Recovering Epipolar Geometry by Reactive Tabu Search

Qifa Ke	Gang Xu	Song De Ma
Carnegie Mellon University	Ritsumeikan University	Chinese Academy of Science
Pittsburgh, PA 15213, USA	Kusatsu, Shiga 525, Japan	Beijing 100080, P.R. China
ke+@cs.cmu.edu	xu@cs.ritsumei.ac.jp	masd@nlpr.ia.ac.cn

Abstract

We propose a new approach to recover epipolar geometry from a pair of uncalibrated images. By minimizing a proposed cost function, our approach matches the detected feature points from an image pair, discards the outliers and recovers the epipolar geometry simutaneously. Experiments on real images show that this approach is effective and fast.

1 Introduction

Recovering epipolar geometry between uncalibrated views is an important task with a lot of applications, such as determining the underlying 3D motion, reconstructing the 3D structure of the scene, and matching the views as a 1D search along the epipolar lines [7, 8, 11]. However, such task is difficult because it usually requires matching the feature points based on the epipolar equation, which is basically a search problem in a very large space, and is sensitive to outliers. Traditional approaches to this problem are relaxation and robust estimation [9, 1], and clustering [11, 1]. Both approaches, however, require expensive computation.

In this paper, we propose a new approach, the Reactive Tabu Search approach, to solve this problem. Tabu Search is a metaheuristic search technique that guides a local heuristic search procedure to explore the solution space beyond local optimality, and has been proven to be effective in many optimization problems [2].

We will first briefly describe the epipolar geometry under affine projections, and restate the problem of recovering epipolar geometry as a problem of matching feature points from an image pair in such a way that a cost function is minimized. The cost function is a measurement of how well the matched points satisfy a common epipolar equation. We will then describe how to apply the Reactive Tabu Search to this particular problem and show the experimental results.

2 Formulating Epipolar Geometry Recovery as Cost Function Minimization

2.1 Determining Epipolar Equation under Affine Projections from Point Correspondences

In general, the epipolar geometry is described by the following equation:

$$\tilde{\mathbf{x}}^T \mathbf{F} \tilde{\mathbf{x}}' = 0 \tag{1}$$

where $\mathbf{F}$ is the 3×3 fundamental matrix, $\tilde{\mathbf{x}} = [x, y, 1]^T$ and $\tilde{\mathbf{x}}' = [x', y', 1]^T$ are the augmented vectors for the corresponding points in the two images.

In the case of affine cameras, the 4 upper left components in $\mathbf{F}$ are zeros. Thus expanding (1) gives a linear equation in the image coordinates,

$$f_{13}x + f_{23}y + f_{31}x' + f_{32}y' + f_{33} = \mathbf{f}^T \mathbf{p} + f_{33} = 0 \tag{2}$$

where $\mathbf{p} = [x, y, x', y']^T$ and $\mathbf{f} = [f_{13}, f_{23}, f_{31}, f_{32}]^T$. This equation can be understood geometrically as epipolar lines on each image.

Recoverying the epipolar geometry means determining the above epipolar equation. All point matches must satisfy the same epipolar equation. Given n pairs of point matches, we can determine the epipolar equation by minimizing squared distances from each point to the epipolar line in each image, taken into account the scale change between the two images. This is proven to be the same as minimizing the squared distances from each pair of point matches, as a 4D point, to the hyperplane in the 4D space determined by the epipolar equation [1],

$$C = \frac{1}{2} \sum_{i=1}^{n} \frac{(\mathbf{p}^T \mathbf{f} + f_{33})^2}{\mathbf{f}^T \mathbf{f}} \tag{3}$$

By minimizing this cost funtion, $\mathbf{f}$ is determined as the eigen vector associated with the smallest eigen value of $\mathbf{W} = \sum_{i=1}^{n} (\mathbf{p_i} - \mathbf{p_o})(\mathbf{p_i} - \mathbf{p_o})^T$ and f_{33} is determined as $f_{33} = -\mathbf{p_o}^T \mathbf{f}$, where

$\mathbf{p}_0 = \frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_i = \frac{1}{n}[\sum_{i=1}^{n}x_i, \sum_{i=1}^{n}y_i, \sum_{i=1}^{n}x'_i, \sum_{i=1}^{n}y'_i,]^T$. The residual of this minimization is exactly the smallest eigen value of $\mathbf{W}$ [1, 10].

