3D Modelling from video

Svetlana Mijakovska[1], Igor Nedelkovski[2]

Abstract – In this paper the research on the 3D modelling from video is presented. We give overview of 3D modeling from video, especially the second step (structure and motion recovery) and the goal is to find the best algorithm for finding and fitting features to create a 3D model from multiple view of images.

Will be considered suitable algorithms to describe the facilities, whether used triangular polygonal mesh etc. cloud of points to describe the exterior of the building.

Keywords – **3D Modelling, 3D Model, Video, Epipolar geometry, Structure from Motion (SfM), RANSAC, MLESAC, MSAC, Least Square.**

I. Introduction

The rapid increase in recent years of the graphics processing capabilities of even relatively modestly priced personal computers has lead to the widespread use of 3D graphics. Complex and large–scale 3D models are commonly used in areas such as animation, computer games and virtual reality. Currently most 3D models are created manually by graphic artists making it a time–consuming, and therefore expensive, process. If the model to be created has no real–world counterpart then there is little choice but to design it by hand. However, in many cases the aim is to create a model of an actual scene or object and, in this case, it is obviously highly desirable to create a process whereby the model may be automatically acquired. The increase in availability of high quality, consumer–level, digital video and still cameras means that the capability to capture high–resolution digital images and subsequently perform processing on them is now within the reach of most people.

The automatic recovery of three dimensional structure from video footage of scenes has been a long–standing area of research in computer vision. This problem, known as Structure from Motion (SfM), involves trying to recover, solely from the sequence of images, the 3D structure of a scene and the position and orientation (pose) of the camera at the moment each image was captured.

Applications for SfM can be broadly split into two categories, those that require geometric accuracy and those that require photorealism:

Geometric accuracy: These types of application are generally less concerned with the visual appearance of the model but require the scene structure and camera motion to be reconstructed with a high degree of accuracy. Robot navigation, for instance, requires high–accuracy models, but the visual appearance of the model is unimportant. The reverse engineering of existing objects for use in CAD requires the structure of the object to be recovered with a high degree of accuracy. Film special effects that place computer–generated objects into the film and other ‘augmented reality’ applications require the camera motion to be very accurately reconstructed but the appearance of the structure is irrelevant as it is never seen in the finished product.

Photorealism: In contrast, there are a growing number of situations geometric accuracy of the underlying reconstruction is less important as long, as, for the purposes of the application the model visually resembles the real scene. This is the case for applications such as virtual reality, simulators, computer games and special effects that require a virtual set based on a real scene [1].

In computer vision, several systems have been developed to automatically recover a cloud of 3D scene points from a video sequence (e.g. [Pollefeys et al. 2004]). However these are vulnerable to ambiguities in the image data, degeneracies in camera motion, and a lack of discernible features on the model surface. These difficulties can be overcome by manual intervention in the modelling process. In the extreme case, a modelling package such as Blender3D can used to build a model manually, but it is difficult and time consuming to create a photorealistic result by this process. A more appealing option is to use all of the information that can be derived from the video using computer vision techniques to inform and accelerate an interactive modelling process [2].

II. Overview of 3D Reconstruction from Video Sequences

Main tasks of 3D reconstruction are:

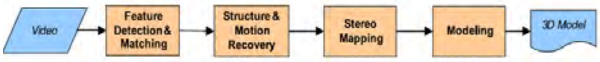


Figure 1: Main tasks of 3D reconstruction

The 3D reconstruction can be divided into 4 main tasks (Figure 1), which are discussed in the following sections:

1. Feature detection and matching: The objective of this step is to find out the same features in different images and match them.

[1] Svetlana Mijakovska is with the Technical Faculty of Bitola, address: ul. Ivo Ribar Lola bb, 7000 Bitola, Macedonia, e-mail: svetlanamijakovska@gmail.com

[2] Igor Nedelkovski is with the Technical Faculty of Bitola, address: ul. Ivo Ribar Lola, 7000 Bitola, Macedonia, e-mail: igor.nedelkovski@uklo.edu.mk.

