendo uma nova CC que surgiu no quadro t , ou seja, ela não estava presente no quadro $t - 1$. Um novo rótulo é então atribuído a c . Em nossos experimentos, utilizamos $d_{\max} = 1.5$ e a intersecção mínima aceitável foi de 30% da área da componente conexa sendo analisada.

3.4. Casamento de Componentes Conexas usando Informações da Textura e da Geometria

Uma vez que as CCs são rastreadas, cada par de CCs correspondentes em quadros subseqüentes deve ser alinhado. Isto é feito por meio da aplicação do algoritmo iterativo de pontos mais próximos (ICP) [1] a um conjunto de pontos salientes selecionados a partir das CCs. O algoritmo ICP é amplamente utilizado para o alinhamento de modelos tridimensionais baseando-se somente na geometria, e às vezes na cor, das malhas. Três passos são seguidos a fim de selecionar os pontos que alimentam o algoritmo ICP:

① **Alinhamento da textura:** As porções da textura correspondentes às CCs consideradas são extraídas e casadas por meio de correlação. O ponto máximo de correlação das duas porções de textura $c(u, v, t) \circ c(u, v, t - 1)$ indica a translação que uma porção deveria sofrer a fim de ocorrer um casamento com a outra. Uma correspondência ponto a ponto entre as duas porções $c(u, v, t)$ e $c(u, v, t - 1)$ é então estabelecida. A principal vantagem de usar textura para criar esta correspondência entre dois quadros, em vez da geometria, é a alta resolução apresentada pelo primeiro;

② **Identificação de pontos salientes nos dados da geometria (quadro t):** Os dados da geometria $d(u, v)$

representam a informação de profundidade medida ao longo dos padrões de luz projetados na cena. Os conjuntos de pontos máximo e mínimo locais ao longo de cada faixa de luz no quadro t são tomados como pontos salientes. Estes pontos são identificados por meio de diferenciação numérica de $d(u, v, t)$ ao longo de cada faixa de luz;

③ **Identificação de pontos salientes correspondentes nos dados de geometria (quadro $t - 1$):** A posição na textura $c(u, v, t)$ de cada ponto saliente de $d(u, v, t)$ é identificada. Devido ao procedimento de alinhamento por correlação da textura, esta também é a posição dos pontos salientes na imagem alinhada da textura $c(u, v, t - 1)$. É importante notar que, devido à alta resolução da imagem da textura, a posição de um ponto saliente no quadro $t - 1$ pode não ter um ponto correspondente amostrado nos dados da geometria do quadro $t - 1$. Portanto, o ponto saliente no quadro $t - 1$ é obtido por interpolação dos pontos da geometria.

O procedimento acima leva a dois conjuntos de pontos salientes correspondentes dos dados da geometria dos quadros t e $t - 1$. Estes dois conjuntos de pontos são então registrados usando o algoritmo ICP, enfim produzindo o resultado desejado.

3.5. O Algoritmo ICP (*Iterative Closest Point*)

O algoritmo ICP foi introduzido por Besl e McKay [1] e tornou-se a principal técnica para o registro de conjuntos de dados 3D. O algoritmo recebe dois conjuntos de dados 3D como entrada. Sejam P e Q dois conjuntos de dados contendo N_p e N_q pontos respectivamente. Ou seja, $P = \{p_i\}, i = 1, \dots, N_p$ e $Q = \{q_i\}, i = 1, \dots, N_q$. O objetivo é calcular uma matriz de rotação R e um vetor de translação t tais que o conjunto transformado $P' = RP + t$ fique bem alinhado com Q . Segue abaixo um sumário do algoritmo (veja a Figura 6 para uma ilustração do funcionamento do ICP).

① **Inicialização:** $k = 0$ e $P_k = P$.

② **Cálculo do ponto mais próximo:** Para cada ponto em P_k , calcula o ponto mais próximo em Q . Consequentemente, produz um conjunto de pontos mais próximos $C = \{c_i\}, i = 1, \dots, N_p$ em que $C \subset Q$ e c_i é o ponto mais próximo a p_i .

