Application of Simulated Annealing to the General Image Point Correspondence Problem

Jay C. McGaffigan John A. Orr
Worcester Polytechnic Institute
Worcester, MA

Abstract

This paper addresses the general image point correspondence problem, with no a priori constraints on camera or object motion. The solution is cast as a cost minimization problem using the simulated annealing algorithm. This is achieved by using a cost function to evaluate the quality of match between pairs of points extracted from each image. Using edge detection and feature extraction techniques points from each image are found which possess strong local features, such as curvature, concavity, standard deviation, and gradient. These features are used to define the cost function for candidate correspondence. The overall algorithm was implemented and tested using a unix workstation. Test image pairs that were used represented a wide range of relative motions.

1. Introduction

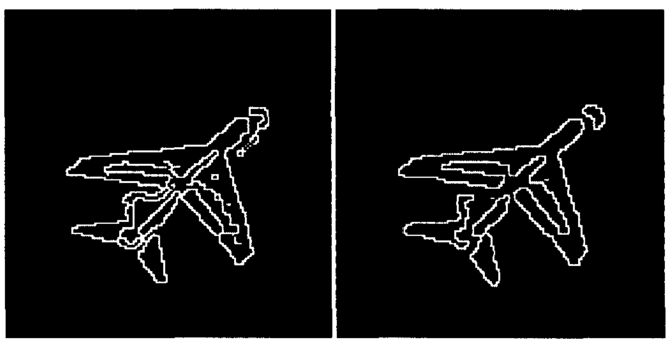
A step common to many computer vision problems is the determination of point correspondences. Solving the point correspondence problem involves matching points between images that represent the same physical location on an object (or objects) in the images, i.e. table corners, wing tips, borders of objects, etc. This correspondence information greatly facilitates the solution for other unknowns, such as motion parameters, structure, and shape. Finding point correspondences is relatively easy for humans, but represents quite a challenging problem to automate.

The essential operation of any point correspondence algorithm consists of two steps: (1) the determination of a set of points in each image that represents distinctive local image information; and (2) the generation of pairs of (hopefully) corresponding points between the two images [1,5,10,16], based on similarity of feature information. Typically a cost function will be applied so that a numerical measure of the quality of each match will be generated and the sum of these values will be optimized over all the possible sets of pairings. As the number of feature points increases, the number of combinations to search through drastically increases. For example if there were 30 points in each image then an exhaustive search would require that $1.34\mathrm{x}10^{32}$ combinations would be investigated.

Much previous work on point correspondence has involved constraints on relative disparity between images, such as in stereo vision. In the work reported here, however, a method for solving the point correspondence problem without constraints on camera or object motion is developed. Feature points are extracted based on edge curvature. Simulated annealing is then used to find a best set of point matches, using information local to the feature points.

2. Feature point extraction

The first step in any point correspondence problem is to select a set of candidate points, which contain distinctive local image information. In this algorithm a Laplacian of Gaussian(LoG) edge detector was constructed. The LoG detector [2,9,14,15,18,19] was then convolved with a set of test images. The resultant images were then convolved with a set of 2x2 Sobel operators. This produced edges that were a few pixels wide. The medial axis transform was used to obtain single pixel wide edges. See Figure 1 for an example of the performance of the edge detector. Figure 1a was convolved with a 3x3 LoG operator and Figure 1b was convolved with a 5x5 LoG operator. The 5x5 operator is a wider gaussian than the 3x3 operator. Note that the larger operator will filter out the smaller edges.



a b

Figure 1. Edge detector with two different LoG masks

The method chosen to extract feature points from edges was to calculate the curvature of all the edge pixels. The curvature values for each edge point were obtained by fitting third order polynomials to the edge points surrounding the edge point being examined [3]. Then edge pixels whose curvature values were above a certain threshold were chosen as feature points. This threshold can be altered to accept more or fewer feature points. Different types of images could have different thresholds.

3. Feature vector

The local area information associated with these feature points was then examined and feature vectors were constructed. Numerical values of the following specific image features are included in the feature vector associated with each selected point:

Curvature: Based on fitting third order polynomials to the x and y coordinates of nine pixels (the selected point and 4 points on each side) a measure of curvature was calculated. [3]

Concavity: The number of convex to concave transitions along the edge that contains the selected feature point is computed.

