Brigham Young University
BYU ScholarsArchive

Faculty Publications

2005-06-01

Correspondence Expansion for Wide Baseline Stereo

Parris K. Egbert
egbert@cs.byu.edu

Kevin L. Steele

Follow this and additional works at: https://scholarsarchive.byu.edu/facpub

Part of the Computer Sciences Commons

Original Publication Citation

Kevin Steele and Parris K. Egbert, "Correspondence Expansion for Wide Baseline Stereo." In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 25), San Diego, CA, Vol 1, pp.155-161, (June 2-26, 25).

BYU ScholarsArchive Citation

Egbert, Parris K. and Steele, Kevin L., "Correspondence Expansion for Wide Baseline Stereo" (2005). *Faculty Publications*. 375.
https://scholarsarchive.byu.edu/facpub/375

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.

Correspondence Expansion for Wide Baseline Stereo

Kevin L. Steele and Parris K. Egbert
Department of Computer Science, Brigham Young University
3361 TMCB, Provo, UT 84602
{steele, egbert}@cs.byu.edu

Abstract

We present a new method for generating large numbers of accurate point correspondences between two wide baseline images. This is important for structure-from-motion algorithms, which rely on many correct matches to reduce error in the derived geometric structure. Given a small initial correspondence set we iteratively expand the set with nearby points exhibiting strong affine correlation, and then we constrain the set to an epipolar geometry using RANSAC. A key point to our algorithm is to allow a high error tolerance in the constraint, allowing the correspondence set to expand into many areas of an image before applying a lower error tolerance constraint. We show that this method successfully expands a small set of initial matches, and we demonstrate it on a variety of image pairs.

1. Introduction

Reliable feature matches between wide baseline image pairs are important for many stereo algorithms in computer vision. Typical feature types include points, lines, curves, and textured regions. Correct feature correspondences enable stereo camera calibration and structure-from-motion algorithms, and permit robust estimation of epipolar geometries between two or more images. Epipolar geometries in turn facilitate further feature matching, image rectification, and the finding of dense image correspondences.

Finding a sufficiently large number of correct feature correspondences between image pairs can determine the success or failure of stereo algorithms that rely on plentiful matches. It is therefore important to be able to generate many correct, high confidence matches from images in a reasonable amount of time. While there exist many wide baseline matching algorithms, most address the problem of finding matches independently; fewer use existing matches in a guided search for more [14, 17, 19].

The most common method for finding point correspondences between two wide baseline images is to first identify points on each image that lend themselves well to matching. Such *interest points* often have characteristics such as high intensity variance and anisotropic texturing in surrounding pixels. Two sets of interest points are then tentatively matched to one another by finding similar feature vectors between points, yielding a set of putative matches. The feature vectors often include correlation measures [3], and geometric and photometric invariances [16].

If each image of the pair has N interest points, the matching complexity is $O(N^2)$. Unfortunately, this common N^2 method of finding a set of feature matches will usually result in many mismatches, due mainly to sampling noise, lighting changes, and foreshortening effects. A robust epipolar geometry estimator such as RANSAC [7] is frequently used as a final step to eliminate the outlying matches. However, if there is a high percentage of incorrect matches given to the estimator, RANSAC executes very slowly. Even worse, matches failing the epipolar constraint are simply discarded, greatly reducing the size of the final correspondence set. A contribution of our work is that we exploit the near-correctness of these discarded matches, locally adjusting their positions to conform to the constraint (see section 3.4).

Our goal is to aggressively search for additional matches using existing matches as starting locations, and to make the final correspondence set as large as possible while preserving the accuracy of its member matches.

1.1. Related Work

Many robust methods exist to create point matches for wide baseline stereo. Baumberg [1] finds affine-invariant features by extracting the relative skew,

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

stretch, and rotation from interest point neighborhoods, and matches points with similar image structure. Applying the extracted affine transformation to the sampling window reduces the number of incorrect matches in the final set of correspondences. Tuytelaars and Van Gool [20] use local affine and photometric invariant features of the points to facilitate matching. An elliptical region surrounding an interest point is examined to find its generalized color moments, which comprise the invariant feature vector. Interest points are matched according to feature vector similarity. Schaffalitzky and Zisserman [19] find texture region matches using affine and photometrically invariant descriptors. Their method is statistically insensitive to the shape of the region, yielding a more stable match descriptor than point-based matches provide. Mikolajczyk and Schmid [15] find affine-invariant feature points by first detecting multi-scale Harris points, and then use these points in an iterative procedure until the points converge to affine invariance. Both the relative scale and the shape of the point neighborhoods are recovered simultaneously.