③ **Cálculo do registro:** Dado o conjunto de pontos mais próximos C , a função objetivo de quadrados médios a ser minimizada é:

$$f(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|c_i - Rp_i - t\|^2$$

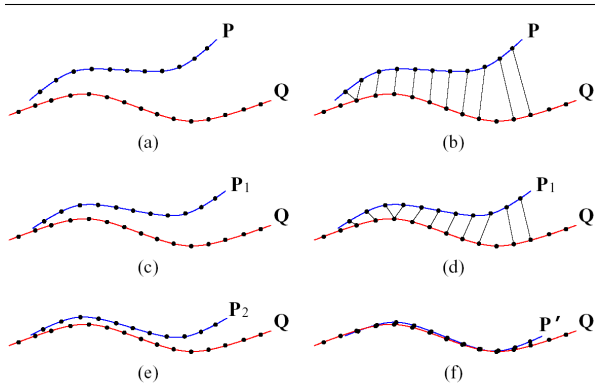


Figura 6. Ilustração do algoritmo ICP.

Note que p_i é um ponto do conjunto original P , não de P_k . Portanto, o registro calculado aplica-se ao conjunto de dados originais P ao passo que os pontos mais próximos são calculados usando P_k .

- ④ **Aplicação do registro:** $P_{k+1} = RP + t$.
- ⑤ **Se a precisão desejada para o registro é alcançada:** Finalize a iteração.
Senão: $k = k + 1$. Repita os passos 2–5.

4. Biblioteca de Malha Dinâmica Adaptativa baseada em Operadores Estelares

Na continuação deste trabalho, foi utilizada uma biblioteca de malhas chamada A48 [11; 12]. Tal biblioteca mantém uma triangulação uniforme de superfícies que variam com o tempo. O usuário fornece uma malha inicial, um procedimento de amostragem da superfície e um conjunto de critérios de adaptação. Conforme a superfície sofre alterações com o tempo, a malha é automaticamente modificada, a fim de manter-se segundo as características definidas pelo usuário. A representação da malha é baseada em uma estrutura de dados conhecida por meia-aresta, desprovida de qualquer requisito extra para armazenamento. A malha possui uma estrutura subjacente de multiresolução semi-regular. Além disso, a especificação das características da malha desejada pode ser baseada em regras de adaptação muito gerais. Este esquema facilita o desenvolvimento de aplicações gráficas que manipulam superfícies trianguladas.

4.1. Visão Geral

Malhas poligonais são indiscutivelmente a representação mais comum para superfícies em Modelagem Geométrica e Computação Gráfica.

Em muitas aplicações, a superfície é dinâmica e muda sua forma com o passar do tempo. Isto abrange um grande número de problemas, da animação de corpos deformáveis à multiescala progressiva para transmissão e visualização. Há outras aplicações em que a forma da superfície é fixa, mas a discretização da mesma deve ser alterada por razões computacionais. Isto ocorre, por exemplo, no caso de simulações de elemento finito (FEM, do inglês *Finite Element Simulations*) com corpos rígidos.

Esta subseção apresenta um esquema simples para criação e manutenção da representação de uma malha de superfícies que variam com o passar do tempo e que trata de todos os requisitos mencionados acima. Como um benefício adicional, a representação da malha possui uma estrutura subjacente de multiresolução semi-regular, a qual pode ser explorada de diversas formas. A presente descrição baseia-se no artigo em que a biblioteca A48 foi apresentada [11].

Mais especificamente, as características relevantes deste programa são:

- ☞ Uma biblioteca de malha dinâmica adaptativa simples baseada na *meia-aresta*, uma estrutura de dados topológica padrão. A implementação desta nova funcionalidade de multiresolução adaptativa não requer nenhum armazenamento extra na representação. Também, uma vez que a meia-aresta é largamente adotada, é fácil incorporar a biblioteca em muitas aplicações.
- ☞ Uma API mínima para criação e adaptação de malhas. Esta interface complementa os tradicionais operadores de consulta topológica e consiste de apenas poucas funções.
- ☞ Uma estrutura de malhas conforme, que altera dinamicamente sua resolução baseada em critérios definidos pelo usuário. Isto torna as potencialidades de adaptação associadas muito gerais e poderosas.
- ☞ Um mecanismo efetivo para o refinamento e a simplificação de malhas semi-regulares que mantém uma estrutura de multiresolução restrita. Este mecanismo é baseado no conceito de uma multi-triangulação binária restrita e na teoria estelar.

A representação de malha dinâmica da biblioteca A48 possui uma estrutura subjacente de multiresolução. Representações de multiresolução podem ser definidas por meio de operações locais ou globais em uma malha [3]. Com o objetivo de dar suporte à adaptação, a estrutura de dados de multiresolução deve ser construída usando operações locais. Malhas progressivas [5] constituem um exemplo de tal estrutura de dados. Outro exemplo é a malha 4-K hierárquica [13]. Neste tipo

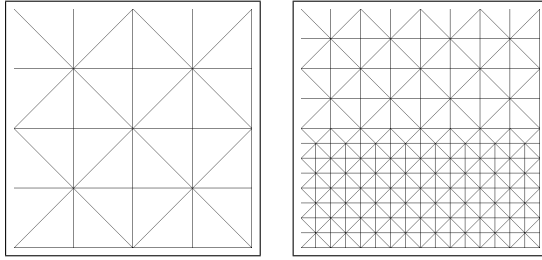


Figura 7. Malha dinâmica adaptativa.

