

Extração Automática de Keyframes para Reconstrução 3D de Cenários Virtuais para Games

Sérgio Carlos Portari Júnior

Departamento de Ciências Exatas e da Terra
Universidade do Estado de Minas Gerais - UEMG
Frutal – MG - Brasil
portari@gmail.com

Antonio Carlos Sementille

Departamento de Computação
Univ. Estadual Paulista Júlio de Mesquita Filho - UNESP
Bauru – SP - Brasil
semente@fc.unesp.br

Resumo — O uso de cenários virtuais em produção de jogos mais realistas vem crescendo com o avanço e barateamento das tecnologias de *hardware* e *software* necessárias a esta utilização. Mas para se obter um cenário virtual que convença o usuário pode-se utilizar técnicas de reconstrução tridimensional fotorealistas como uma possível alternativa. Este trabalho apresenta uma proposta de um método para o pré-processamento de um vídeo, capturado no mundo real, onde serão extraídos *frames* adequados à reconstrução 3D (*keyframes*) pelo método SFM (*Structure From Motion*). Desta forma o processamento para a reconstrução 3D utilizará apenas os *frames* considerados essenciais, diminuindo o esforço computacional e as falhas.. Com a utilização deste método conseguiu-se reduzir o tempo de processamento durante a reconstrução 3D sem prejudicar o resultado visual do objeto reconstruído, comparando-se com os métodos tradicionais de seleção de imagens, onde não existe uma prévia seleção de *keyframes*, utilizando-se diferentes ferramentas de reconstrução baseadas no método SFM.

Palavras Chave — *Cenário virtual; Structure from motion – SFM; Reconstrução 3D; Visão computacional.*

I. INTRODUÇÃO

A crescente evolução dos *hardwares* gráficos na Computação tem possibilitado o desenvolvimento de técnicas mais apuradas de produção de cenários. Como visto em [1], uma destas técnicas é conhecida como *chroma-key* e começou a ser utilizada inicialmente em cenários de televisão. Utilizando esta técnica é possível produzir uma composição de imagens, permitindo desde a colocação de um ator em qualquer lugar do mundo até a criação de um mundo totalmente novo, que não existe, numa produção de jogos foto-realistas.

Os cenários podem ser muito complexos e exigem grandes espaços e construções para parecerem reais. Além disso, requerem um grande espaço de armazenamento quando não estão sendo utilizados. Estes fatores podem elevar o custo da produção podendo, inclusive, inviabilizar algumas produções que podem ser cruciais para seu desenvolvimento.

Para conseguir amenizar estes problemas, alternativas envolvendo computação estão sendo criadas. Uma delas é a utilização de estúdios virtuais na criação de cenários virtuais. Segundo [2], o termo “estúdio virtual” foi criado para distinguir estes novos sistemas que permitem a utilização, por meio de Computação Gráfica, de objetos tridimensionais

virtuais, dos tradicionais sistemas de *chroma-key*. Os Jogos passaram a utilizar este recurso de composição de cenas mais realísticas.

A modelagem de objetos 3D, de forma manual, não é trivial e geralmente não produzem resultados satisfatórios quanto ao aspecto realismo.

Um método alternativo é o *Structure From Motion* (SFM), como em [3].

Este método requer um conjunto de imagens (fotos), capturadas previamente de um objeto, que pode ser difícil de conseguir na primeira tentativa de captura, mesmo por quem conhece o método.

Uma alternativa é utilizar um fluxo de vídeo que possui um maior número de imagens em sequência, aumentando as possibilidades de captura de nuances importantes sobre o objeto ou cena a ser reconstruída.

O objetivo principal do trabalho é, a partir do levantamento das características consideradas essenciais à reconstrução baseada em SFM, desenvolver um protótipo em *software* que atue como filtro para seleção automática de *frames*. Portanto, considerando o pipeline resumido de reconstrução baseado em SFM (Figura 1), o enfoque do trabalho localiza-se sem sua fase inicial, ou seja, a etapa de captura e seleção de imagens.