2. Structure and Motion Recovery: This step recovers the structure and motion of the scene (i.e. 3D coordinates of detected features; position, orientation and parameters of the camera at capturing positions).

3. Stereo Mapping: This step creates a dense matching map. In conjunction with the structure recovered in the previous step, this enables us to build a dense depth map.

4. Modeling: This step includes procedures needed to make a realistic model of the scene (e.g. building mesh models, mapping textures).

Feature detection and matching (Fig.3) is process that detects and match features in different images. Video sequence is created of more images so in this step we must find interested points (point feature), i.e. detectors and descriptors.

The most important information a detector gives is the location of features, but other characteristics such as the scale can also be detected. Two characteristics that a good detector needs are repeatability and reliability. Repeatability means that the same feature can be detected in different images. Reliability means that the detected point should be distinctive enough so that the number of its matching candidates is small.

A descriptor is a process that takes information of features and image to produce descriptive information i.e. features description which are usually presented in form of features vectors. The descriptions then are used to match a feature to one in another image. A descriptor should be invariant to rotation, scaling, and affine transformation so the same feature on different images will be characterized by almost the same value and distinctive to reduce number of possible matches.

The second task Structure and motion recovery recovers the structure of the scene and the motion information of the camera. The motion information is the position, orientation, and intrinsic parameters of the camera at the captured views. The structure information is captured by the 3D coordinates of features. Because the fact that video sequence is created of more images, for this step we must research 3D reconstruction from multiple views i.e. multiple view geometry.

For the calibrated case, the essential matrix E [3] is used to represent the constraints between two normalized views. Given the calibration matrix K (a 3x3 matrix that includes the information of focal length, ratio, and skew of the camera),
the view is normalized by transforming all points by the inverse of K: $\hat{x} = K^{-1}x$, in which x is the 2D coordinate of a point in the image. The new calibration matrix of the view is now the identity. Then with a corresponding pair of points (x, x') in homogeneous coordinates, E is defined by a simple equation: $\hat{x}`^{T}E\hat{x}$ = 0.

The research has later been extended to the uncalibrated case. During the 1990s, the concept of fundamental matrix F was introduced and well-studied by Faugeras [4] and Hartley [5]. The F matrix is the generalization of E and the defining equation is very similar: $x`^{T}Fx = 0$.

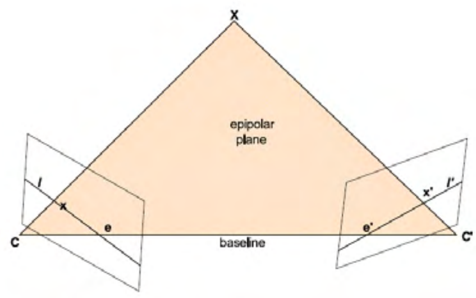


Fig.1 Two-view geometry

Three-view geometry is also developed during the 1990s. The geometry constraints are presented by trifocal tensors that capture relation among projections of a line on three views. The trifocal tensor defines a richer set of constraints over images (Fig.2). Apart of a line-line-line correspondence, it also defines point-line-line, point-line-point, point-point-line, and point-point-point constraints. Furthermore, it introduces the homography to transfer points between two views.

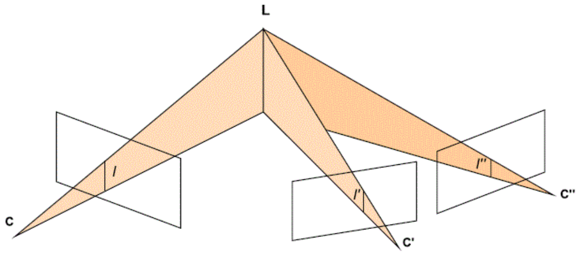


Fig.2 Line correspondence among three view - basis to define trifocal tensors

Reconstruction with only knowledge of feature correspondences is only possible up to a projective reconstruction and there are many ways to obtain projection matrices from a geometry constraint, i.e. a fundamental matrix or a focal tensor. Hence projective reconstruction is mainly the recovery of fundamental matrices or focal tensors. Methods, implementation hints, and evaluations are well discussed by Hartley and Zisserman in [6]. If the input, i.e. feature correspondences, includes outliers, robust methods such as RANSAC, LMS can be employed to reject them.

