Creating Virtual Models from Uncalibrated Camera Views

Felicia Brisc, Paul Whelan
School of Electronic Engineering
Dublin City University
{briscf, whelanp}@eeng.dcu.ie

Abstract

The reconstruction of photorealistic 3D models from camera views is becoming an ubiquitous element in many applications that simulate physical interaction with the real world. In this paper, we present a low-cost, interactive pipeline aimed at non-expert users, that achieves 3D reconstruction from multiple views acquired with a standard digital camera. 3D models are amenable to access through diverse representation modalities that typically imply trade-offs between level of detail, interaction, and computational costs. Our approach allows users to selectively control the complexity of different surface regions, while requiring only simple 2D image editing operations. An initial reconstruction at coarse resolution is followed by an iterative refining of the surface areas corresponding to the selected regions.

I.3.7: [Computer Graphics]: Three-Dimensional Graphics and Realism, I.4.5: [Image processing and computer vision]: Reconstruction

1. Introduction

Image-based modeling and rendering (IBMR) techniques have emerged in recent years as the alternative of choice for synthesizing photo-realistic 3D models. In this paper, we present a versatile IBMR pipeline for generating customizable virtual models from images acquired with a standard digital camera. Our main objective is to provide non-expert users with a low-cost, interactive tool for building 3D models compatible with applications ranging from computer games to virtual worlds and augmented reality.

With the widespreading of distributed and networked applications, 3D models are amenable to access through diverse representation modalities that typically imply trade-offs between level of detail, interaction and scalability. Since perceptual importance is determined ultimately by the human factor, our approach [Bri04] enables users to control the relative complexity of different surface regions, while requiring only common image editing operations. Our method exploits a *Space Carving* technique and recasts it's solution as an efficient process for creating multi-resolution 3D models.

Space Carving approaches [Dye01, SCM*03] have proven to be a strong alternative to traditional correspondence-based methods due to their flexible visibility models and explicit handling of occlusions. Space Carving methods process the 3D scene in a way that resembles the work of an artist sculpting a raw block of marble. The space in which the scene occurs is represented through a tesselated volume of voxels and occupancy decisions are made about whether a volumetric element belongs to the objects in the scene. The decision mechanism is the *Photo-Consistency Criterion* [SD97, CMS99, KS99], consisting in a color similarity test of the voxels. The resulting *photo hull*, represents the union of all possible photo-consistent scene reconstructions.

The voxels need to be traversed in a monotonic order during reconstruction for a correct visibility handling. The *Space Carving* algorithm introduced by Kutulakos and Seitz [KS99] evaluates one plane of voxels at a time, and performs multiple scans, typically along the directions of the three axes. Our approach builds on the *Generalized Voxel Coloring* algorithm (GVC) presented in [CMS99]. GVC simply iterates over every border voxel, providing a two-way mapping between surface voxels and image pixels.

2. Camera Self-calibration

The input to our system is a sequence of uncalibrated images of a scene acquired with a single moving digital

camera, so that we need to perform self-calibration prior to the 3D reconstruction in order to recover the camera intrinsic and extrinsic parameters. The 3D reconstruction pipeline is outlined in Figure 1.

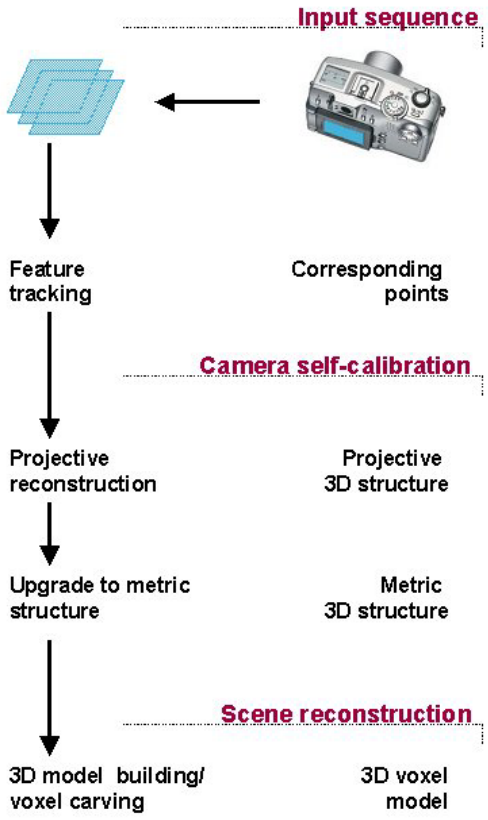

Figure 1. 3D reconstruction pipeline

We use a stratified approach to recover both camera and scene structure similar to methods presented in [PKG99, HK00]. One advantage of the employed approach is that it allows to recover an Euclidean reconstruction of the scene without any initial solution and amounts to solving only linear systems. Also, it allows the use of varying focal length throughout the sequence, so that the user can perform zoom in/out operations.

