Pose Clustering Guided by Short Interpretation Trees

Clark F. Olson
University of Washington, Bothell
Computing and Software Systems
18115 Campus Way NE, Box 358534
Bothell, WA 98011-8246
cfolson@u.washington.edu

Abstract

It is common in object recognition algorithms based on viewpoint consistency to find object poses that align many of the object features with features extracted from a search image. Algorithms usually treat these features as having no information other than location. However, in many applications, the features are much more distinctive than this. This distinctiveness can be used to improve recognition with respect to both the search time and the reliability of the recognition. We modify an efficient clustering method for detecting objects using geometry to incorporate short trees that help prune many of the possible matches between object features and image features prior to the more expensive clustering step. The methodology is applied to a problem of computing a spacecraft position with respect to a celestial body by recognizing the configuration of craters visible on the surface.

1. Introduction

The recognition of objects using a geometrical model (for historical reasons, sometimes inaccurately called CAD-based vision) has fallen out of favor in the computer vision community. In large part, this is due to the requirement of an accurate geometrical model of the object. However, there are still many problems that can be solved using this framework. One example, of particular interest to us, is determining the position of a spacecraft with respect to a celestial body through recognizing the configuration of craters visible on the surface of the body.

Previous work has often assumed that the point features extracted from a model and an image contain no additional information. However, this is not usually the case. For example, in matching craters to a previously constructed object model, the crater position can be augmented with the radius and orientation in three dimensions. If an elliptical model for craters is used, the ratio of the major and minor axes provides additional information. In our work, we have found that the craters are largely round, although they appear elliptical from most viewpoints.

We build upon a previous pose clustering technique [6] that has been generalized for the detection of any parameterized model [7], through a combination with interpretation tree search. Our solution integrates pose clustering with ideas developed for searching an interpretation tree [4]. The interpretation tree is a data structure where each node in the tree represents a set of matches between the object features and the image features. The set represented by each node is the union of the set represented by the parent of the node and one additional match. Branches of the tree are pruned when a set of geometrically inconsistent matches have been hypothesized in some node.

Our method starts searching the interpretation tree, but stops at the third level of the tree and aggregates the match triples that remain at this level in separate groups that share pairs of matches. Randomization is used to reduce the overall number such pairs that need to be examined, similar to RANSAC [3] and other hypothesize-and-test methods.

2. Efficient pose clustering

Pose clustering [8] is a method of object recognition that builds upon the ideas of the Hough transform. A conventional application of the technique considers many hypothetical matches between small sets of model and image features. Each set yields a finite set of object poses that bring the features into alignment. Clusters of such poses in the pose space yield parameters that (nearly) bring many model features into alignment with image features and are, thus, good candidates for positions of the object.

Let k be the minimum number of matches between model features and image features for which a finite set of object poses brings them into alignment. Note that k is three for three-dimensional objects in arbitrary poses. We will call these minimal sets of matches *match sets*. A simple pose clustering method could examine all of the match sets with cardinality k, determine the poses that bring them into alignment, and then find clusters among these poses in the pose space. Randomization can be used to reduce the total number of poses that must be computed.

We build upon an efficient formulation of pose clustering that examines constrained subproblems [6]. It is not difficult to prove that subproblems that examine only match sets that share $k - 1$ of the same matches (called the *distinguished matches*) will yield essentially the same results as the full set of poses, if the distinguished matches are, in fact, correct matches. This leads to a method that is a hybrid of hypothesize-and-test methods and pose clustering. The distinguished matches are hypothesized (using randomization to limit the number of hypotheses) and pose clustering is used to test whether the distinguished matches are correct. Analysis has shown that this methodology has both a low complexity and high resistance to noise, occlusion, and clutter in the image.

3. Pruning using short trees

The pose clustering methodology described above assumes that any set of k matches can be brought into geometrical alignment by some pose of the object, but this is not the case if the features contain more information than a point in space. For example, if the points are oriented, then simply bringing the locations of the points into alignment is not sufficient for a match, since the orientations should also be in agreement.

We use a short tree to evaluate the feasibility as each match is added to the match set, pruning those that can be shown to be infeasible. See Figure 1. When the third level of the tree is reached, the method reverts to the pose clustering methodology described above. That is, each feasible set of three matches that share some pair of distinguished matches are clustered. The matches that share a pair of distinguished matches are simply those that are children of the same node at level two, since the third level is generated by adding additional matches to the sets at level two.

At this stage, we use the method of Huttenlocher and Ullman [5] to compute the poses (under weak-perspective) that bring the match sets at level three into alignment. Clustering is performed using a hierarchical binning method that examines the separate pose parameters in sequence. Each cluster in the previous parameter is expanded along a new parameter, keeping only the clusters are present in all of the parameters examined so far. This continues until the complete set of parameters has been examined.

Note that randomization is easily included in this framework. This corresponds to expanding only the nodes necessary to examine a randomly selected set of the nodes that would be present at level two of the tree. In this manner, only a fraction of the possible sets of distinguished matches need to be considered.