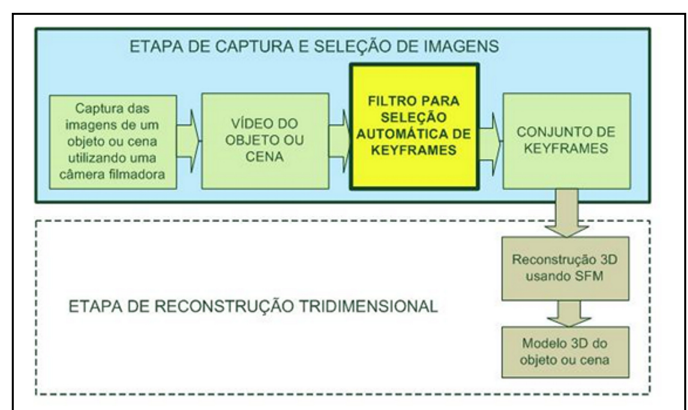


Figura 1 – Etapa de captura e seleção de imagens: foco do trabalho.

II. O MÉTODO SFM

Como visto em [3], [4] e [5], a partir de imagens sequenciais, adquiridas movimentando uma câmera em torno de um objeto ou ambiente, é possível obter uma representação 3D em escala de um objeto e, em seguida, texturizá-la com as próprias imagens reais obtidas para aumentar o realismo da reconstrução.

O primeiro passo para iniciar uma reconstrução por SFM é a extração dos pontos característicos das imagens. Os pontos, denominados pontos característicos (*keypoints*), são cantos, retas, circunferências ou pontos que diferencie-os dos pontos vizinhos.

O passo seguinte é a detecção da correspondência das características entre um par de imagens e as poses das câmeras. Pelas poses das câmeras é possível determinar se a câmera deslocou-se o suficiente para justificar uma escolha do frame ou se ele deve ser descartado.

Além disso, é necessário que as imagens não estejam borradas devido ao movimento brusco da câmera (*motion blur*). Uma imagem borrada é uma imagem produzida por um movimento brusco da câmera, o que faz com que esta imagem não seja visível completamente. Esta imagem borrada prejudica a triangulação dos pontos, no momento da reconstrução, pois a detecção de pontos característicos e correspondentes da imagem o objeto necessita de imagens bem nítidas, e prejudica, também, a texturização do objeto final.

III. SISTEMA DESENVOLVIDO

O propósito deste trabalho é investigar, implementar e validar um protótipo de sistema que permita a geração automática de um conjunto de imagens adequado (*keyframes*) para a etapa de construção de cenários virtuais, baseado no algoritmo de reconstrução por SFM.

Sua utilização permite selecionar um vídeo de entrada e, com a aplicação de filtros de diferença de *pixels* e filtro de borramento, individualmente ou combinados, obter um conjunto de imagens (*keyframes*) que sejam satisfatórios para uma reconstrução em 3D pelo método SFM, por meio de um *software* que utilize este tipo de algoritmo.

A utilização se dará na etapa de pré-processamento das imagens, antes da reconstrução, como mostrado na Figura 1.

Nem todas as características do método SFM foram analisadas, pois a ideia central deste trabalho não é a realização de uma reconstrução completa.

Considerou-se que os *frames* descartados são aqueles que possuem pouca diferença de movimento de câmera, comparados com o limiar especificado pelo usuário. Caso isto ocorra será prejudicial ao método, pois pode atrapalhar na detecção de pontos de correspondência e no momento de realizar a texturização do modelo. Aqueles *frames* que estiverem abaixo do limiar de borramento estipulado pelo usuário serão, então, descartados.

O *software* é composto por três módulos principais e um módulo auxiliar, conforme ilustrado na Figura 2.