Two–D Standard Deviation: the standard deviation of the intensity in the 5x5 neighborhood around the selected point is computed.

Textural energy: the method proposed by Laws [12] as applied to a 7x7 neighborhood is used to generate measures of textural energy.

Gradient: the gradient of image intensity across the edge at the selected point (three pixels on each side) is calculated.

The above elements are used to construct the feature vectors. Using these feature vectors a measure of match is developed. A measure of match, Eqn 2.1, is defined as the

$$\Delta = \|\mathbf{v}_1 - \mathbf{v}_2\| \qquad (2.1)$$

norm of the vector difference between two feature vectors. Then a total measure of match is defined by summing all these differences over a set of possible matches.

4. The simulated annealing algorithm

Given the set of features in each image, the next step of the process is to apply an optimization technique to find a solution to the point correspondence problem. Two sets of feature points have been isolated, and using these points two sets of feature vectors have been constructed. In the matching process the match metric is calculated as the sum of all vector differences over candidate matches. The number of possible pairs is constrained by the image containing the fewest feature points. For example, if one image has 10 feature points and the other has 12, then at most we can only have 10 possible point correspondences. The next step, ideally, is to find the set of matches that has the most correct pairings, up to 10 in the previous example. As mentioned earlier, the best set of matches has the smallest total difference (ideally zero).

It should be noted that this optimization is directed at minimizing the vector difference among candidate match pairs. The feature set has been selected in an attempt to identify distinguishing characteristics of specific image points, but corresponding features may have non–zero (and in some cases large) vector differences, and non–corresponding features may have small vector differences.

To guarantee finding the smallest total difference, every possible set of matches must be examined and its difference calculated. When there is a large number of points to choose from, then the cost of searching through all the possible matches becomes prohibitive, on the order of **n**! In the previous example, we would have to examine

$$T = \frac{N!}{2!} \qquad (3.1)$$

3,628,800 different combinations to find the best set of point matches, and as the number of points increases the search space also drastically increases. Exhaustive searching through all these possibilities quickly becomes unmanageable, so a more efficient non–exhaustive search technique needs to be applied.

The simulated annealing algorithm was chosen because of its ability to avoid terminating at local minima and to keep searching for the global minimum [4,7,8,11]. The simulated annealing algorithm is analogous to the annealing process for metals, wherein the material is raised to a high temperature and then gradually cooled, allowing the atoms to settle into their most desirable states. In our case the desirable end state is the global optimum, and atomic motion due to thermal energy is simulated by a randomization routine which allows "uphill" acceptances of states with probability controlled by the temperature parameter.

Parameters of the simulated annealing process are: starting temperature, T, rate of cooling, L (measured in iterations at each temperature), and cooling schedule, α (change in temperature, $T' = \alpha T$, after each set of iterations, L). These parameters directly affect the run time of the simulated annealing algorithm. If the rate of cooling is changed the chance of settling into the best set of matches changes also. An example of this is shown in Figures 2a and 2b. In the first graph ($\alpha = 0.95$) the temperature is decreasing at a slower rate than that of the second graph ($\alpha = 0.8$). This means the annealing algorithm is stopping at more temperatures and spending more time searching for the global minimum. In the second graph the algorithm is stopping at fewer temperatures and thus reducing the search space. This provides a speed increase in the runtime of the algorithm but at a cost of finding fewer correct matches.

The annealing algorithm uses the iterative method together with randomization techniques as described by Metropolis [11]. This method starts with an initial state, and then calculates its cost. Then a new state is randomly generated by perturbing the current state. If the cost of the new