First, a number of relevant features are selected manually in a reference view, then their corresponding points are tracked throughout the sequence.

In the initial stage, a projective reconstruction is computed from the set of correspondences, followed by an upgrade to Euclidean (metric) structure by enforcing metric constraints on the intrinsic camera parameters [HZ00].

Under the pinhole camera model, the projection of a 3D point $\mathbf{X}_j = [x_j, y_j, z_j, 1]^T$ to view i in homogeneous coordinates is obtained as:

$$\lambda_{ij} \begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} = P_i \mathbf{X}_j \quad (1),$$

where $[u_{ij}\ v_{ij}\ 1]^T$ are the pixel image coordinates, λ_{ij} is an arbitrary scale factor (projective depth), and P is the 3x4 projective camera matrix, encoding both its intrinsic and extrinsic parameters.

Equation (1) for m views and n tracked points can be combined into one matrix equation :

$$\mathbf{W}_s = \begin{bmatrix} \lambda_{11}\begin{bmatrix} u_{11} \\ v_{11} \\ 1 \end{bmatrix} & \cdots & \lambda_{1m}\begin{bmatrix} u_{1m} \\ v_{1m} \\ 1 \end{bmatrix} \\ \vdots & \vdots & \vdots \\ \lambda_{n1}\begin{bmatrix} u_{n1} \\ v_{n1} \\ 1 \end{bmatrix} & \cdots & \lambda_{nm}\begin{bmatrix} u_{nm} \\ v_{nm} \\ 1 \end{bmatrix} \end{bmatrix} =$$

$$\begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} [\mathbf{x}_1 \ldots \mathbf{x}_m] = \mathbf{PX} \quad (2)$$

where :

$\mathbf{W}_s$ is the $3n \times m$ scaled measurement matrix

$\mathbf{P}$ is the $3n \times 4$ perspective matrix

$\mathbf{X}$ is the $4 \times m$ shape matrix.

Ideally, $\mathbf{W}_s$ should be a rank-4 matrix, so that a rank-4 factorization of it produces a projective reconstruction of the points. However, in reality, due to noise and measurement errors its rank will be different and the rank-4 constraint has to be enforced.

On the other hand, Equation (2) holds only if the correct scale factors λ_{ij} are applied to each of the measured points $\mathbf{x}_{ij}$. In order to fulfill both requirements, a rank-4 factorization needs to be applied on $\mathbf{W}_s$ until the recovered projective depths make Equation (2) consistent. We are employing an iterative factorization approach where the projective depths are rescaled at each iteration to give a closer rank-4 approximation of $\mathbf{W}_s$ [Che00].

The factorization of Equation (2) is not unique, motion and shape are recovered only up to a projective transformation. The next stage is concerned with the upgrade to metric structure, which is reduced to the recovery of a rectifying transformation, called the *Projective Distortion Matrix*, that removes the projective ambiguity. Our approach is computationally equivalent to the recovery of the *Absolute Quadric* method proposed in [Tri97].

In the absence of any additional information, some assumptions need to be made, translating to constraints on the internal camera parameters:

- principal point is at the center of the image plane
- zero skew of the pixels
- aspect ratio equal to 1

These assumptions leave only the focal length as a variable parameter, and yield four equations from each view. The self-calibration equations combine to an overdetermined linear system of $4 \times m$ equations with a unique solution for $m \geq 3$.

3. Region Labelling

For selective surface refining, the user delimits polygonal regions (e.g. corresponding to salient features) in one or more images using familiar selection tools, such as polylines and scissors, and assigns them a label ID corresponding to the chosen resolution. Adjacent polygons must intersect along common boundaries.

4. Volumetric Multi-resolution 3D Reconstruction

After determining the camera views positions and parameters, the next step is the 3D reconstruction of the scene through a volumetric, voxel-based method.

The reconstruction is initiated with a volume containing the space in which the scene occurs, determined by upscaling the spatial bounds of the 3D points recovered during self-calibration. This volume is discretized into a 3D lattice of voxels, the goal of the carving algorithm being the determination of the voxels representing the surfaces in the scene.

We are considering a point voxel projection, i.e. only the voxel center is projected to the input images, leading to a single pixel in each view. The algorithm removes, or carves the empty space voxels with the help of the photo-consistency criterion. In order to be photo-consistent , a voxel modeling the shape surface has to project to similar, or consistent colors in each camera view its visible from.

The determination of voxel visibility is essential to compute photo-consistency, otherwise voxels that do belong to the scene surface could erroneously be declared inconsistent.