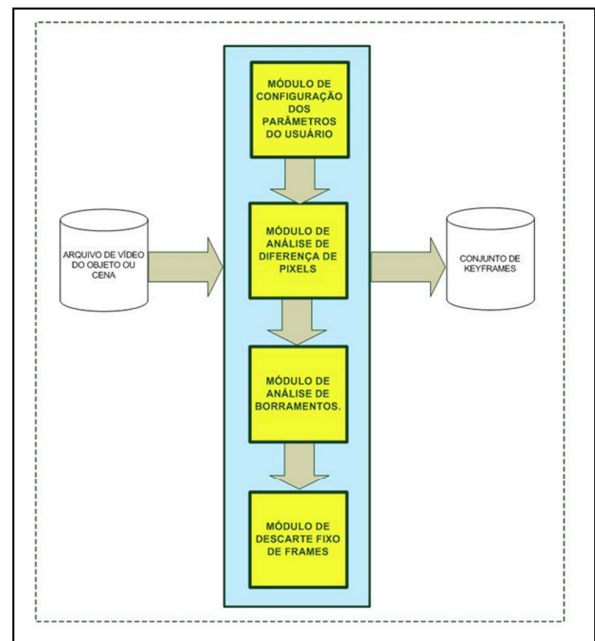


Figura 2 – Estrutura dos filtros do protótipo desenvolvido

O primeiro módulo é o de configuração dos parâmetros. Sua função é obter os parâmetros de configurações para a realização da extração dos *keyframes*, como nome do arquivo de entrada, pasta e arquivos de saída, dentre outras configurações.

O segundo módulo é o filtro de diferença de *pixels*, que realiza a comparação das quantidades de *pixels* que sofreram alterações, de um frame para outro, comparando de acordo com um limiar especificado pelo usuário.

A diferença é feita convertendo-se duas imagens (*frames* do vídeo) coloridas para imagens em escala entre preto e branco e realizando a subtração de uma imagem pela outra.



Figura 3 – Imagem resultante da diferença de *pixels*

O resultado é uma imagem, como vista na Figura 3, em que pode-se extrair o valor de quantos *pixels* foram diferentes de uma imagem para outra.

Este valor será comparado com o limiar determinado pelo usuário e, caso supere-o, selecionará o frame, do contrário irá descartá-lo e é extraído o próximo frame do vídeo para repetir-se o processo de diferença deste novo frame obtido com a diferença para o primeiro frame analisado na operação anterior.

No terceiro módulo foi desenvolvido um filtro de borramento, que realiza uma checagem do nível de borramento da imagem e, de acordo com um limiar do usuário, determina se o frame deve ser descartado ou selecionado.

Este filtro foi baseado no trabalho de [6], que descreveu uma métrica para detectar o borramento de acordo com o conceito de que o borramento de uma imagem $i(x,y)$ é dado pela distorção de um núcleo $d(x,y)$ que produz esta imagem borrada $b(x,y)$.

Calculando a força espectral (power spectrum) $PS(x,y)$, que é uma imagem escalar, entre 0 e 1, dado por:

$$PS(x,y) = \log(1 + |F(x,y)|^2) / \log(1 + |F(0,0)|^2) \quad (\text{EQ. 1})$$

onde F é a transformada rápida de Fourier (*Fast Fourier Transform*). Quanto mais próximo de zero for a força espectral, menor será o borramento da imagem analisada.

O módulo procura pelos *frames* abaixo de um limiar especificado pelo usuário para considerar a imagem borrada (valor acima do limiar) ou não (valor abaixo do limiar).

O módulo descarte de *frames* realiza saltos fixos de *frames*, estipulados pelo usuário, e também é utilizado pelos módulos de borramento e diferença de *pixels*.

Este módulo pode ser utilizado para provocar saltos constantes, com uma quantidade fixa especificada pelo usuário, para poder acelerar o processo em vídeos onde o movimento da câmera é muito lento ou para extrair todos os *frames* do vídeo, sem aplicar nenhum dos filtros.

O salto dos *frames* é feito extraindo-se o frame do vídeo e automaticamente descartando-o em seguida, comparando com o limiar especificado pelo usuário.

IV. TESTES

Todos os testes foram realizados utilizando o mesmo *hardware* descrito nesta seção.

O *hardware* utilizado foi um notebook Dell Inspiron 5010 dotado de um processador Intel Core i5 M450, com 2 núcleos de 2.40Mhz, 4Gb de memória RAM, com sistema operacional Windows 7 Ultimate, 64bits.

Para captura dos vídeos foram utilizadas: uma câmera Sony H9, que gera vídeos em arquivos Mpeg com resolução de 640x480 *pixels* e uma câmera Nikon Coolpix S3100, que gera arquivos de vídeos Avi com resolução de 1280x720 *pixels*.

Para realizar a contagem de pontos e polígonos da malha gerada por SFM, foi utilizado o *software* Meshlab [7] na versão 1.3.2, da Visual Computing Labs, ISTI – CNR. Ele é gratuito e

permite manipulação de malhas, pontos e texturas, bem como trabalha com uma grande variedade de formatos de arquivos.