Matas *et al.* [14] find corresponding regions using an improved similarity measure that adds a voting scheme to the commonly-used Mahalanobis distance. Their method also improves on the large-scale invariance of [16]. Ferrari *et al.* [4] propose a method to utilize multiple images (more than two) to establish point correspondences between all images. We focus on using only two images, and our method for correspondence expansion is independent of the type of comparison function used to score the fitness of a match. In fact, any method previously used to identify feature matches can be leveraged to initialize our algorithm, and correspondence expansion can easily be appended to any existing matching scheme to increase the number of final matches.

Several methods have been proposed that use known matches to guide the search for additional point matches. Lourakis *et al.* [12] find point and line feature correspondences on a common plane by using a randomized search strategy to find an initial set of point and line matches. They then use the derived homography of three lines to verify the point locations and to predict the location of further matches. Their method relies on the presence of planar features, while our algorithm makes no assumptions on geometric properties. Pritchett and Zisserman [17] compute local homographies at existing matches to guide the search for new matches. They first use existing homographies to predict match locations and then employ a hierarchical approach to create new homographies to carry out additional searches. Our algorithm uses local affine transformations to guide searches, rather than homographies. Matas *et al.* [14] improve their number of matched regions by finding affine transformations of correspondences that survive a preliminary RANSAC cull. They then include those portions of regions whose transformed correlation are above a pre-selected threshold in a second RANSAC cull. This roughly doubles or triples the number of correspondences from simply using a single RANSAC prune. A major contribution we make beyond both [17] and [14] is to apply an epipolar constraint at each iteration, rather than once, as described in section 3.2, enabling many more correct matches to be found.

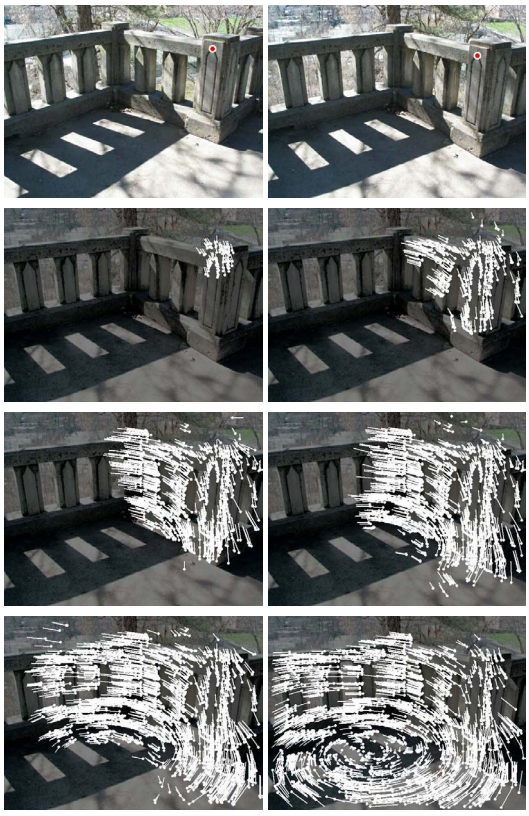

Figure 1: An example of correspondence expansion occurring at each step of the iterative algorithm. The top two images of a wall are the left and right images to be matched. One initial point was selected by a user, shown as a red and white circle in the top two images. As matches are found and added to the correspondence set, the matches "grow" outward from the original point, and incorrect matches quickly disappear. Point correspondences are shown as white lines in the image sequence. The last image in the sequence shows the final correspondence set containing 1,925 matches. The lower six images have been darkened to better highlight the correspondences.

Ferrari *et al.* [5, 6] introduce a method that selects an initial set of feature correspondences as anchors to an iterative exploration of the surrounding image areas. The resulting correspondence set is able to detect matches between images exhibiting non-rigid deformations, and they use this ability as input to an object recognition system. Our proposed method is similar to that of Ferrari *et al*. in that we iteratively augment and constrain our expanding match set (see section 3). However we employ a novel use of the epipolar constraint (section 3.2) that allows a high growth rate while retaining an approximate epipolar geometry, then refine the final match set to a correct epipolar geometry upon completion. This is in contrast to the method of Ferrari *et al.* in which they purposefully avoid epipolar constraints in order to permit non-rigid deformations.