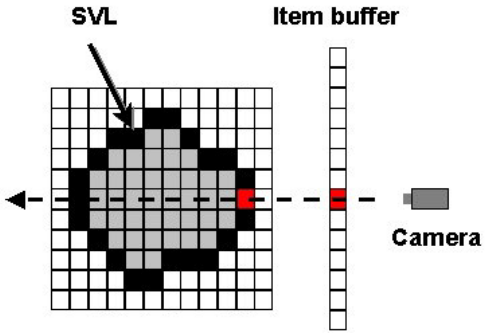

Figure 2. The item buffer records for each pixel the ID of the closest visible voxel that projects onto it

After assigning each voxel an unique ID, we maintain two data structures that provide a bidirectional voxel-pixel mapping and a correct visibility handling : the surface voxel list (SVL), and the item buffer (IB) [CMS99]. The SVL contains a list of the actual consistent voxels situated on the surface of the set of uncarved voxels, while the item buffers (IBs) store for each view the ID of the closest visible voxel.

The SVL is initialized with the outside layer of voxels of the bounding box. Carved voxels are removed from the SVL, while adjacent uncarved voxels which become visible are added to the SVL. The item buffer is computed for each image by performing a sequential scan of the SVL in order to identify all the pixels that a voxel $\mathbf{V}$ projects onto. If the distance from the camera to $\mathbf{V}$ is less than the distance recorded for the pixel, then the pixel's stored distance and voxel ID are over-written with those of $\mathbf{V}$ (Figure 2).

The set of pixels $vis(\mathbf{V})$ from which a voxel is visible is determined by projecting each voxel on all views and comparing its ID to the ID stored in the item buffer for the respective pixels. If the two ID values are equal, the pixel is added to the set $vis(\mathbf{V})$. If $\mathbf{V}$ is visible from a pixel belonging to a labeled region , it is flagged and the label ID is stored

The color-consistency check is done by computing the standard deviation σ of the normalized color components $c_1, \ldots, c_k$ of the pixels from the set $vis(\mathbf{V})$. The normalization of the colors by the sum of components increases the robustness in respect to varying illumination conditions between the different images. The voxel is consistent if $\sigma < \tau$, where τ is a predefined threshold. Voxels found to be consistent are assigned the mean value of the color components, while inconsistent voxels are carved.

The item buffers need to be recomputed periodically, because carving a voxel changes the visibility of the remaining uncarved voxels (Figure 3). Voxel carving completes when every voxel is found to be color-consistent, the remaining surface voxels represent the 3D shape of objects in the scene.

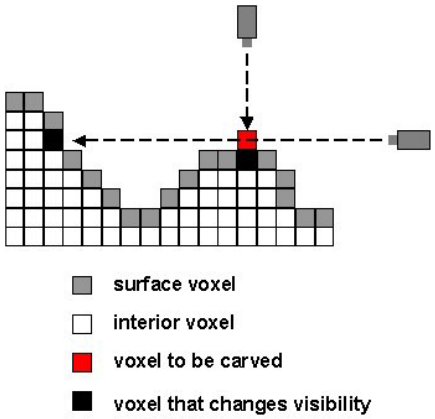

Figure 3. Voxels that change visibility

First we perform a coarse reconstruction, in order to isolate and differentiate the voxel groups that project to labeled regions. The 3D bounds of each voxel group are computed and a spatial constraint grid is applied, which restricts further refining to labeled voxels. The resolution is increased by subdividing each voxel into eight voxels [PD98]. Next, voxel carving is performed on the higher resolution voxels. The above steps are repeated iteratively until the required resolution is obtained.

5. Results

Figure 4 shows an input sequence of five images, with a human subject with placed markers and 63 tracked correspondences. The left image in Figure 5 shows the selected corresponding points, while the right image shows the metric structure of the correspondences, as well as the camera positions recovered during the self-calibration preceding the volumetric reconstruction. The left image in Figure 6 shows the 3D shape reconstructed at resolution r=25. With the face area of the subject selected for refinement, we performed the algorithm for two resolution increases, resulting in a final resolution r=6. The multi-resolution reconstruction is shown in the right image of Figure 6. Figure 7 presents detail views of the above reconstructions.

6. Conclusion and Future Work

We presented a complete pipeline for reconstructing 3D scenes from a set of camera views. Our system reduces considerably the computational overhead caused by high-resolution processing, allowing users to detail only features significant to their perception with simple image editing operations.

Currently we assume that the reconstructed scenes are Lambertian, a simplifying but limiting assumption commonly made in reconstruction algorithms. However, real surfaces interact with light in complex ways, producing view-dependent effects such as specularities and reflections. Our future investigations will focus on more sophisticated modeling of the bidirectional reflectance distribution function in order to improve the flexibility of the reconstruction algorithm. [MKZ*01, THS04].

