AUTOMATIC EDGE MATCHING ACROSS AN IMAGE SEQUENCE BASED ON RELIABLE POINTS

Yixiang Tian [a, b, *], Markus Gerke [a], George Vosselman [a], Qing Zhu [b]

[a] International Institute for Geo-Information Science and Earth Observation (ITC), Hengelosestraat 99, P.O.Box 6, 7500AA, Enschede, the Netherlands - (ytian, gerke, vosselman)@itc.nl

[b] State Key Lab of Information Enginneering in Surveying Mapping and Remote Sensing, Wuhan University, 129 Luo Yu Road, Wuhan, Hubei, 430079, P.R. China - zhuq66@263.net

ICWG III/V

KEY WORDS: Image sequence, Video, Edge, Matching, Analysis, Reconstruction

ABSTRACT:

This paper presents a new method for matching edges across a video image sequence. The method can deal with uncalibrated images acquired with a hand held camera. Compared to previous work, the method employs geometric constraints between edges based on reliable matched points, which reduces the search space for corresponding 2D edges in the frames. The 3D edge parameters are estimated from these matched 2D edges by using the Gauss-Markoff model with constraints. End points of each 3D edge are found by analyzing the end points of the corresponding 2D edges. The results show that the developed algorithms are able to efficiently and accurately reconstruct 3D edges from image sequences.

1. INTRODUCTION

Modelling 3D objects and scenes from image sequences is a research topic since several years [Baltsavias, 2004; Pollefeys, 2004; Remondino and EL-Hakim, 2006]. The high overlapping of images within a video sequence lead to highly redundant information, which is exploited for tie point extraction, feature tracking and 3D object extraction. However, the short baseline between the images also leads to a poor ray intersection geometry and thus to mismatches across the sequence.

At the same time, using edge information for the reconstruction of man-made objects from images has been concerned by researchers from the fields of photogrammetry and computer vision for a long time [Hartley and Zisserman, 2000]. As the point clouds obtained from applying feature and camera tracking steps to a video sequence are not dense enough, which do not allow a complete description of 3D scene and not all import points for object reconstruction, such as corner points, can be extracted. Edge features can provide more constraints about objects' shape than point features. Existing approaches for edge detection can obtain acceptable result [Canny, 1986; Meer and Georgescu, 2001]. Nevertheless, edge matching is still a difficult problem for several reasons. One is that edges belonging to the same entity in object space are often extracted incompletely and inaccurately in the single images of the sequence. Sometimes, an ideal edge might be broken into two or more small segments that are not connected to each other. Further, the end points are not reliable, and even with a correct orientation, it is difficult to build up topological connections between edges. A second reason for the complexity of edge matching is due to the fact that there is no strong disambiguating geometric constraint available over two or more views during edge matching. There is only a weak overlap constraint for edge segments of finite length arising from applying epipolar geometry constraint to end points [Schmid and Zisserman, 1997; Baillard et al., 1999] .

Existing approaches to edge matching in the literature are generally categorized into two types. One is matching individual edges between images based on a similarity measure. The similarity measure is based on the comparison of edge attributes, such as orientation, edge support region information. The other strategy is structural matching, which considers more geometrical and topological information among edge features. But this kind of methods often have a high complexity and they are sensitive to error in the segmentation process[Armstrong and Zisserman, 1995; Baillard et al., 1999; Kunii and Chikatsu, 2004; Klein and Murray, 2006]. When camera projection information is available, matching individual edges can get precise and efficient results and reduce the complexity and computation time.

Most edge matching methods are based on stereo or triplet image pairs, and the results are merged together if there are more images [Baillard et al., 1999; Zhang et al., 2005]. How to use redundant information from video image sequence for edge matching and how to eliminate errors caused by short base line when reconstructing 3D edge geometry are the main task of this paper. All the problems mentioned above are considered in this method. Each step considered in this method will be explained in the following paragraphs.