Finally, Schaffalitzky and Zisserman [19] improve on the number of matched regions they find by matching pixels within matched regions. This approach is very successful in generating large numbers of correctly matched points. Their method is limited, however, to isotropic texture regions. In contrast, we are not limited by any statistical property of the input images.

1.2. Contribution

In this paper we present a new method that expands an initial set of wide baseline correspondences by an iterative two-step process. We do not propose a new method of establishing initial feature matches. Rather, we propose a novel technique to iteratively grow a set of correspondences outward from a small initial set of matched points.

We use a local affine transform approximation to predict search locations near existing matches. Newly found matches are incrementally added to the correspondence set, and the expanded correspondence set is refined using a high error tolerance RANSAC measure. The result is an expanding set of high confidence correspondences that "grow" outward from existing matches, as shown in figure 1. We discuss the details of our algorithm in the next sections. Section 2 briefly describes the initialization of the algorithm, section 3 discusses how the process iteratively expands the correspondence set, section 4 shows the results of the method, and we conclude in section 5.

2. Initial Correspondence Set

Prior to employing the correspondence expansion algorithm, we must have a potentially small set G of putative matches between two source images I_1 and I_2. These matches need not all be correct. For correspondence expansion to work, at least one match needs to be correct. The more correct matches contained in G, the faster the algorithm will perform. Note that a single initially correct match is a necessary, but not sufficient condition, i.e., given at least one correct match, the algorithm provides no guarantee that matches will be expanded. We have found in practice that this is of no practical concern, since generally there are several correct matches in an initial correspondence set, all of which tend to expand quickly as the algorithm proceeds.

As discussed in the introduction, there have been many wide baseline feature correspondence algorithms proposed over the past several years (see [4, 8] for references to more algorithms), and any of these could be used to create G. These matches could also be input by a user if desired. While not novel, we briefly mention the initialization procedure we used to create G.

We start by detecting Harris corners [9] in each of the two source images I_1 and I_2. We employ the often-used $O(N^2)$ scheme of comparing each corner point detected in I_1 to every corner point detected in I_2. We also determine the relative local rotation between image patches surrounding the corner points, and following [2] we attempt to match at several resolutions to find a characteristic scale between image regions.

We measure similarity by taking the sum of the squared differences between pixels in the local image region, and assign as matches point pairs with the highest similarity. Since the matching assignment may be a many-to-one mapping, point pairs with the highest similarity are bi-directionally checked, making it an $O(N^3)$ procedure. Matches passing the bi-directional comparison check are finally added to G. While not optimal, for a small number of initial corner points the computation time is negligible when run on a modern processor, exploiting a strength of our algorithm of not needing many initial matches.

3. Guided Matching

With a set G of initial matches, we employ our correspondence expansion algorithm to grow the set to include additional matches. The expansion algorithm is iterative, and adds matches to the correspondence set at each step. There are two parts to each iterative step: *aggregation* and *constraint*. In the aggregation step, we use the current set of matches as seed points to "grow" additional matches that are nearby, adding the new matches to the current set. In the constraint step, we constrain the newly-enlarged correspondence set to an epipolar geometry, so that when the points in the set are