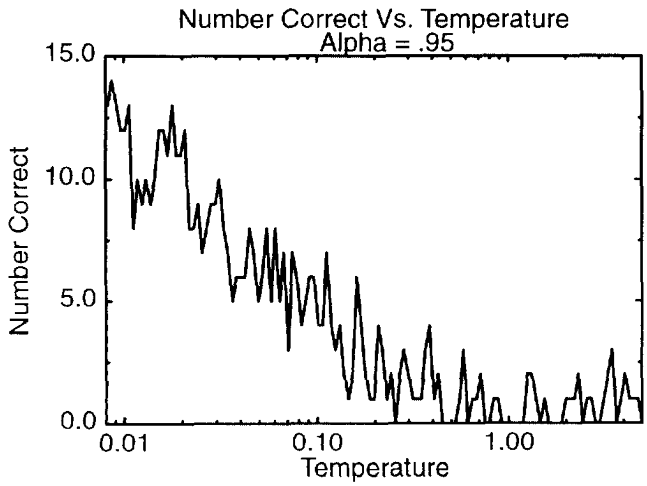


Figure 2a. Effect of cooling rate

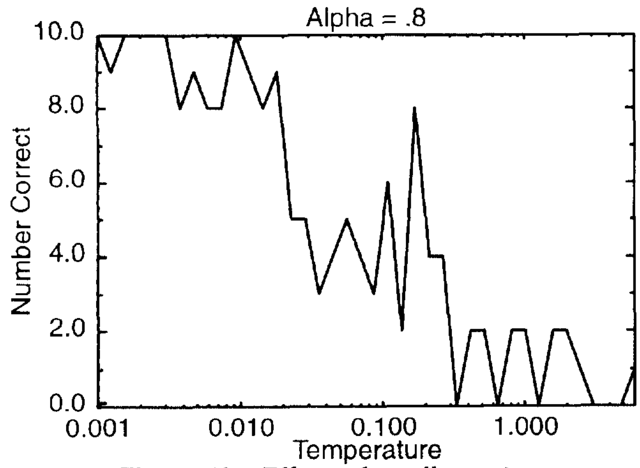


Figure 2b. Effect of cooling rate

state is less than the old then the process continues with the new state and the old state is discarded. If the new cost is greater than the old, there is still a chance that this state will be accepted and the old discarded. In this state, the Metropolis criterion, given by $M = \exp(\frac{-\Delta C}{k_b T})$, is calculated. A random number, x, uniformly distributed over the interval [0,1) is then chosen. If x is larger than M then the new state is rejected. Otherwise the process continues with the new configuration. The following pseudo-code is a brief outline of the simulated annealing process:

```
Select starting parameters T, L, and α
Generate initial configuration
repeat
   for i=1 to L
      begin
         generate(perturb current config. and
        calculate cost C_trial)
         if C_trial < C_cur then replace current
              config with trial config.
         else if Metropolis criterion >
            random[0,1) replace C_cur with C_trial
         else continue with current config
      end
   select new temperature
until stop criterion = true
```

5. Results of correspondence algorithm

In testing the algorithm, the starting values of the parameters were determined heuristically via a set of test images whose correct matches were known. These starting parameters were selected to help achieve an optimal solution in an acceptable amount of time.

Test images were generated by rotating, scaling, and translating the simple images of an aircraft. The following discussion examines some of the results of the point matching algorithm.

Figure 3 shows a test image pair with some of the feature points, as identified automatically by the algorithm, indicated.

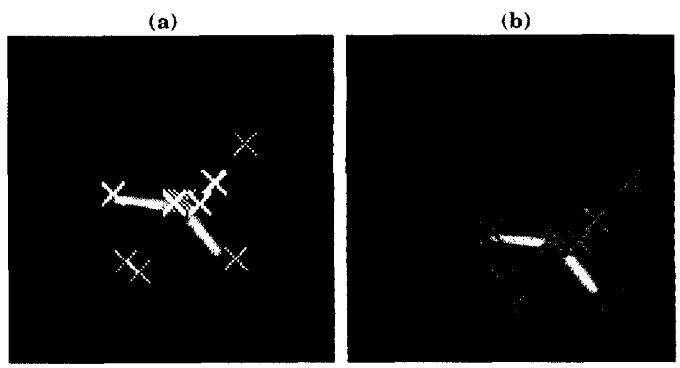


Figure 3. Feature points extracted; slight translation and rotation from a to b

In the first image, a total of 35 feature points were extracted, and in the second image there were a total of 33 feature point extracted. A comparison was done by eye and 21 possible *correct* matches were identified. out of the 33 total possible point matches. Running the point correspondence algorithm several times with differing initial conditions produced an average of 14 correct matches. Once we have a set of possible matches, the parameters of the transformation from one image to the other can be found with an error dependent upon the number of correct matches. Using the matches, a set of motion parameters is calculated and then applied to the second image to produce an image that, if we have found all the correct matches, will be very similar to the original image. This set of motion parameters can be used to determine how well the algorithm has performed. This set of motion parameters can also be used to rule out false point matches.