Since the point matches are not given *a priori*, we need to find the correspondences such that they satisfy the same epipolar equation. The problem can now be restated as trying to find as many pairs of points as possible such that the smallest eigen value of $\mathbf{W}$ is as small as possible.

2.2 Formulating the Cost Function

Suppose we have detected n_1 points in the first image and n_2 points in the second image. Our algorithm is to find the global minimum of a cost function, which corresponds to the state of correct point correspondences. Before we give the cost function, we first introduce the *match matrix*. We assume that we are given two 2D point sets, X_i and X'_j, then the *match matrix* $\{m_{ij}\}$ is defined as:

$$m_{ij} = \begin{cases} 1 & \text{if point } x_i \text{ corresponds to point } x'_j \\ 0 & \text{otherwise,} \end{cases}$$

It is obvious that the matrix $\mathbf{W}$ can be completely determined from $\{m_{ij}\}$. Since some points may not have correspondences in the other image, we add an extra row $R = [r_1, r_2, \ldots, r_{n_2}]$ and an extra column $C = [c_1, c_2, \ldots, c_{n_1}]$ to the *match matrix*. That is, if the sum of a row or a col in $\{m_{ij}\}$ equals to zero, then the corresponding element in the extra row R or column C is set to 1. Other elements in the extra row or column are set to zero. In fact, the extra row and column are called slacks. Now our cost function can be defined as:

$$\begin{aligned} E(m) &= \frac{1}{2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} m_{ij}\frac{(\widetilde{\mathbf{x}}_\mathbf{i}\mathbf{F}\widetilde{\mathbf{x}}_\mathbf{j})^2}{\mathbf{f}^T\mathbf{f}} \\ &\quad -\lambda\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} m_{ij} \\ &= V(\mathbf{W}) - \lambda\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} m_{ij} \end{aligned} \tag{4}$$

where $V(\mathbf{W})$ is the smallest eigenvalue of $\mathbf{W}$. The λ term is added to include all correct matches while rejecting outliers (points that do not have matches) simultaneously. For a given pair $(\mathbf{x_i}, \mathbf{x_j})$, if $\lambda > \frac{1}{2}\frac{(\widetilde{\mathbf{x}}_\mathbf{i}\mathbf{F}\widetilde{\mathbf{x}}_\mathbf{j})^2}{\mathbf{f}^T\mathbf{f}}$, then this pair will not be regarded as outlier (i.e. will not be discarded) since making $m_{ij} = 1$ will lead to a less cost.

Our task is to minimize the above cost function to find the correct point matches, that is, to find the correct *match matrix*. The optimization technique used here is Reactive Tabu Search.

2.3 Determining Epipolar Equation under Full Perspective Projection

Our approach can also be applied to the full perspective projection case. Under full perspective projection, from Eq.(1) we have::

$$\mathbf{p^T f} + f_{33} = 0 \tag{5}$$

where

$$\begin{aligned} \mathbf{p} &= [xx', yx', x', xy', yy', y', x, y]^T \\ \mathbf{f} &= [f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}]^T \end{aligned} \tag{6}$$