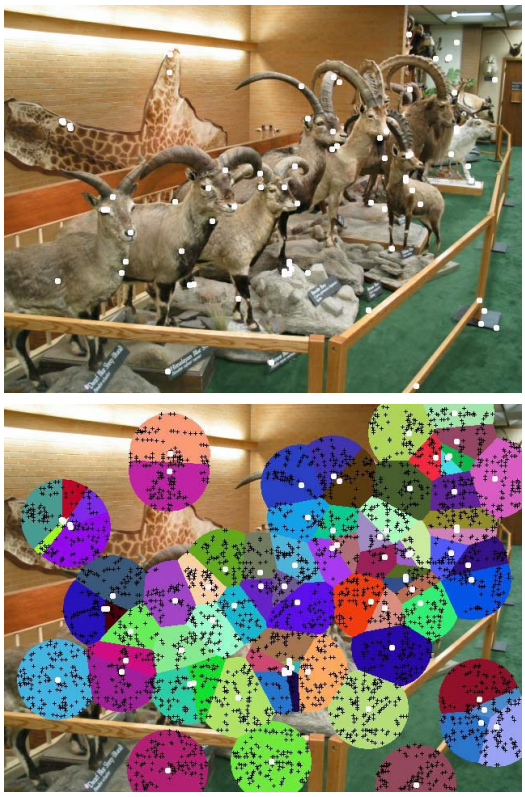

Figure 2: Voronoi regions for a set of existing correspondences. The top image of a taxidermy display contains a set of points, highlighted in white, matched to points in another image of the same scene (not shown). The bottom image shows the same set of points surrounded by their corresponding Voronoi regions—each separate region is highlighted in a different shade. The black Harris corners within each Voronoi region will potentially be matched using the local rotation and scale of the nearest existing match.

used as seed points in the next iteration, they will have a higher likelihood of growing correct matches. Important to the success of the algorithm is the need for a high error tolerance on the epipolar geometry, as will be explained in section 3.2.

3.1. Aggregation

Before beginning the iterative cycle, we detect a set of several thousand Harris corners P_1 in I_1. These serve as interest points which will be matched together with locations in I_2 to form correspondences as the iteration proceeds. We also maintain a current correspondence set C which is initialized to the original match set G.

For each point p_i in P_1, we find the nearest point c_{1i} in C; c_{1i} has already been matched to a point c_{2i} in I_2. To quickly find nearest points, Voronoi maps over I_1 and I_2 are constructed for the points in C. We use graphics hardware to quickly build Voronoi diagrams by rendering cones into two depth buffers, one each for I_1 and I_2 [11, 21]. The cones are centered at each point in C and have a finite base. The colors of the rasterized cones determine the identity of the Voronoi regions. In this way Voronoi regions can be looked up from a 2D location in constant time. Figure 2 shows an example of the Voronoi regions for an image.

Having found the closest matched point c_{1i} in C to the unmatched feature point p_i, we compute an affine transformation that maps the image region surrounding c_{1i} to the region surrounding c_{2i}. Baumberg observed that small planar surface patches undergo affine transformations when seen from different viewpoints [1], and that non-planar smooth patches can successfully be approximated by planar surface patches for correlation. Rather than estimate the whole affine transform which, given the match location in C amounts to finding 4 parameters, we only consider the local rotation and scale.

Rotation is estimated by using the best correlation from a small set of candidate rotations. A precomputed lookup table is used to accelerate the rotated locations of each pixel in the window. We could use nearby matches to estimate the rotation, or even the full affine transformation. However, local 2D rotations differ greatly across the image due to the projection of 3D camera rotation, so matches not in the immediate neighborhood of c_{1i} yield incorrect rotations. Local scale is less susceptible to the effects of 3D camera rotation, so we compute it directly from a nearby match. We first find the closest match c_{1j} to c_{1i} in C, then using these two existing matches the local scale is the ratio of their distances (figure 3).

Once the local rotation and scale are estimated, the feature point p_i in I_1 is transformed to a new location p' in I_2 (not necessarily a corner point). A steepest-ascent hill-climbing strategy is used to find the best match in I_2—the correlation at p' is compared with the correlation at all the pixel neighbors, and p' is moved until a local maximum is reached.

With the putative match identified, the whole process is reversed, where the inverse rotation and scale are used to transform p' back to I_1 to predict the original location of p_i in I_1. Again the hill-climbing strategy is used to find the best match p'' in I_1. If the original feature point p_i and the point p'' are within a threshold distance (we use 1 pixel), then the match $[p_i, p']$ is considered valid and added to the set C of current correspondences, and p_i is removed from the set P_1 of feature points.

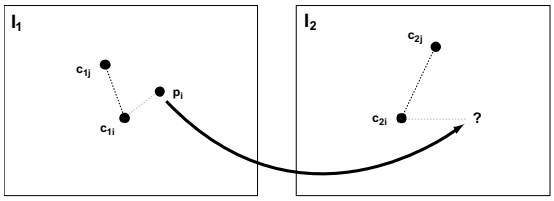

Figure 3: The local transform is used to predict new match locations. The feature point p_i was detected using the Harris corner detector, and its closest existing matching point c_{1i} provides a local rotation and scale to guide the search for a match for p_i in I_2. The local rotation is found by maximizing the correlation from a set of candidate rotations. To find the local scale around c_{1i}, its closest match c_{1j} is located, then the scale to transform p_i to I_2 is computed as the ratio of the distances: $|c_{2i} - c_{2j}| / |c_{1i} - c_{1j}|$.