In section 2 our preprocessing steps on feature extraction are described. The main method is explained in section 3, divided to four parts (overview, point quality analysis, 3D edge estimation and end points decision). Results are shown in section 4 and discussed in section 5, which also indicates some further work.

2. PREPROCESSING

2.1 Feature Presentation

* Corresponding author.

Features in this paper are points and edges in 2D and 3D. Points in 2D and 3D are represented as homogenous vector $\mathbf{x}=(x,y,1)^T$ and $X=(X,Y,Z,1)^T$. 2D edges use angle-distance form $\mathbf{l}=(\cos(\theta),\sin(\theta),-d)^T$, and 3D edges are presented by Plücker coordinates similar to [Hartley and Zisserman, 2000; Heuel, 2004], $\mathbf{L} = (L_1, L_2, L_3, L_4, L_5, L_6)^T$, where the homogeneous part $L_h = (L_1, L_2, L_3)^T$ is constrained to be orthogonal to the Euclidean part $L_o = (L_4, L_5, L_6)^T$, i.e. $L_h^T L_o = 0$. The homogeneous part presents the edge direction and the Euclidean part decides the distance from the origin to the edge. Thus the 6-vector **L** has 4 degrees from freedom, considering both the orthogonal and homogeneous constraint.

2.2 Point and Camera Parameters Extraction

Usually when dealing with video image sequences, this step (point detecting and matching) is also named feature tracking. The most widely used tracker is KLT tracker [Lucas and Kanade, 1981]. By determining 2D-2D point correspondences in consecutive video frames, the relative camera geometry is established. We use the commercial software Boujou [2d3, 2008] to get camera projection information and corresponding 2D and 3D points.

2.3 Edge Extraction

Edges are first detected in each frame separately. First, an 8-bit binary edge map is generated in each image by running EDISON edge detector. As a improvement for Canny detector, a confidence measure is introduced in EDISON edge detector, which results in more connected and smoothed edges [Canny, 1986; Meer and Georgescu, 2001]. The second step is to use Hough transformation to extract straight edges from the edge map.

3. APPROACH

3.1 Method Overview

The most common model for cameras is the pinhole camera model: a point in 3D space is projected into an image by computing a viewing ray from the unique projection center to the point and intersecting this viewing ray with a unique image plane. During preprocessing steps, camera projection matrices for each frame are obtained with some corresponding points in 2D and 3D. Using reliably matched points as guidance for edge matching is the key point in this method, only edges near these good quality points are considered, which reduces the search space for corresponding 2D edges in frames. The workflow is described below:

1. Compute the covariance matrix of each tracked 3D feature point and chose reliable points based on it. This part will be explained in section 3.2.

2. Project a reliable 3D point to an image in which it is visible (or using corresponding image point), and calculate the distance between the 2D point and all edges detected in the same image. The distance here is the distance between a point and a finite edge. If the distance is less than one pixel, the edge is considered as an edge candidate in that image.

3. Use the same method described in step 2 to analyze edges from all the images in which the same 3D point is visible. By this measurement, edge candidates in images are obtained. This method is much faster than applying epipolar beam from end points to find candidates.

4. In order to enlarge the baseline to get a more accurate result, a 3D edge hypothesis is made between candidate edges from the first and the last images. Because the corresponding edge usually can not be extracted in every image, and also considering computation time, we choose the candidate edges from first and last ten percent images. A 3D infinite edge hypothesis is the intersection of two planes, each defined by one optical center and the corresponding 2D edge.

$$A = P_a^T l_a \tag{1}$$

$$B = P_b^T l_b \tag{2}$$

$$L = A \cap B \tag{3}$$

Where, l_a, l_b are 2D edges in image a and image b; P_a, P_b are projection matrixes of image a and image b; L is the intersection of plane A and plane B

5. Project the 3D infinite edge to each image. As a projection matrix P for points is known, $\mathbf{x} = P\,\mathbf{X}$, it's able to construct a projection matrix Q that can be applied to 3D edges, $\mathbf{l} = Q\,\mathbf{L}$, where Q is a 3×6 matrix. More details are given in [Hartley and Zisserman, 2000; Heuel, 2004].