Foram realizados dois tipos de testes para verificar a eficácia do protótipo de extração de *keyframes*.

No primeiro teste foram realizadas reconstruções com os *softwares* Autodesk 123D Catch [8] e Visual SFM [9], onde foram comparados os números de pontos e faces gerados e observação visual da qualidade da reconstrução obtida.

No segundo teste foi utilizado o *software* Visual SFM [9] e comparado o tempo de processamento para realizar uma reconstrução com todos os *frames* do vídeo e apenas com os *keyframes* selecionados pelo protótipo. Foram comparados os números de pontos gerados, entre os *frames* selecionados pelo protótipo e todos os *frames* de um vídeo, e o tempo de processamento.

A escolha do Autodesk 123D Catch [8] foi feita devido ao resultado final da reconstrução estar em forma de uma malha de triângulos, texturizados com as imagens obtidas dos *keyframes*. Este formato é totalmente compatível com a maioria dos *softwares* de modelagem do mercado e facilmente importado pelos programadores de jogos. A única desvantagem do *software* é seu processamento ocorrer nos servidores da empresa Autodesk, ou seja, nas nuvens. Isto torna necessária uma conexão com a internet para envio dos *keyframes* e download do resultado. Com isso, não existe uma forma de avaliar o tempo de processamento.

Já a escolha do Visual SFM [9], apesar de não resultar em uma malha poligonal, se justifica por ser uma aplicação que pode ser executada localmente e resulta em uma nuvem de pontos, que podem ser comparados com os pontos resultantes na reconstrução com o Autodesk 123D Catch [8].

Os resultados podem ser vistos na Tabela 1. Uma ilustração dos testes é apresentada na Figura 4.

TABELA I – RESULTADO DOS TESTES

Objeto	RESULTADO DOS TESTES				
	Resolução	Tempo todos frames	Pontos todos frames	Tempo Keyframes	Pontos Keyframes
Carrinho	640x480	5221s	2031	188s (3,6%)	1791 (88,2%)
Esfinge	640x480	43751s	14738	304 (0,7%)	8862 (60,1%)
Sergio	640x480	42250s	23937	333 (0,79%)	22300 (93,2%)
CPD	1280x720	44568s	13025	446 (1,0%)	8669 (66,5%)
Servidor	1280x720	28498s	25660	295 (1,0%)	25144 (97,9%)

Tabela I – Comparação de resultados de reconstruções com utilização todos os *frames* e apenas com os *keyframes* selecionados pelo *software* desenvolvido

No melhor resultado obtido nos experimentos de reconstrução com [8] obteve-se uma redução de 94% no número de *frames* e no pior caso, uma redução de 75%.

Em relação ao tempo de processamento com [9], necessitou-se de 0,7% do tempo no melhor caso e 3,6% do tempo no pior caso, comparando com a reconstrução com todos os *frames*.



Figura 4 – Reconstrução obtida da captura de uma pessoa (objeto Sérgio), com destaque para a trajetória da câmera. Neste experimento utilizou-se o Autodesk 123 Catch [8].

A manutenção das características das reconstruções dos objetos permite sua utilização na construção de cenários virtuais para jogos. Quando reconstruídos apenas com os *keyframes* selecionados pelo protótipo observou-se, ainda, um ganho em tempo de processamento, em relação à reconstrução com utilização de todos os *frames*.

Na Figura 5 são apresentados alguns pontos de vista do objeto esfinge, reconstruído nos testes.