After all feature points in P_1 have been processed in this manner, C will potentially be much larger. The rate at which matches are added to C depends on the size of the cones determining the Voronoi regions, the number of Harris corners detected within the Voronoi region of each existing match, and the heterogeneity of the texture surrounding the detected corners. In images containing a large portion of high frequency detail, we have found that C increases in size by 50% to 200% at each iteration until the saturation point is reached (discussed in section 3.3).

3.2. Constraint

It is imperative that C contain many correct matches, since these are used to seed the growth of additional matches in subsequent iterations. To further ensure that most or all matches in C are correct, they are constrained to an epipolar geometry. It has been well demonstrated that a robust epipolar geometry can be determined from a set of putative matches using RANSAC [7, 13]. RANSAC is an iterative algorithm, and the number of iterations needed can be automated as shown in [10]. The generation of the epipolar geometry using this algorithm also serves to effectively segment, or cull, correct correspondences from incorrect, outlying correspondences. It is for this second purpose that we employ the RANSAC algorithm.

Even though the RANSAC algorithm is robust, care must be taken to avoid too many incorrect matches in the input set, as the number of iterations required will quickly grow very large. For instance, given an input set with an estimated 75% outliers, the number of iterations required to ensure correct segmentation of inliers from outliers with 95% probability will be about 50,000, as given by the following equation from [10]:

$$N_{iterations} = \log(1-.95) / \log(1-(1-\varepsilon)^7) \quad | \quad \varepsilon = .75$$

When generating the consensus sets during RANSAC culling, we intentionally use a high inlier error tolerance—matches within 5 pixels of their epipolar lines are considered inliers. This is an important aspect of our algorithm. Though it results in a less accurate epipolar geometry, it permits many more matches to be added to C. By including more matches in this way even if they are slightly incorrect, we speed up the inlier/outlier segmentation considerably, giving the algorithm a fast iteration cycle, and more importantly, improving the ability to grow more matches in the next iteration. Thus, at this stage of the algorithm, the set C temporarily contains a large number of incorrect matches due to the high error tolerance. However, as a result, the algorithm is able to create an average of 50% more final correct matches in our test images than it does by using a low RANSAC error tolerance, such as .5 or .1 pixels. This is because the high error tolerance permits matches to expand into regions of I_1 and I_2 that otherwise would have contained fewer candidate matches.

Inlier matches that survive the epipolar constraint are kept in the set C of current correspondences. Matches that fail are removed from the set, and each point c_{1i} from the failed matches are placed back in the set P_1 of feature points for future matching consideration.

It is important to note that we do not use the epipolar constraint to guide the search for new matches, as it has been used historically. Doing so would potentially contaminate the correspondence set with false matches following an incorrectly estimated epipolar geometry. Rather, as explained in section 3.1, we use an approximated affine transform to guide the search, and we utilize the epipolar constraint to refine the augmented correspondence set along the way.

3.3. Saturation

The correspondence expansion iterations are allowed to proceed to a saturation point, when no additional matches are added to C in the aggregation step. This occurs when all the feature points in P_1 either have been matched or have no correlating matches that can be found. We have observed that during the aggregation and constraint phases, the size of C may occasionally drop slightly as the matches it contains shift to a more accurate epipolar geometry. Immediately fol-

Im#	Initial Matches	Final Matches	Time in Secs.	Itera-tions
1	53	144	33	11
2	25	189	29	14
3	29	270	20	19
4	37	480	28	17
5	53	531	46	14
6	53	531	55	15
7	35	607	43	13
8	52	661	39	13
9	63	818	37	11
10	98	849	48	11
11	27	871	29	15
12	100	908	42	12
13	46	1208	34	13
14	81	1307	25	10
15	58	1401	36	18
16	109	1916	33	13
17	66	1942	43	17

Table 1: Results from the correspondence expansion algorithm, sorted by the number of final matches found. The initial matches were computed using the initialization procedure of section 2. The Final Matches column reports the number of matches found after expansion. The average expansion time in seconds for all images in the table is 36.5 seconds; this time does not include finding the initial match set. The average number of iterations is 14. All tests were run on a 3.2 GHz Pentium 4 CPU.

lowing such adjustments, the size of C usually increases dramatically since the increased accuracy will admit more correct matches. To permit these desirable fluctuations, we allow the size of C to drop a predetermined number of times (we use 3) before terminating the iteration cycle. This conservative termination criterion ensures that correspondence expansion is able to advance into most areas of I_1 and I_2 that would be reached by an unbounded number of iterations.