Calculate distance and angle between projection results and edge candidates. If the distance and the angle is less than a predefined threshold, the edge candidate is considered as a corresponding edge for the 3D edge hypothesis.

Compare the number of corresponding edges with the number of images considered. If the rate is higher than fifty percent, the hypothesis is confirmed. Otherwise, it should be rejected and new hypothesis need to be made from edge candidates. Return to step 4.

6. When the hypothesis is confirmed, the corresponding edge in each image can received. From these 2D edges, 3D edge estimation is done see section 3.3 below. The 3D edge can still be rejected if the estimated variance factor is lager than a suitable threshold or if the solution does not converge.

7. Compute end points for the estimated 3D edge. By backward projecting rays from the end points of the corresponding 2D edges and taking the intersection with the estimated 3D edge, we get two sets of end point candidates for the 3D edge. The method described in section 3.4 is used to fix the end points.

8. Take next reliable 3D point, until all points are processed.

3.2 Point Quality Analysis

Assume a 3D point (X,Y,Z) is visible in n+1 images, a set of corresponding image points (x_i, y_i), $i = 0,\cdots,n$, and camera projection matrices P_i, $i = 0,\cdots,n$, for each frame in which the 3D point is visible are known. It is usually not the case that the rays of the corresponding points in the images intersect precisely in a common point, which means, the points' quality should be analyzed first. As the relation between 3D point and

its corresponding image point is $\mathbf{x} = P\,\mathbf{X}$, projecting a 3D point from object space to image plane i, the calculated image point (x_i', y_i') can be obtained.

The difference d between tracked image points (x, y) and corresponding calculated image point (x', y') can be expressed by

$$d = \sqrt{(x - x')^2 + (y - y')^2} \tag{4}$$

Then the standard deviation σ of image points corresponded to point (X, Y, Z) can be calculated by

$$\sigma^2 = \frac{\sum_{i=0}^{n} d_i^2}{n}, i = 0, \cdots, n \tag{5}$$

The 3D coordinate of point (X, Y, Z) is estimated by intersecting all its viewing rays as

$$\begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{pmatrix} = A \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{6}$$

Where,

$$A = \begin{bmatrix} \left(\frac{\partial x}{\partial X}\right)_1 & \left(\frac{\partial x}{\partial Y}\right)_1 & \left(\frac{\partial x}{\partial Z}\right)_1 \\ \left(\frac{\partial y}{\partial X}\right)_1 & \left(\frac{\partial y}{\partial Y}\right)_1 & \left(\frac{\partial y}{\partial Z}\right)_1 \\ \vdots & \vdots & \vdots \\ \left(\frac{\partial x}{\partial X}\right)_n & \left(\frac{\partial x}{\partial Y}\right)_n & \left(\frac{\partial x}{\partial Z}\right)_n \\ \left(\frac{\partial y}{\partial X}\right)_n & \left(\frac{\partial y}{\partial Y}\right)_n & \left(\frac{\partial y}{\partial Z}\right)_n \end{bmatrix} \tag{7}$$

The partial derivatives are computed directly using the Euclidean interpretation of the projection matrix. So, the covariance matrix C for 3D point can be obtained by

$$C = \sigma^2 \cdot (A^T A)^{-1} \tag{8}$$

Then, the theoretical precision of the computed 3D point can be expressed as error σ_{3D} according to,

$$\sigma_{3D}^2 = \sum_{i=0}^{2} C_{ii} \tag{9}$$

If σ_{3D} is larger than a suitable threshold, the 3D point is not accurate.

3.3 3D Edge Estimation

The geometric construction can be described as an estimation task, where an unknown 3D edge has to be fitted to a set of 2D edges from different images.