Stereo mapping task can be divided into two sub tasks: rectification and dense stereo mapping. The first one exploits the epipolar constraint to prepare the data for the second one by aligning a corresponding pair of epipolar lines along the same scan line of images thus all corresponding points will have the same y-coordinate in two images. This makes the second task, roughly search and match over the whole image, faster.

The final step is to map texture on the model. Triangulation is quite a simple task. Points of each stereo map are triangulated to generate depth maps. Those maps are used to construct the mesh of the scene and finally, with texture extracted from frames, the complete textured model can be built.

III. Structure and Motion Recovery

This step is actually the main step in 3D modeling from video, because in this step we must choose which algorithm to be used for find corresponding points of two images or more images with moving cameras at different points in time, with moving objects using different methods such as feature matching and block matching. We are research RANSAC, Least Squares, MSAC and MLESAC.

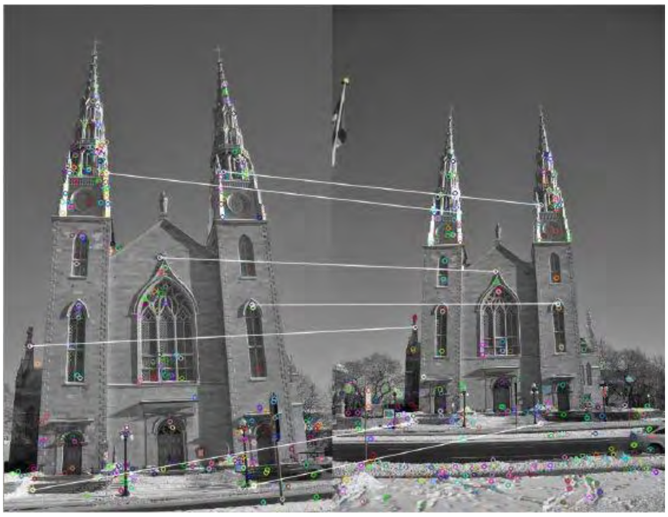


Fig.3 Feature matching

RANdom SAmple Consensus algorithm is:

- introduced Fischler and Bolles in 1981
- iterative method
- non deterministic

1. randomly select smallest possible subset of data (hypothetical inliners) an create model
2. test data against model, expand hypothetical inliners with all points inside a threshold
3. reestimate model with all points supporting the model
4. repeat and keep models with most support

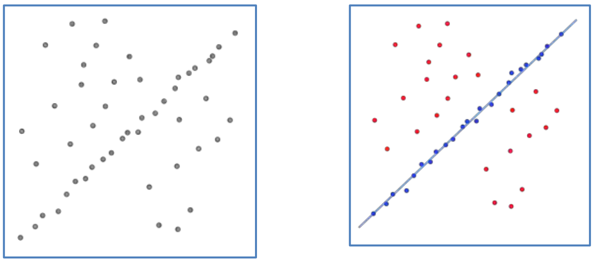
RANSAC algorithm is method to estimate the parameters of a certain model1 starting from a set of data contaminated by large amounts of outliers of a model using datasets containing more than 50% of outliers. A datum is considered to be an outlier if it will not fit the "true" model instantiated by the "true" set of parameters within some error threshold that defines the maximum deviation attributable to the effect of noise. [7]

Despite many modifications, the RANSAC algorithm is essentially composed of two steps that are repeated in an iterative fashion (hypothesize and test framework):

• **Hypothesize**. First minimal sample sets (MSSs) are randomly selected from the input dataset and the model parameters are computed using only the elements of the MSS. The cardinality of the MSS is the smallest sufficient to determine the model parameters (as opposed to other approaches, such as least squares, where the parameters are estimated using all the data available, possibly with appropriate weights).

• **Test**. In the second step RANSAC checks which elements of the entire dataset are consistent with the model instantiated with the parameters estimated in the first step. The set of such elements is called consensus set (CS).