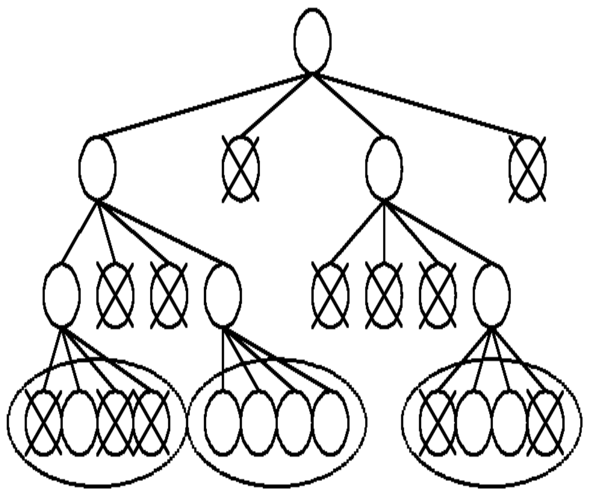


Figure 1. A short tree is used to find the sets of three matches that satisfy geometrical constraints. Pruning is performed at each level to remove sets that are infeasible. At level three of the tree, the sets that share two matches (they have the same parent) are clustered to determine if a single pose brings many of the sets into alignment.

Overall, the average complexity is highly dependent on how much pruning can be performed at each level of the tree. With randomization, the worst case complexity is $O(mn^3)$, where m is the number of model features and n is the number of image features. A complete analysis can be found in [6]. Pruning reduces the number of matches between model and image features that must be considered. This reduces the effective values of m and n.

4. Application to crater matching

In order to determine the position of a spacecraft with respect to a celestial body, we use the techniques described above to match the craters visible to the spacecraft to a previously catalog of craters on the body. Each crater is treated as an attributed point corresponding to the center of the crater, where the attributes are the radius and orientation of the crater. The radius and orientation have two dimensions in the image, but three dimensions in the crater catalog. The spacecraft pose is computed with the full six degrees of freedom.

The crater attributes are used to remove matches that are incompatible early in the search. In addition, an initial estimate of the spacecraft position is used to prune matches that are not feasible. The following subsections describe the methods used to prune sets of crater matches.

4.1. Crater pairs

For each crater detected in the image, the major axis of the ellipse detected corresponds to a cross-section of the crater. The ratio of the axes between two craters in the image must be the same as ratio of the crater radii in the catalog (modulo image noise and detection error). We eliminate pairs of craters if the ratio is not within 50% of the correct ratio from the catalog. We set this threshold to prune only those craters that are clearly wrong, since other tests will also eliminate many cases.

In addition, each pair of craters must be mutually visible from some viewpoint. We prune any pair of craters that has more than a 60 degree difference in orientation. While this constraint will prune a few more crater pairs than necessary, those that are pruned are less likely to yield good results, since at least one crater will be considerably foreshortened in the image.

4.2. Crater triples

If all three pairs of matches in the triple pass the previous test, we compute the poses that bring the crater centers into alignment. We can do further pruning on these poses. For example, if the pose requires that one or more of the craters is on the wrong side of the asteroid to be seen, then it can be pruned. We check to see whether the pose specifies that one of craters has an orientation greater than 75 degrees away from the camera. If so, then the pose is pruned, since the crater would be either on the wrong side of the asteroid or extremely foreshortened.

The estimated pose also tells us what size the crater should be in the image, what the ratio of the major and minor axis lengths should be, and the orientation of the crater in the image. These values are also used for pruning poses from consideration.

4.3. Pose filtering

If we have an initial estimate of the spacecraft position and an error covariance matrix, the pose estimation process can be made much more efficient by pruning the matches that are not consistent with the position estimate.

We represent the spacecraft orientation using a quaternion q and the position with a 3-vector t, so the overall spacecraft position is represented by 7 values (4 for the quaternion and 3 for the position), although only 6 are independent. Given the error covariance matrix, with values c_{ij}, for $1 \leq i, j \leq 7$, we can use covariance propagation methods to project the error covariances into an ellipse in the image space centered at the position given by the projection of the catalog crater according to the estimated spacecraft position.

Let $\hat{p}$ be the vector $[0\ p]$, so that we can use quaternion multiplication to rotate the vector. The equation that takes points from the asteroid frame of reference to the spacecraft camera frame of reference is:

$$p' = q\hat{p}q^* + t. \tag{1}$$

If we view the point from the spacecraft camera with focal length f, the image coordinates are:

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \begin{bmatrix} \frac{fp'_x}{p'_z} \\ \frac{fp'_y}{p'_z} \end{bmatrix} \tag{2}$$

The position covariance is propagated into the image coordinates through linearization by taking the partial derivatives of this equations with respect to the pose parameters (i.e. the Jacobian J).