Eq. (6) is identical in form to Eq.(2) if we subtitute $\mathbf{p}, \mathbf{f}$ with Eq.(6). Therefore, $\mathbf{f}$ is the eigen vector associated with the smallest eigen value of $\mathbf{W} = \sum_{i=1}^{n}(\mathbf{p_i} - \mathbf{p_0})(\mathbf{p_i} - \mathbf{p_0})^T$ and $f_{33} = -\mathbf{p_0}^T\mathbf{f}$, where

$$\begin{aligned} \mathbf{p_0} &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_i \\ &= \frac{1}{n}[\sum_{i=1}^{n} x_i x_i', \sum_{i=1}^{n} y_i x_i', \sum_{i=1}^{n} x_i', \sum_{i=1}^{n} x_i y_i', \\ &\quad \sum_{i=1}^{n} y_i y_i', \sum_{i=1}^{n} y_i', \sum_{i=1}^{n} x_i, \sum_{i=1}^{n} y_i]^T \end{aligned} \tag{7}$$

The formulation of cost function for full perspective case is the same as that for the affine cameras in Equation (4). Therefore, our approach can be applied to both affine and full perspective projections.

3 Applying Reactive Tabu Search

3.1 Introduction to Reactive Tabu Search

In this section we give a general description on Tabu Search and Reactive Tabu Search. Please refer to[2] for more details. Tabu search has been proven effective for many optimization problems. It is a metaheuristic method that guides a local heuristic search procedure to explore the solution space beyond local optimality[2]. It is different from the well-known hill-climbing local search techniques in that the tabu search may move to a worse soultion in the hope that it will eventually achieve a better solution. It is also different from the stimulated annealing [4] and genetic algorithm [5] because the tabu search includes a memory mechanism. According to Glover's idea, in order to solve a problem using tabu search, the following components must be defined[2, 3].

Configuration: *Configuration* is a solution or an assignment of values to variables.

Move: A *move* characterizes the process of generating a feasible solution to the problem that is related to the current solution (i.e. a move is a procedure by which a new solution is generated from the current one).

Neighbourhood: A *neighbourhood* of the solution is the collection set of all possible moves out of a current configuration. Note that the actual definitions of the neighbourhood depend on the particular implementation and the nature of problem.

Tabu Conditions: In order to avoid a blind search, tabu search technique uses a prescribed problem-specific set of constraints, known as *tabu conditions*. *moves* that violate the *tabu conditions* are known as *tabu moves*. A *tabu list* is maintained to record these forbidden moves.

Aspiration Condition: These are rules that override tabu restrictions, that is, if a certain move is forbidden by some tabu restriction, then the aspiration criterion, when satisfied, can make this move allowable.

With the above basic components, the tabu search algorithm can be described as follows.

(i) Start with a certain (current) configuration and evaluate the criterion function for that configuration.

(ii) Generate a neighbour of the current configuration, that is, a set of candidate moves. If the best of these moves is not a tabu move, or if the best is a tabu move but it satisfies the aspiration criterion, then pick that move and consider it to be the new current configuration; otherwise pick the best move that is not a tabu move and consider it to be the new current configuration.

(iii) Repeat (i) and (ii) until some termination criteria are satisfied.

The best solution in the final loop is the solution obtained by the algorithm. Note that the move picked at a certain iteration is put in the tabu list so that it is not allowed to be reversed in the next iterations. So Tabu Search is a memory search method. The size of the tabu list represents its memory ability. The larger the size, the stronger the memory. However, with a fixed list size, it is possible that the searching trajectory may form a limit cycle (endless repeats of a sequence of states), which can be avoided by increasing the list size. But long list size will cause low efficiency and most of time a small list size is enough. Based on the above discussion, some researchers propose the Reactive Tabu Search (RTS) method [6].

The RTS method maintains the basic steps of Tabu Search, except that the size of tabu list is adaptive to the problem and current evolution of the search. By storing the configurations visited during the search and its corresponding iteration numbers, we can calculate the interval between two visits when configuration repetition happens. The list size is increased when configurations are repeated; otherwise the size is reduced in regions of the search space where large size of list is not needed.