So a relation between a 3D line and 2D line can be defined as:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = Q \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \end{pmatrix} \tag{10}$$

The relation to the angle-distance form of a 2D line is given by a multiplication factor $1/\sqrt{a^2 + b^2}$:

$$l = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ -d \end{pmatrix} = 1/\sqrt{a^2 + b^2} \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{11}$$

If there are n lines matched across the image sequence, the 3D edge can be estimated by using Gauss-Markoff Model with constraints: N=3n observations l for U=6 unknown parameters L in Plücker coordinates with H=2 constraints h.

$$l + \hat{v} = f(\hat{L}) \tag{12}$$

$$h(\hat{L}) = 0 \tag{13}$$

In order to get corrections Δl and ΔL, the following Jacobians are needed:

$$A = \left.\frac{\partial f(L)}{\partial L}\right|_{L = L^0} \tag{14}$$

$$H = \left.\left(\frac{\partial h(L)}{\partial L}\right)^T\right|_{L = L^0} \tag{15}$$

An initial covariance matrix C_{ll} of the observed 2D edges can be calculated from the uncertainty of edge extraction result. More details are given in [Heuel, 2004]. So,

$$\begin{bmatrix} A^T C_{ll}^{-1} A & H \\ H^T & 0 \end{bmatrix} \begin{bmatrix} \Delta\hat{L} \\ \mu \end{bmatrix} = \begin{bmatrix} A^T C_{ll}^{-1} \Delta l \\ c_h \end{bmatrix} \tag{16}$$

$$\hat{v} = -(\Delta l - A\Delta\hat{L}) \tag{17}$$

Where, $\Delta l = l - f(L^0)$, $c_h = -h(L^0)$ and μ is Lagrangian multiplier [McGlone et al., 2004].

Then, the covariance matrix for unknown 3D edge $\hat{L}$ and the estimated residuals $\hat{v}$ can be obtained

$$C_{\hat{w}\hat{w}} = C_{ll} - AC_{\hat{L}\hat{L}}A^T \tag{18}$$

With, $C_{\hat{L}\hat{L}} = M^{-1} - M^{-1}H(H^T M^{-1} H)^{-1} H^T M^{-1}$ (19)

And $M = A^T C_{ll}^{-1} A$ (20)

The estimated variance factor $\hat{\sigma}^2$ is given by

$$\hat{\sigma}^2 = \frac{\hat{v}^T C_{ll}^{-1} \hat{v}}{N + H - U} \quad (21)$$

Finally the estimated covariance matrix can be obtained

$$\hat{C}_{\hat{L}\hat{L}} = \hat{\sigma}^2 C_{\hat{L}\hat{L}} \quad (22)$$

The initial value for the 3D edge is the intersection of backward projecting 2D edge from first and last image. The stopping criterion for iteration is that the changes $\Delta\hat{L}$ to estimation should be less than 1% with respect to their uncertainty or over a maximum iteration value.

If the estimated variance factor $\hat{\sigma}^2$ is larger than a suitable threshold $\sigma^2_{\max}$ or if the solution does not converge, the estimated 3D edge is rejected.

3.4 End Points Decision

The last part of the algorithm is the computation of the end points of the 3D edges. By backward projecting rays from the end points of one corresponding 2D edge and taking the intersection with the estimated 3D edge, we can get two endpoints. Considering the direction vector of 3D edge, we can separate intersection points to two groups, as shown in figure 1. The red circle area shows where the intersection points are. Then we get a set of end point candidates for each 3D edge end point.

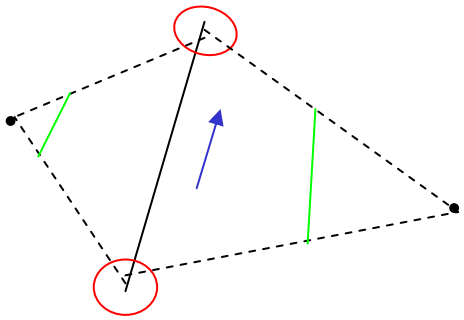


Figure 1. End points decision
Optical centers (black points), 2D edges (green), 3D edges (black solid line), viewing rays (black dashed lines), direction vector (blue)

The uncertainty value of corrections for each 2D edge is used as a weight for its affection on end points of 3D edge. The weight value can be obtained from covariance matrix of estimated residuals.