Additional test images were generated which are rotated, scale and translated versions of the simple aircraft image in Figure 3. The algorithm was observed to locate some correct matches in each case, Figure 4a–h. In all cases the left column image is the original plane image and the right column is the altered version. Due to some large transformations being applied to the images, many of the features located in the first image have disappeared. This algorithm performs the best when there are small changes between the images. Table 1 shows the number of correct matches found, the number of possible correct matches, and the number of possible paired matches for different sets of test images. The first 4 rows show the performance where there were large rotations applied between images. The last two rows show some results for small changes between images. Image pair A and image pair B are not shown here as they are very similar to Figure 1. Image pair A contained a large translation with no rotations. Image pair B has the same rotations as in Figure 1, but there is a much smaller translation. Both of these images are of the same plane as shown in the other figures.

Table 1 Correct Point Matches From Test Images

Image	Correct Matches	# Possible Correct	# Possible Matches
Fig 4 a–b	5	13	25
Fig 4 c–d	3	6	16
Fig 4 e–f	4	14	35
Fig 4 g–h	5	7	13
Fig 3 a–b	15	22	33
A	31	33	33
B	15	21	33

6. Conclusion

An algorithm for solving the general image point correspondence problem is presented. Features associated with certain edge characteristics were extracted from a pair of

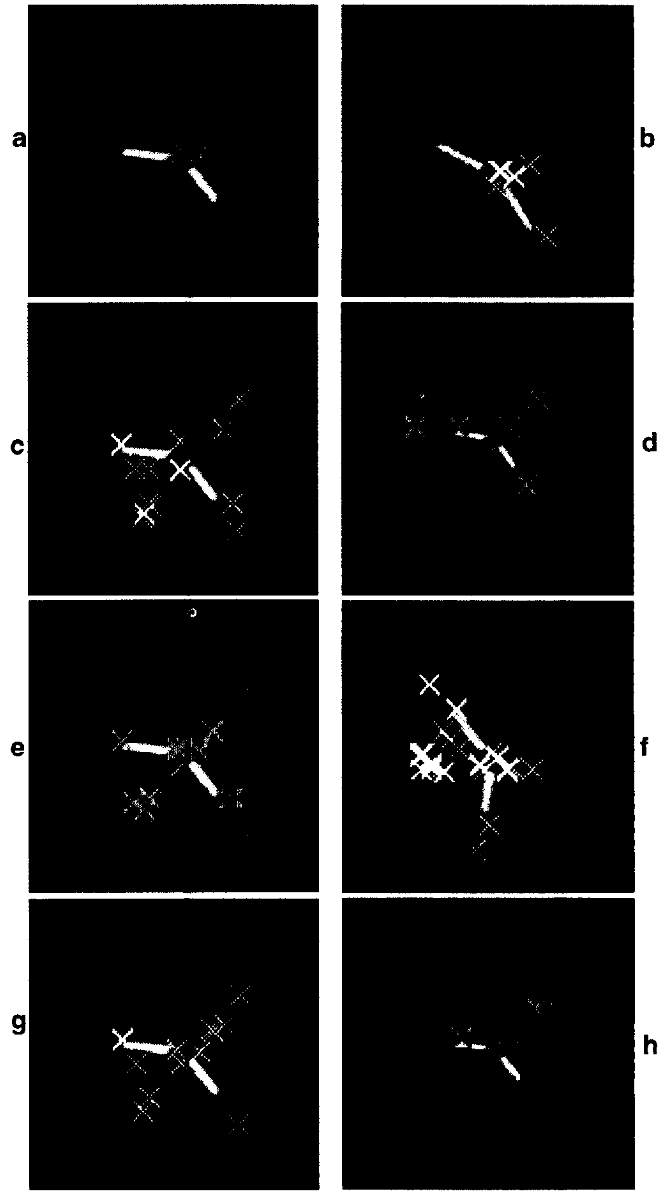


Figure 4. Match points

images. The point correspondence algorithm uses simulated annealing to find a best set of matches between feature points with distinct local area information. The overall performance of the algorithm is encouraging.