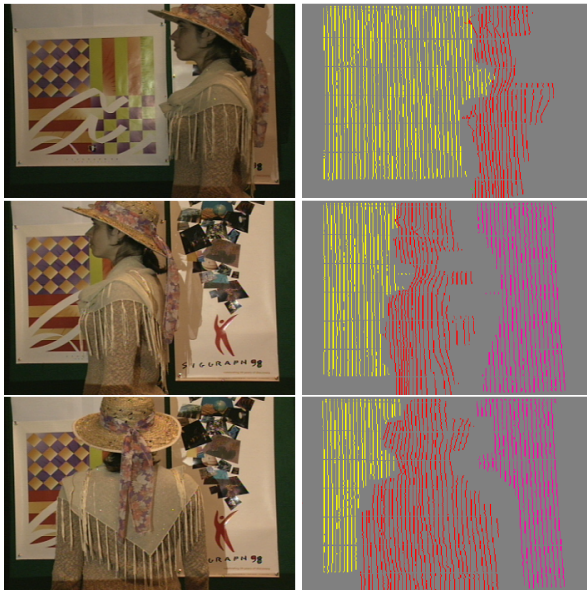


Figura 8. Rastreamento de componentes conexas em vídeo.

de representação, diferentes malhas podem ser extraídas dinamicamente a partir da estrutura de dados. Contudo, operações locais precisam ser explicitamente armazenadas. No esquema em questão, somente a malha atual é armazenada na representação. Os movimentos legais que alteram a resolução da malha são definidos implicitamente.

A Figura 7 mostra um exemplo da malha aplicada a uma superfície bidimensional (à esquerda) e após serem realizadas duas refinações e considerar o limiar na metade da malha (à direita).

5. Resultados Experimentais

Nesta seção, apresentamos resultados experimentais usando seqüências de vídeo 3D. A Figura 8 mostra a

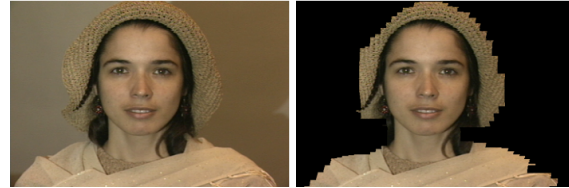


Figura 9. Casamento de texturas.

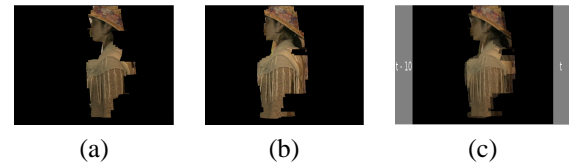
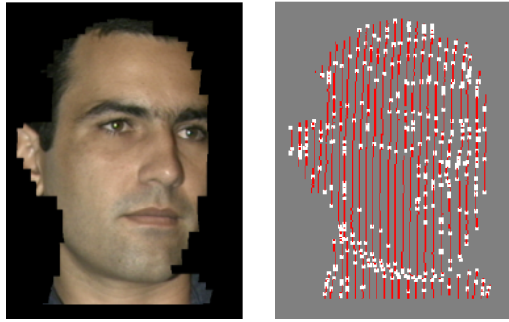


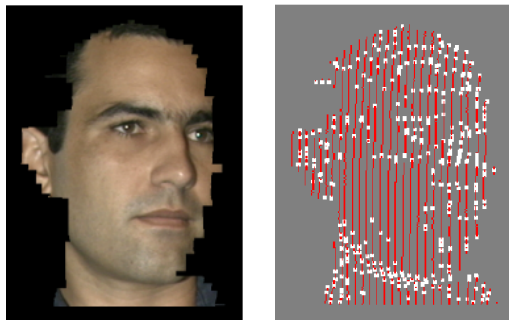
Figura 10. Translação baseada no casamento de textura.

informação de textura de três quadros de uma seqüência de vídeo (coluna da esquerda) juntamente com os correspondentes dados de geometria segmentada (coluna da direita). No primeiro quadro (linha do topo) há uma pessoa em frente a um fundo plano, portanto definindo duas componentes conexas no espaço da geometria. As CCs são codificadas por meio de cores diferentes. À medida que a pessoa se move, as CCs correspondentes são rastreadas conforme esperado.