RANSAC terminates when the probability of finding a better ranked CS drops below a certain threshold. In the original formulation the ranking of the CS was its cardinality (i.e. CSs that contain more elements are ranked better than CSs that contain fewer elements).



Data with outliers

Line obtained with RANSAC, no influence of the outliers. .

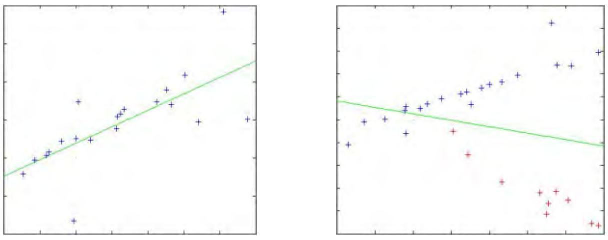
Fig.4 Example of line obtained with RANSAC algorithm without influence of outliers.

The benefits of RANSAC are:

- only takes into account the number of inliers
- RANSAC minimizes cost.

Least Squares

- Calculate parameters of model function
- Overdetermined data set
- Minimize sum of squared residuals



Least squares without outliers

Least squares with outliers.

Fig.5 Example of line obtained with Least Squares

MSAC - M-estimator SAmpling Consensus use score for inliners.

MLESAC - Maximum Likelihood Estimation SAmple Consensus

The MLESAC algorithm is an example of RANSAC that uses different cost function than the cardinality of the support. The algorithm was introduced by Torr and Zisserman [8] and further improvements were made by Tordoff and Murray [9]. Instead of maximizing the support of the model, the likelihood of the model is maximized. The error distribution is represented as a mixture of inlier and outlier distributions.

IV. CONCLUSION

The process of 3D modeling over the four main steps: feature extraction and matching, structure and motion recovery, stereo mapping, and modeling. Each step or even sub-step is already a field of research. The goal of this paper is to give overview of 3D modeling from video, especially the second step (structure and motion recovery) and to find the best algorithm for finding and fitting features to create a 3D model from multiple view of images. Of the process of choose the algorithm and its testing, we can define as:

- ❖ correspondence problems provide environments with high number of outliers
- ❖ least squares fails in these environments
- ❖ RANSAC provides significant improvement in presence of high numbers of outliers
- ❖ Performance can be additionally improved by using more complex error models
  _ Counting (RANSAC)
  _ Square distance (MSAC)
  _ Negative log likelihood (MLESAC).

REFERENCES

[1] Cooper, O., Campbell, N., Gibson, D. 2003. Automated meshing of Sparse 3D Point Clouds. In Proceedings of the SIGGRAPH 2003 conference on Sketches and Applications, San Diego.

[2] Gibson, S., Hubbold, R., Cook, J., and Howard, T. 2003. Interactive reconstruction of virtual environments from video sequences, Computers & Graphics 27, 2 (April), 293–301.

[3] H.C. Longuet Higgins. A computer algorithm for reconstructing a scene from two projection. Nature, 1981.

[4] S. Maybank O.D. Faugeras, Q. Luong. Camera self-calibration: Theory and experiment. European Conference on Computer Vision, 1992.

[5] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras, Lecture Notes In Computer Science, 588, 1992.

[6] R. Hartley and A. Zisserman. Multiple view geometry in computer vision 2nd edition. Cambridge University Press, 2004.

[7] Overview of the RANSAC Algorithm, Konstantinos G. Derpanis, Version 1.2, May 13, 2010.

[8] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. CVIU, 78:138–156, 2000.

[9] B. Tordoff and D.W. Murray. Guided sampling and consensus for motion estimation. In Proc. 7th ECCV, volume 1, pages 82–96. Springer-Verlag, 2002.

[10] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381–395, 1981.

[11] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. University Press, Cambridge, 2001.

[12] P. Torr and C. Davidson. IMPSAC: A synthesis of importance sampling and random sample consensus to effect multi-scale image matching for small and wide baselines. In European Conference on Computer Vision, pages 819–833, 2000.

[13] P. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding, 78(1):138–156, 2000.

`