3.2 RTS for Finding Point Matches Under Epipolar Geometry

In this section, we will show how to use RTS to find point correspondences, that is, how to solve the optimization problem of Eq.(4) with RTS. We first give the main components of RTS in the context of this specific problem.

Configuration: Here it is the *match matrix*.

Move: There are three kinds of moves: exchange two rows in the extended *match matrix*; discard a matched pair as a outlier; or add a matched pair $(\mathbf{x_i}, \mathbf{x_j})$ in the case of $r_i = c_j = 1$.

Neighbourhood: At a current solution (configuration), all kinds of possible move are considered as its neighbours.

Tabu List: Contains the moves that violate the tabu conditions. The current chosen move is added to the list, whose size is adaptive.

Aspiration Condition: If a move leads to a better solution, this move is chosen even if it is in the Tabu List.

Let I_c, I_t and I_b denote the current, trial and best configurations (*match matrix*), and f_c, f_t and f_b denote the corresponding current, trial and best objective function values, respectively. As described in the previous subsection, we start with a configuration which is known as the current solution I_c and then through *moves*, as explained above, we generate trial solutions I_t. As the algorithm proceeds, we also save the best solution found so far which is denoted by I_b. Corresponding to these configurations, we also record the objective function values f_c, f_t and f_b, respectively. We denote $MTLS$ as the current tabu list size and TLL as the number of elements in the tabu list.

For the point correspondence problem, the RTS algorithm can be described as:

Step 1 *Initialization*: Let I_c be an arbitrary solution, and f_c be the corresponding objective function value computed using equation Eq.(4). Let $I_b = I_c$ and $f_b = f_c$. Select values for the following parameters: $MTLS = 1$ (initial tabu list size), let $IMAX$ be the maximum number of iterations. Let $k = 1$, where k is the iteration step index.

Step 2 *Generating Neighbourhood*: Using I_c to generate NTS trial solution $I_t^1, I_t^2, \cdots, I_t^{NTS}$ (NTS is the number of all possible moves), evaluate their corresponding objective function values $f_t^1, f_t^2, \cdots, f_t^{NTS}$, and go to step 3.

Step 3 Order $f_t^1, f_t^2, \cdots, f_t^{NTS}$ in an ascending order, and denote them by $f_t^{[1]}, f_t^{[2]}, \cdots, f_t^{[NTS]}$. If $f_t^{[1]}$ is not a tabu, or if it is a tabu but $f_t^{[1]} < f_b$ (i.e., the aspiration condition is satisfied), then let $I_c = I_t^{[1]}$ and $f_c = f_t^{[1]}$, and go to Step 4; otherwise, let $I_c = I_t^{[L]}$ and $f_c = f_t^{[L]}$, where $f_t^{[L]}$ is the best objective function of $f_t^{[2]}, \cdots, f_t^{[NTS]}$ that is not a tabu and go to Step 4. If all $f_t^{[1]}, f_t^{[2]}, \cdots, f_t^{[NTS]}$ are tabu, decrease the size of tabu list and select $I_t^{[1]}$ as the chosen move in spite of their tabu status, then go to step 5.

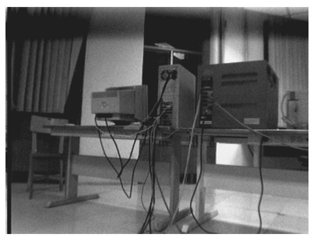
step 4 Execute the chosen move $I_t^{[1]}$ and check for repetition. If a repetion of a previously encountered configuration occurs, increase the tabu list size to discourage the additonal repetitions. However, if the list size is becoming too large, it is slowly reduced. Go to step 5.

Step 5 Insert the current move I_c at the bottom of the tabu list and let $TLL = TLL + 1$ (if $TLL = MTLS + 1$, delete the first element in the tabu list and let $TLL = TLL - 1$). If $f_b > f_c$, let $I_b = I_c$ and $f_b = f_c$. If $k = IMAX$, stop (I_b is the best solution found and f_b is the corresponding objective function value); otherwise, let $k = k + 1$ and go to Step 2.