3.4. Final Guided Matching

At its saturation point, the current correspondence set C contains many more matches than it did initially; however, the matches do not adhere closely to the epipolar constraint as a result of the high error tolerance permitted earlier in the algorithm. As a final step, we wish to impose a tighter epipolar constraint to ensure a correct set of matches. Unfortunately, applying the constraint would eliminate many matches in C that are close to correct, but are far enough away from their epipolar lines to fail the epipolar constraint.

Rather than discard all of these near-correct matches, we adjust their matched positions prior to the final constraint application. We use a simple guided search strategy that, in contrast to section 3.2, does use the epipolar constraint as a guide. Matched points c_{2i} from C in I_2 are projected orthogonally to their corre-

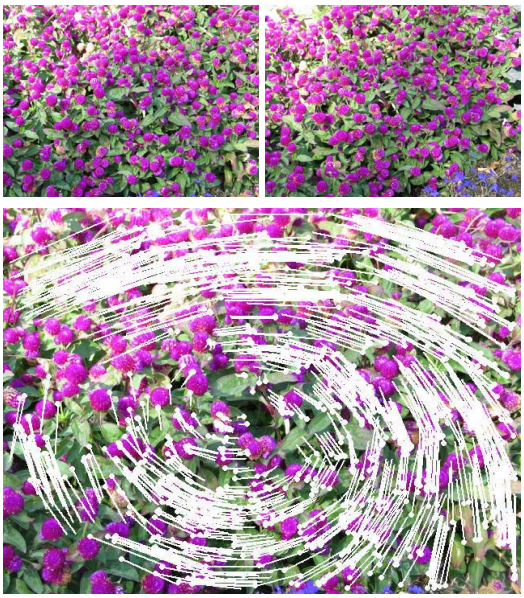

Figure 4: Flowerbed image pair. This correspondence set was expanded from 35 initial matches to 931 final matches. The top images are the original pair, and the bottom image illustrates the final correspondences with lines.

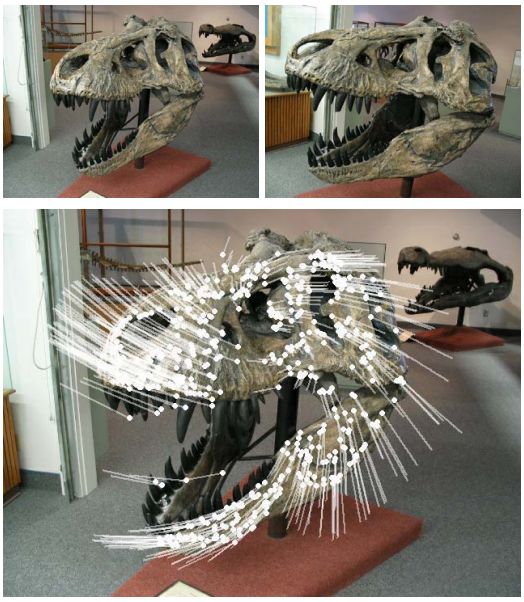

Figure 5: T-Rex skull image pair, expanded from 30 initial matches to 532 final matches. Correspondences are shown with white lines.

sponding epipolar lines, then their counterpart points c_{1i} are matched to points in I_2 along short segments of the epipolar lines. Those matches with better correlations replace the old matches, while those not having better correlations are discarded. Finally, the set C is constrained to an epipolar geometry with a low error tolerance of .5 pixels to create the final correspondence set.

4. Results

We tested the correspondence expansion algorithm on images acquired from a variety of indoor and outdoor environments. Starting with a few dozen matches, our algorithm performed extremely well, expanding the correspondence set by approximately 3 to 30 times its original size. We also found that there were very few incorrect final matches.