Um exemplo da informação de textura e da segmentação correspondente usando a geometria da CC é mostrado na Figura 9. Esta textura segmentada é usada pelo procedimento de alinhamento de textura para criar um mapeamento entre as texturas de quadros subsequentes, como mostrado na Figura 10. Nela, (a) e (b) mostram a textura de uma CC nos quadros $t - 10$ e t , respectivamente. Em (c) vemos o quadro t transladado em $(u, v) = (1, 71)$ com relação ao quadro $t - 10$ e sobreposto a este último. O quadro t é mostrado com 50% de transparência. Usamos uma diferença de 10 quadros apenas com o propósito de uma melhor visualização. Em quadros consecutivos, a translação geralmente é pequena. Os pontos salientes anteriormente calculados para o quadro mostrado na Figura 11(a) são mostrados na Figura 11(b), juntamente com seus pontos salientes interpolados correspondentes no quadro anterior (Figuras 11(c) e 11(d)). Os pontos salientes casados resultantes da aplicação do algoritmo ICP (o programa *Scanalyze* tem sido usado em nossos experimentos—<http://graphics.stanford.edu/software/scanalyze/>) são mostrados na imagem (a) da Figura 12, corroborando assim o método introduzido. A imagem (b) mostra um

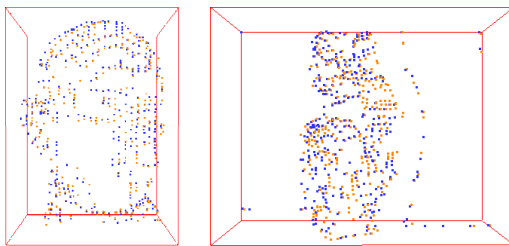


(a) (b)



(c) (d)

Figura 11. Dados de profundidade e pontos salientes.



(a) (b)

Figura 12. Casamento realizado pelo algoritmo ICP.

outro conjunto de pontos alinhados de acordo com a Figura 10.

A Figura 13 mostra a aplicação da malha dinâmica adaptativa sobre a superfície de um quadro obtido a partir de um vídeo 3D.

Para tornar possível o experimento de novos algoritmos com os dados oriundos do vídeo 3D, deu-se início à implementação de um sistema de visualização

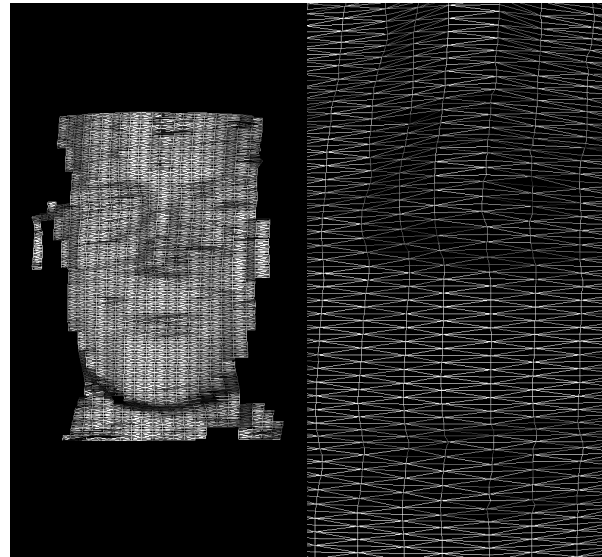


Figura 13. Malha A48 no vídeo 3D.

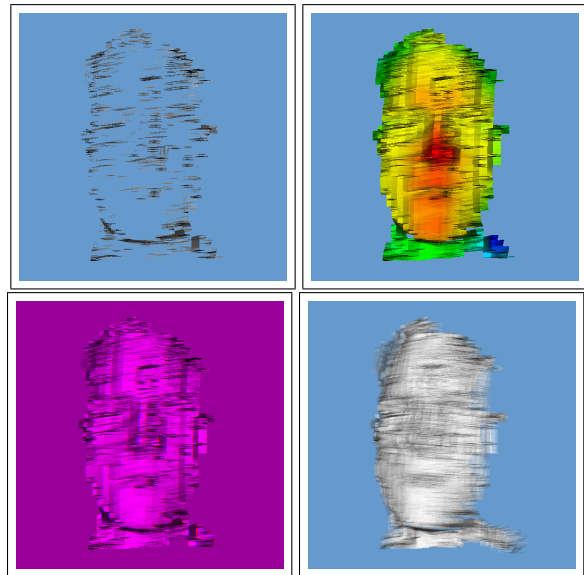


Figura 14. Aplicação do sistema de visualização.

do mesmo, cuja aplicação é mostrada na Figura 14. Nela temos, de cima para baixo, da esquerda para a direita: decimação quádrupla com 80% de redução dos dados, mapeamento de cores segundo a profundidade, renderização estéreo vermelho-azul e efeito de borramento por movimento da câmera.