The error covariance matrix in the image space is given by $C_i = JC_pJ^T$, where C_p is the covariance matrix in the pose space. We want to decide if an image crater is close enough to the estimated position of a catalog crater, so we calculate the error vector:

$$e = p_i - q\hat{p_d}q^* + t, \tag{3}$$

where p_i is the center of the crater in the image and p_d is the center of the crater in the catalog. If the errors yield a multi-variate normal distribution around the estimated point in the image, then we get a chi-squared test statistic (with 1 degree of freedom) using:

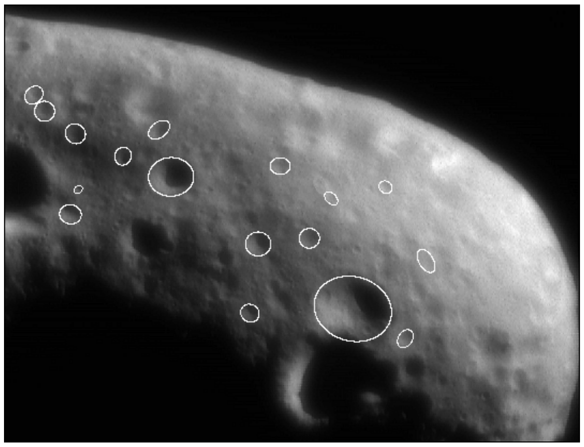
$$\chi^2 = e^T C_i^{-1} e. \tag{4}$$

If this test statistic is above a threshold (we use 3.841), then the crater can be eliminated from consideration.

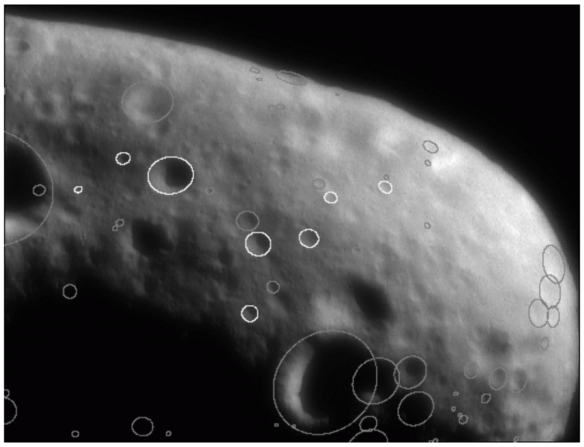
This test is used every time we consider a match between a particular image crater and a catalog crater. Over the course of the algorithm, matches are often considered several times. We further improve the efficiency by maintaining a look-up table for matches that have been previously considered, so that the computations need not be performed again.

5. Results

Figure 2 shows an example of a matching problem solved using this methodology. This example uses an image of the Eros asteroid captured during the NEAR



(a)



(b)

Figure 2. Example result using NEAR imagery of the Eros asteroid. (a) Craters detected. (b) Pose of asteroid computed after crater matching. Matched craters are white. Unmatched craters are grey.

(Near Earth Asteroid Rendezvous) mission [1]. A catalog of 955 major craters on the asteroid was constructed for the mission. We used this catalog for automatically estimating the spacecraft position off-line using images of Eros.

We, first, detected craters in the images [2]. The crater detection techniques found 17 craters in the test image shown. During the matching phase, the best cluster matched 8 of the detected craters to the catalog. The remaining 9 craters were not present in the previously constructed crater catalog. The algorithm required approximately 2 seconds of processing on a 333 MHz Sun UltraSPARC.

The techniques have been tested on dozens of additional images of Eros and Mars.

6. Summary

This work has examined methods to improve the efficiency of object recognition using pose clustering by incorporating a short tree search. The tree search examines individual matches, as well as pairs and triples of matches, between object and image features. When an incompatibility is found, the branch of the tree is pruned. Finally, matches are clustered in the pose space at the third level of the tree search. The resulting method is much faster than the original pose clustering technique. It has been successfully applied to spacecraft pose estimation by crater matching.

Acknowledgments

We gratefully acknowledge funding of this work by the NASA Intelligent Systems Program.

References

[1] A. F. Cheng, A. G. Santo, K. J. Heeres, J. A. Landshof, R. W. Farquhar, R. E. Gold, and S. C. Lee. Near-Earth asteroid rendezvous: Mission overview. *Journal of Geophysical Research*, 102(E10):23695–23708, October 1997.

[2] Y. Cheng, A. E. Johnson, L. H. Matthies, and C. F. Olson. Optical landmark detection and matching for spacecraft navigation. In *Proceedings of the 13th AAS/AIAA Space Flight Mechanics Meeting*, February 2003.

[3] 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*, 24:381–396, June 1981.

[4] W. E. L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.

[5] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.

[6] C. F. Olson. Efficient pose clustering using a randomized algorithm. *International Journal of Computer Vision*, 23(2):131–147, June 1997.

[7] C. F. Olson. A general method for geometric feature matching and model extraction. *International Journal of Computer Vision*, 45(1):39–54, October 2001.

[8] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40:361–387, 1987.