The point correspondence algorithm produces good results (on the order of 60% correct matches) when there are small changes in camera orientation between images. Also, in the presence of large changes the algorithm worked well (about 50% correct matches) when the feature extractor identified corresponding features in the two images. When feature points were manually selected in images with large changes the simulated annealing algorithm succeeded in locating about 65% of the correct point matches. This indicates that improvements in feature extraction would be potentially beneficial.

7. References

1. M.T. Boraie and M.A. Sid–Ahmed, "Points of Correspondence in stereo images with no specific geometrical constraints using mathematical morphology," *Computers in Industry*, Vol 20,1992 pp 295–310.

2. Canny, John, "A Computational Approach to Edge Detection," *IEEE Trans. PAMI*, PAMI–8 November 1986, pp 679.

3. D. Cyganski, J. A. Orr, T. A. Cott, R.J. Dodson, "Object Identification and Orientation Estimation from Contours Based on an Affine Invariant Curvature," SPIE Vol 848 Intelligent Robots and Computer Vision: Sixth in a Series, 1987, pp 33.

4. A. Dekkers, E. Aarts, "Global optimization and simulated annealing," *Mathematical Programming*, vol 50, 1991, pp. 367.

5. U. Dhond and J.K. Aggarwal, "Structure from Stereo–A Review," *IEEE Trans. Systems, Man, and Cybernetics*, Vol 19, No 6, November/December 1989, pp 1489.

6. B. Fannery, W. Press, S. Teukolsky, W. Vetterling, Numerical Recipes in C 2nd Edition, Cambridge Press, 1992

7. S. Geman, D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of images," *IEEE Trans. PAMI* , Vol. PAMI–6, no 6, November 1984.

8. Gonzalez and Woods, Digital Image Processing, Addison–Wellseley Publishing:Reading MA, June 1992.

9. Hildreth, E. C., "Implementation of a Theory of Edge Detection," MIT, April 1980

10. R. Horaud, T. Skordas, "Stereo Correspondence Through Feature Grouping and Maximal Cliques," *IEEE Trans. PAMI*, Vol 11 No 11, November 1989, pp 1158.

11. P.J.M. van Laarhoven and E.H Aart, Simulated Annealing: Theory and Applications, Kluwer Academic Publishers, Boston, 1988.

12. K.I. Laws, "Rapid Texture Identification," Proceedings of the SPIE Conference on Image Processing for Missile Guidance, San Diego, 1980, pp. 376–380.

13. J.C. McGaffigan, Application of Simulated Annealing to the General Image Point Correspondence Problem, Masters Thesis,WPI

14. T. Pavlidis, T.T. Liow, "Integrating Region Growing and Edge Detection." *IEEE Trans. PAMI*, Vol 12 No 3, March 1990, pp 225.

15. A. Rosenfeld, R.A. Hummel, S.W. Zucker, "Scene labeling by relaxation operation," *IEEE Trans. Syst. Man Cybern.*, vol. SMC–6,June 1976, pp. 420–423.

16. Y.C. Shah, R. Chapman, R.B. Mahani, "A New Technique to Extract Range Information From Stereo Images," *IEEE Trans. PAMI*, Vol 11 No 7, July 1989, pp 768.

17. D. Sherman, S. Peleg, "Stereo by Incremental Matching of Contours," *IEEE Trans. PAMI*, Vol 12 No 11, November 1990, pp 1102.

18. G.E. Sotak, Jr., K.L. Boyer, "The Laplacian–of–Gaussian Kernel: A Formal Analysis and Design Procedure for Fast, Accurate Convolution and Full–Frame Output," *Computer Vision, Graphics, and Image Processing*, 48, 1989, pp. 147.

19. H.L. Tan, S.B. Gelfand, E.J. Delp, "A Cost Minimization Approach to Edge Detection Using Simulated Annealing," *IEEE Trans. PAMI*, Vol 14 No 1, January 1992, pp. 3.