To measure the accuracy of the matches, we hand-picked a small set of correspondences in each image pair to compute an accurate fundamental matrix **F**. We then started the expansion algorithm from a different, small set of automatically derived initial matches, and tested the expanded set against **F** to measure the match distances from their respective epipolar lines. We measured this accuracy on a set of 17 image pairs. The average error for the hand-picked correspondences was 0.766 pixels, and the average error for the expanded set of correspondences was 1.862 pixels, with a standard deviation of 1.18. This indicates a high accuracy for the expanded correspondence set, considering that the point matches are not made to sub-pixel precision. Note also, that this measure does not check for mismatches which lie along correct epipolar lines.

Table 1 reports the number of matches found, the expansion run-time in seconds, and the number of iterations needed for the 17 images used in the accuracy check. Figures 4 - 6 show examples of the expansion algorithm finding matches in image pairs.

5. Conclusion and Future Work

We have presented a method to expand an initial set of wide baseline correspondences to many times its original size. Using an iterative two-step process, we first aggregate additional matches around existing "seed point" matches. An approximate affine transformation (translation, rotation, and scale) maps the points of the seed match and is used to predict new match locations. Second, we constrain the aggregated matches to a high error tolerance epipolar constraint using RANSAC. These steps are iterated until no matches are added to the expanded set. Using correspondence expansion successfully yields many more wide baseline matches than are obtained using previous methods alone. Our algorithm does not replace previous matching algorithms; rather it augments existing methods as a "post-process" to increase the number of final, high quality correspondences.

Figure 6: Shelf image pair exhibiting scale change and camera cyclo-rotation. This set was expanded from 19 initial matches to 112 final matches.

We presently do not consider photometric differences between most image pairs while matching. To match images with significant lighting change, we perform intensity histogram equalization in YIQ space as a preprocess to correspondence expansion. While this reduces photometric differences and performs fairly well in practice, we would like to use an invariant measure. A possible solution is to use color moments as in [16, 20], however any local measure will impose a large computational burden on our algorithm. A pre-processed contrast and brightness normalization procedure such as in [18] may be a more efficient approach.

References

[1] A. Baumberg. Reliable feature matching across widely separated views. In *Proc. CVPR*, pages 774-781, 2000.

[2] Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *Proc. CVPR*, pages 612-618, 2000.

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

1063-6919/05 $20.00 © 2005 **IEEE**

[4] V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline multiple-view correspondences. In *Proc. CVPR*, pages I-718 - I-725, 2003.

[5] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. *ECCV* vol. 1, pages 40 - 54, 2004.

[6] V. Ferrari, T. Tuytelaars, and L. Van Gool. Integrating multiple model views for object recognition. In *Proc. CVPR*, pages 105 - 112, 2004.

[7] M. Fischler and R. Bolles. RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Comm. ACM* 24(6), pages 381-395, 1981.

[8] T. Goedemé, T. Tuytelaars, and L. Van Gool. Fast wide baseline matching for visual navigation. In *Proc. CVPR*, pages 24-29, 2004.

[9] C. Harris and M. Stevens. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conf.*, pages 147-151, 1988.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[11] K. Hoff III, J. Keyser, M. Lin, D. Minocha, and T. Culver. Fast computation of generalized voronoi diagrams using graphics hardware. In *Proc. SIGGRAPH '99*, pages 277-286, 1999.

[12] M. Lourakis, S. Halkidis, and S. Orphanoudakis. Matching disparate views of planar surfaces using projective invariants. In *Proc. BMVC*, pages 94-104, September 1998.

[13] Y. Ma, S. Soatto, J. Košecká, and S. Shankar Sastry. *An Invitation to 3-D Vision*. Springer-Verlag, 2004.

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, pages 384-396, 2002.

[15] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. *ECCV* vol. 1, pages 128-142, 2002.

[16] F. Mindru, T. Moons, and L. Van Gool. Recognizing color patterns irrespective of viewpoint and illumination. In *Proc. CVPR*, pages 368-373, 1999.

[17] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. ICCV*, pages 754-760, Jan. 1998.

[18] P. Sand and S. Teller. Video matching. Tech. Report LCS TR 947, MIT, 2004. http://graphics.csail.mit.edu/~sand/vid-match/tech-report.html

[19] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proc. ICCV*, pages 636-643, 2001.

[20] T. Tuytelaars and L. Van Gool. Wide Baseline Stereo based on local, affinely invariant regions. In *Proc. BMVC*, pages 412-422, 2000.

[21] M. Woo, J. Neider, and T. Davis. *OpenGL Programming Guide, Second Edition*. Addison Wesley, 1997.