4. EXPERIMENT

4.1 Data Description

Above video data was captured by a hand-hold Canon IXUS camera moving along a street. The images are 640×480 pixels, 15 frames per second and 134 frames in total. Figure 2 shows first and last frame from input image sequence with reliable points and edge extraction results. Tracked points from Boujou with σ_{3D} less than 8mm are considered as reliable points. The number of extracted edges from each frame varies from 72 to 96, about 87 in average.

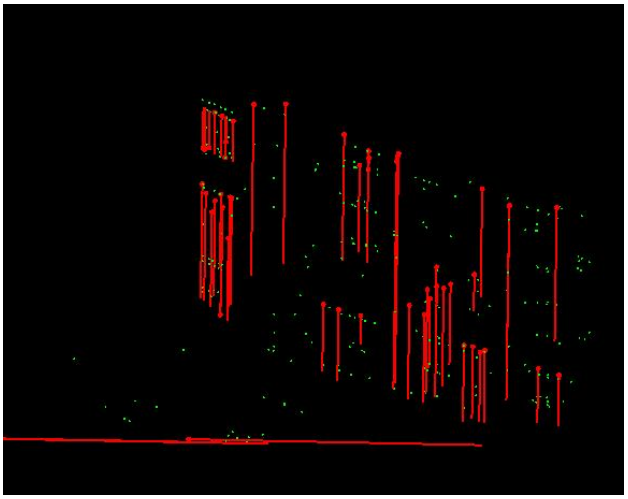
Figure 2. Input video image sequence with reliable points and extracted edges,
Reliable points (green), edges with end points (yellow)
frame 0 (upper), frame 133 (lower)

4.2 Results

A common way for taking video is to maintain a constant height of the camera during capture. As the camera is moving horizontally, horizontal edges are almost at epipolar plane between different view points. For such poor geometry relation, they are difficult to be correctly estimated. By setting the suitable threshold for estimated variance factor, those incorrect 3D edges can be eliminated.

We chose 200 as the max iteration value during 3D edge estimation and $\sigma^2_{\max}$ =0.1 for estimated variance factor. Figure 3 show first and last frame from image sequence with edges that successfully reconstruct 3D edges that are showed in figure 4. As there are many cars in front of the building, edges on the ground and cars are usually connected and easily mixed up, which leads to two incorrect 3D edges extracted in front of the building. But all the other edges fix the wall plane very well and the main building plane can be seen from the extracted 3D edges. Comparing figure 2, figure 3 and figure 4, our method can correctly match edges in short range video image sequence.

Figure 3. Matched 2D edges with reliable points, Reliable points (green), edges with end points (red) frame 0 (upper), frame 133 (lower)



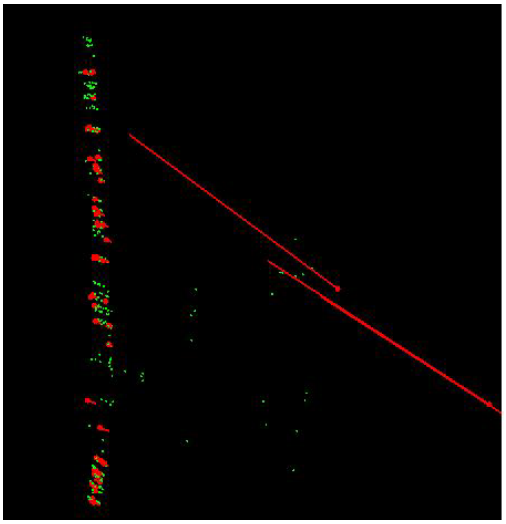


Figure 4. 3D view on estimated 3D Edges with reliable points, side view (upper), top view (lower)