6. Discussão e Conclusão

O método para rastreamento e casamento de componentes conexas apresentado tem se mostrado efetivo em nossos experimentos, sendo conveniente para realizar a integração de vídeos 3D. Este é um passo importante para a construção do vídeo 4D por meio da acumulação da geometria no tempo. O método lida corretamente com diferentes eventos que podem ocorrer durante o vídeo, tais como o surgimento, desaparecimento e a divisão de CCs.

Uma das originalidades previstas por este trabalho mostra-se tanto na reunião inédita de técnicas já conhecidas como na melhoria e adaptação do algoritmo ICP aos dados oriundos de um vídeo 3D, objetivando a modelagem tridimensional. Com isso, será obtido um avanço inicial em direção ao conceito de vídeo 4D [14].

6.1. Trabalhos Futuros

Embora o programa ICP adotado leve a resultados satisfatórios, parte de nosso trabalho atual está focado no aperfeiçoamento da implementação do ICP de modo a permitir o registro de CCs em tempo real [2; 4; 7–9]. Isto será feito a partir de alguma biblioteca já implementada e disponível, possibilitando o desenvolvimento de uma interface para trabalhar com os dados fornecidos pelo sistema de vídeo 3D.

Filtros de Kalman serão aplicados a fim de acelerar todo o processo de vídeo 4D por meio da combinação de diferentes fontes de dados disponíveis para prever a posição e a pose das CCs conforme novos quadros são adquiridos.

Uma aplicação que se tem em mente é o rastreamento de faces em cenas de vídeo, envolvendo a segmentação e a reconstrução 3D das mesmas.

Finalmente, um procedimento para a união de formas usando a informação de registro das CCs está atualmente em desenvolvimento e será divulgado no devido tempo.

Referências Bibliográficas

- [1] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Fevereiro 1992.
- [2] G. Dalley. A software test-bed for the registration of 3D range images. Master's thesis, Ohio State University, 2000.
- [3] M. Garland. Multiresolution modeling: Survey & future opportunities. In *Eurographics'99, State of the Art Report (STAR)*, 1999.
- [4] S. Gupta, K. Sengupta, and A. A. Kassim. Compression of dynamic 3D geometry data using iterative closest point algorithm. *Computer Vision and Image Understanding*, 87:116–130, 2002.
- [5] H. Hoppe. Efficient implementation of progressive meshes. *Computer and Graphics*, 22(1):27–36, 1998.
- [6] P. S. Huang, C. Zhang, and F.-P. Chiang. High-speed 3-D shape measurement based on digital fringe projection. *Optical Engineering*, 42(1):163–168, 2003.
- [7] A. Johnson and S. B. Kang. Registration and integration of textured 3D data. *Cambridge Research Laboratory - Technical Report Series*, Outubro 1996.
- [8] T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding*, 61(3):295–307, 5 1995.
- [9] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. *Third International Conference on 3D Digital Imaging and Modeling*, 2001.
- [10] A. Sá, P. C. Carvalho, and L. Velho. (b, s)-BCSL: Structured light color boundary coding for 3D photography. In *Proceedings the 7th Fall Workshop on Vision, Modeling, and Visualization*, 2002.
- [11] L. Velho. A dynamic adaptive mesh library based on stellar operators. *Journal of Graphics Tools*, 9(2):1–29, 2004.
- [12] L. Velho. A48: A dynamic adaptive mesh library, 2006. <http://w3.impa.br/~lvelho/a48/>.
- [13] L. Velho and J. Gomes. Variable resolution 4-K meshes: Concepts and applications. *Computer Graphics Forum*, 19:195–212, 2000.
- [14] M. B. Vieira, L. Velho, A. Sá, and P. C. Carvalho. A camera-projector system for real-time 3D video. In *Proceedings of IEEE International Workshop on Projector-Camera Systems (PROCAMS)*, San Diego, California, USA, Junho 2005. Em conjunto com CVPR 2005.
- [15] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Space-time faces: High resolution capture for modeling and animation. *ACM Transactions on Graphics*, 23(3):548–558, 2004.