7. Acknowledgements

The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

Figure 4. The 5-image input sequence

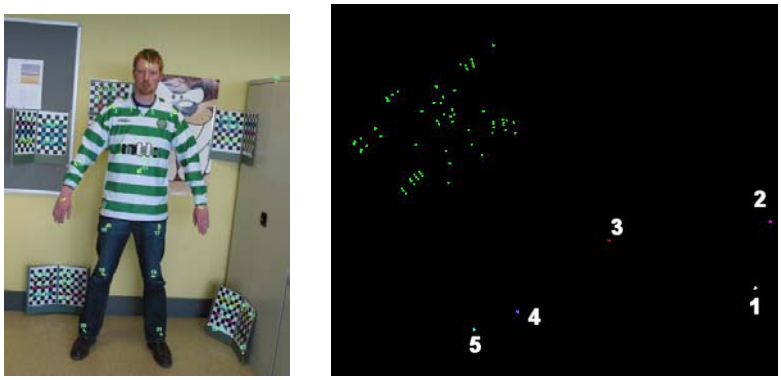

Figure 5. Left: A sequence image with the tracked points.

Right: The recovered metric structure of the tracked points and the camera positions

Figure 6. Left : the reconstructed human model at resolution *r*=25.

Right : same 3D model with the face region refined at resolution *r*=6.

Figure 7. Detail views of the above left and right images, respectively

References

[Bri04] Brisc F., "Multi-resolution Volumetric Reconstruction Using Labeled Regions", in *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, Lake Tahoe, USA, 2004.

[Che00] Chen Q., "Multi-view Image-Based Rendering and Modeling", Ph.D. thesis, University of Southern California, 2000.

[CMS99] W. B. Culbertson, T. Malzbender, G.Slabaugh, "Generalized voxel coloring", *International Workshop on Vision Algorithms*, Corfu, Greece, Springer Verlag Lecture Notes on Computer Science, pp. 100-115, 1999.

[Dye] C. Dyer, "Foundations of Image Understanding", chapter Volumetric Scene Reconstruction from Multiple Views, pg. 469-489. Kluwer, Boston, 2001.

[HZ00] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision", Cambridge University Press 2000.

[HK00] M. Han and T. Kanade, "Creating 3D models with uncalibrated cameras", *IEEE Computer Society Workshop on the Application of Computer Vision, (WACV2000)*, 9(2), pp. 137-154, 2000.

[KS99] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," in *Proc. 7th Int. Conf.Computer Vision*, pp. 307–314, 1999.

[Kut00] K. N. Kutulakos, "Approximate N-view stereo," *Proceedings of the European Conference on Computer Vision*, Springer Lecture Notes in Computer Science 1842, Vol. 1, pp. 67-83, June/July 2000.

[MKZ*01] S. Magda, D. Kreigman, T. Zickler and P. Belhumeur "Beyond Lambert: Reconstructing Surfaces with Arbitrary BRDFs," *Proceedings of International Conference on Computer Vision*, vol. 2, pp. 391–398, 2001.

[PKG99] M. Pollefeys, R. Koch and L. van Gool, "Self-Calibration and Metric Reconstruction Inspite of Varying and Unkonwn Intrinsic Camera Parameters," *International Journal of Computer Vision*, vol. 32, pp. 7–25, Jan. 1999.

[PD98] A.Prock and C. Dyer, "Towards real-time voxel coloring", *Image Understanding Workshop*, 1998, pp. 315-321.

[SD97] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," in *Proc.Computer Vision and Pattern Recognition Conf.*, pp. 1067–1073, 1997.

[SK98] S. M. Seitz , K. N. Kutulakos, "Plenoptic image editing", *Proc. Fifth International Conference on Computer Vision*, pp. 17-24, 1998.

[SCM*03] G. Slabaugh, W. B. Culbertson, T. Malzbender, M. R. Stevens and R. W. Schafer, "Methods for Volumetric Reconstruction of Visual Scenes," *International Journal of Computer Vision*, 2003.

[SSH02] G. Slabaugh, R. W. Shafer, M. C. Hans, "Multi-resolution Space Carving Using Level Set Methods", *Proc. International Conference on Image Processing (ICIP),* 2002.

[Tri97] B. Triggs, "Autocalibration and the absolute quadric", in *Proc. ComputerVision and Pattern Recognition*, 1997.

[THS04] A. Treuille, A. Hertzmann and S. M. Seitz, "Example-Based Stereo with General BRDFs", *ECCV,* 2004.