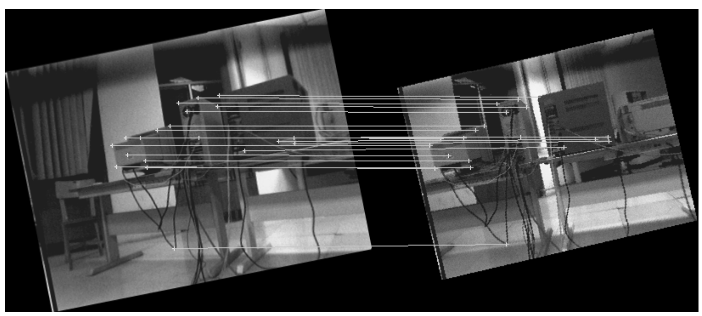
The result of the above algorithm is the match matrix that maximize the number of matches and minimize the cost as in (3).

4 Experimental Results and Discussion

We have tested the above algorithm on real images. Fig.1(a) and Fig.1(b) are a pair of indoor images. The above algorithm is applied to the detected corners in this pair of images. The fundmental matrix is obtained at the same time as the points are correctly matched. Using the recovered epipolar equation, we can rectify the images [1, 8] so that the epipolar lines become horizontal. Fig.1(c) shows the result, where the matched points are linked by horizontal lines.



(a) (b)



(c)

Figure 1: Finding point correspondence and epipolar equation via RTS. (a) and (b): Real indoor image pair. (c): the rectified images with horizontal lines linking the matched points.

The RTS has been found to be able to find the correct match points very quickly in the case of less than 50 feature points. For example, in the case of Fig.1(a) and Fig.1(b), there are 36 and 41 feature points. The computation time is about 20 seconds on a Sun Sparc 10 machine.

If the number of corners in the image is large, e.g., several hundreds or thousands, we need other mechanisms to break the "large problem" into several small ones first and then deal with each small problem. For example, we can use grouping techniques first. At each time we only deal with one small group of features from each image. This will greatly reduce the computation cost. Another technique is using correlation techinque to reduce the match candidates. The impossible configurations can be obtained from the initial match results and can be regarded as the tabu configurations. We can then refine the initial match results with RTS. This refining process is computationally effective.

5 Summary and Conclusions

We have proposed using Reactive Tabu Search to find point matches between two uncalibrated views such that they satisfy the identical epipolar equation. The preliminary experimental results show that this approach is very effective and efficient. The authors believe that the RTS can be applied to many other computer vision problems, and hope that this paper can stimulate interest among CV researchers in this approach.

References

[1] G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*, Kluwer Academic Press, 1996.

[2] F. Glover, Tabu Search, *in* Modern Heuristic Techniques for Combinatorial Problems, C.R. Reeves ed., John Wiley & Sons, Inc., 1993.

[3] Khaled S. Al-Sultan, A tabu search approach to the clustering problem, *Pattern Recognition*, **28**:1443-1451,1995.

[4] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, Optimization by Stimulated Annealing, *Science*, **220**:621-680, 1983.

[5] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[6] Roberto Battiti etc, The reactive tabu search, *ORSA Journal on computing*, **6**:126-140,1994.

[7] H. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, *Nature*, **293**:133-135,1981.

[8] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press,1993.

[9] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong, A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, *Artificial Intelligence Journal*, **78**:87-119,1995.

[10] L. Shapiro, A. Zisserman and M. Brady, 3D motion recovery via affine epipolar geometry, *Int. Journal of Computer Vision*, **16**:147-182,1995.

[11] G. Xu, A unified approach to image matching and segmentation in stereo, motion and object recognition via recovery of epipolar geometry, *VIDERE: A Journal of Computer Vision Research*, **1**, 1997.