5. CONCLUSIONS

Video as an easy obtainable and low cost image data source has been of interest for many researchers in recent years. However recovering accurate structure of objects in realistic environment from video image sequence is still a difficult problem. Especially reconstruction from images captured by hand held camera is a challenging topic, whose image resolution is low, and not all corner points can be detected. Edge information can provide more constraints about objects' shape. But quickly searching corresponding edges from image sequence and reconstructing their accurate position correctly is difficult. In this method, only edges near the reliable matched points are considered as edge candidates, which can reduce the search space for corresponding 2D edges in frames and is much faster than applying epipolar beam from end points to find candidates. So, this method can significantly simplify and speed up the edge matching procedure. In order to avoid a poor ray intersection geometry caused by the short baseline between the images, all the edge candidates are used to estimate a 3D edge. Based on the estimated variance factor, only good 3D edge estimations are accepted, which ensures the accurate position of matched 3D edges. But it should be mentioned here that the decided end points may not correspond to corner points.

Although we can not get enough information from edge matching results to reconstruct the whole objects, these matched edges contain relation with points. Further on, surface patch and other feature aggregates are connected, which are useful topological constraints for recovering 3D structures.

REFERENCES

2d3, 2008. *Homepage boujou software*. http://www.2d3.com (accessed 27 March 2008)

Armstrong, M. and Zisserman, A., 1995. *Robust object tracking*, Asian Conference on Computer Vision, pp. 58-61.

Baillard, C., Schmid, C., Zisserman, A. and Fitzgibbon, A., 1999. *Automatic line matching and 3D reconstruction of buildings from multiple views*, International Archives of Photogrammetry and Remote Sensing, pp. 69-80.

Baltsavias, E.P., 2004. *Object extraction and revision by image analysis using existing geodata and knowledge: current status*

and steps towards operational systems. ISPRS Journal of Photogrammetry and Remote Sensing, 58(3-4): 129-151.

Canny, J., 1986. *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6): 679-698.

Hartley, R.I. and Zisserman, A., 2000. *Multiple View Geometry in Computer Vision (second edition)*. Cambridge University Press.

Heuel, S., 2004. *Uncertain Projective Geometry: statistical reasoning from pholyhedral object reconstruction*. PhD Thesis, University of Bonn, Germany.

Klein, G. and Murray, D., 2006. *Full-3D Edge Tracking with a Particle Filter*, British Machine Vision Conference (BMVC'06), Edinburgh, pp. 1119--1128.

Kunii, Y. and Chikatsu, H., 2004. *Efficient line matching by image sequential analysis for urban area modelling*, XX ISPRS Congress, Youth Forum Istanbul, Turkey.

Lucas, B.D. and Kanade, T., 1981. *An Iterative Registration Technique with an Application to Stereo Vision*, Int. Joint Conf. on Arificial Intelligence

McGlone, J.C., Mikhail, E.M., Bethel, J. and Mullen, R., 2004. *Manual of Photogrammetry*. American Society for Photogrammetry and Remote Sensing.

Meer, P. and Georgescu, B., 2001. *Edge detection with embedded confidence*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23(12): 1351-1365.

Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J. and Koch, R., 2004. *Visual modelling with a hand-held camera*. International Journal of Computer Vision, 59(3): 207-232.

Remondino, F. and EL-Hakim, S., 2006. *Image-Based 3D Modeling: A Review*. The Photogrammetric Record, 21(115): 269-291.

Schmid, C. and Zisserman, A., 1997. *Automatic line matching across views*, the 1997 conference on computer vision and pattern recognition (CVPR'97), Washington DC, USA, pp. 666-671.

Zhang, Y., Zhang, Z., Zhang, J. and Wu, J., 2005. *3D Building Modelling with Digital Map, Lidar Data and Video Image Sequences*. The Photogrammetric Record, 20(111): 285-302.