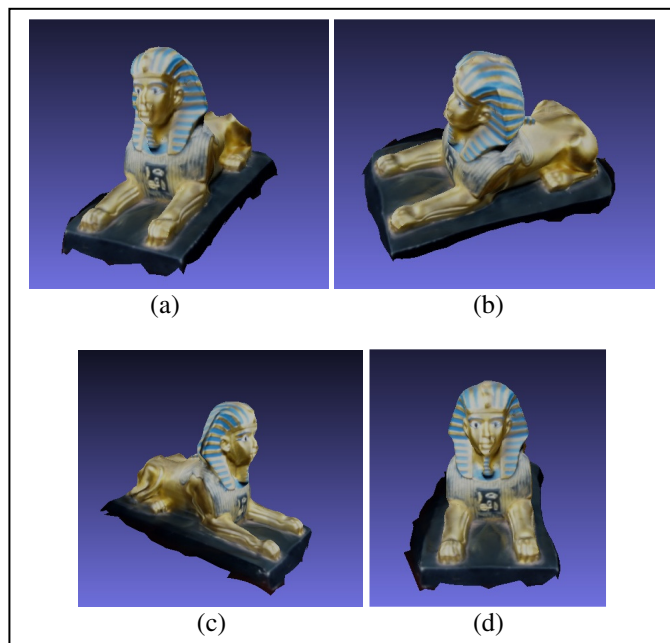


Figura 5 – Objeto Esfinge em diversos pontos de vista

CONCLUSÕES E TRABALHOS FUTUROS

A partir do trabalho apresentado, é possível concluir que o protótipo do *software* de extração automática de *keyframes* é viável para reconstruções que utilizam o método SFM.

Os resultados mostraram uma efetiva redução de esforço computacional, se comparados com a utilização de todos os *frames* do vídeo.

A diferença de número de pontos não foi tão significativa no resultado final, onde o objeto recriado manteve todas suas características inalteradas.

Os testes realizados mostraram que a pré-seleção na escolha dos *frames* antes de realizar uma reconstrução resulta em objetos com menores erros em sua malha, como deformações ou mescla do objeto reconstruído com outros elementos presentes no fundo ou abaixo do objeto.

Também verificou-se que há uma melhoria na malha obtida, quando são analisadas as informações de borrimento e diferença de *pixels* para proporcionar um salto de *frames* eficiente.

Obtém-se, assim, maiores taxas de acertos em comparação com escolhas manuais feitas pelo método tradicional ou com a utilização de todos os *frames*.

Destacam-se, como trabalhos futuros:

- a elaboração de um método automático para o cálculo dos limiares de borrimento e diferença de *pixels*;
- a realização da comparação com o método proposto em [10];
- melhorias nos filtros de borrimento, onde o objeto de interesse deve ser isolado do restante da imagem;
- análises de outras características existentes nas demais etapas do método SFM.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] VAN DENBERGH, F.; LALIOTI, V. *Software* chroma keying in an immersive virtual environment. In South African Computer Journal n 24 p 155-162, África do Sul, 1999.
- [2] GRAU, O.; PRICE, M.; THOMAS, G. A. Use of 3-D Techniques for Virtual Production. BBC Research & Development, White Paper WHP 033, Londres, 2002.
- [3] POLLEFEYS, Marc. Visual 3D Modeling *from* Images. Tutorial Notes, University of North Carolina. Chapel Hill, USA, 2002.
- [4] SNAVELY, S.; SEITZ, M.; SZELISKI, R. Skeletal graphs for efficient *Structure from motion*. In CVPR, p 1–8, 2008..
- [5] AGARWAL, S.; FURUKAWA, Y.; SNAVELY, N; CURLESS, B; SEITZ, S.; SZELISKI, R. Reconstructing Rome. IEEE Computer, p. 40-47, Junho, 2010.
- [6] FULTON, John R. Sensor Orientation in Image Sequence Analysis. 299f. Tese (Mestrado em Filosofia) Melbourne University. Melbourn, 2007.
- [7] CIGNONI, P.; CALLIERI, M.; CORSINI, M.; DELLEPIAINE; GANOVELLI, F.; RANZUGLIA, G. MeshLab: an open-source mesh processing tool. Eurographics Italian Chapter Conference. V, Sacrano R. De Chiara, U. Erra (Eds.). Eurographics Association, pp. 129-136. Italia, 2008.
- [8] AUTODESK. 123D Catch. Disponível em: <<http://www.123dapp.com/catch>>. 2012. Acesso em: 18 agosto 2012.
- [9] WU, Changchang. VisualSFM: A Visual *Structure From Motion* System. Disponível em <<http://homes.cs.washington.edu/~ccwu/vsfm/>>. 2011. Acesso em 18 de agosto de 2012
- [10] SEO, Y.; KIM, S.; DOO, K.; CHOI, J. Optimal keyframe selection algorithm for three-dimensional reconstruction in uncalibrated multiple images. Opt. Eng. 0001;47(5):053201-053201-12.