ITERATIVE DENSE CORRESPONDENCE CORRECTION THROUGH BUNDLE ADJUSTMENT FEEDBACK-BASED ERROR DETECTION

Mauricio Hess-Flores[1], Mark A. Duchaineau[2], Michael J. Goldman[3] and Kenneth I. Joy[4]
[1,4]*Institute for Data Analysis and Visualization, University of California - Davis, Davis, CA, U.S.A.*
[2,3]*Lawrence Livermore National Laboratory, Livermore, CA, U.S.A.*

Keywords: Dense correspondences, Pose estimation, Scene reconstruction, Bundle adjustment, Resolution pyramid, Error analysis.

Abstract: A novel method to detect and correct inaccuracies in a set of unconstrained dense correspondences between two images is presented. Starting with a robust, general-purpose dense correspondence algorithm, an initial pose estimate and dense 3D scene reconstruction are obtained and bundle-adjusted. Reprojection errors are then computed for each correspondence pair, which is used as a metric to distinguish high and low-error correspondences. An affine neighborhood-based coarse-to-fine iterative search algorithm is then applied only on the high-error correspondences to correct their positions. Such an error detection and correction mechanism is novel for unconstrained dense correspondences, for example not obtained through epipolar geometry-based guided matching. Results indicate that correspondences in regions with issues such as occlusions, repetitive patterns and moving objects can be identified and corrected, such that a more accurate set of dense correspondences results from the feedback-based process, as proven by more accurate pose and structure estimates.

1 INTRODUCTION

The computation of dense image correspondences has been of great importance recently in several Computer Vision applications. For example, the registration of succesive frames in a video sequence into a common coordinate system at the pixel level enables applications such as dense 3D reconstruction of the viewed scene, segmentation of background and moving objects and data compression. The main problem with the use of dense correspondences is that inaccuracies in their estimation can arise whenever there are certain conditions such as occlusions and moving objects present in the scene, and also in regions with little texture or repetitive patterns. Such conditions do not necessarily affect algorithms for sparse feature matching, but certain applications, like the ones mentioned, strictly call for the use of dense correspondences, and any of these adverse conditions ultimately affect the quality of such applications.

To this end, this paper describes a novel method for detecting and correcting inaccurate dense correspondences, giving the proof of concept for the case of two input images. The detection and correction mechanism is enabled by feedback after estimating camera poses and scene structure from the two views and applying bundle adjustment. Using reprojection error after bundle adjustment as the metric to separate high-error and low-error correspondences, an affine neighborhood-based coarse-to-fine iterative algorithm is applied to correct high-error correspondences. The main assumption is that the input dense correspondence set must be unconstrained; for example it cannot have been generated from techniques such as guided matching (Hartley and Zisserman, 2004) for the algorithm to work. The reprojection error metric used to detect errors has no meaning for correspondences constructed assuming a perfect fit of these to a given epipolar geometry, as will be detailed later. An important motivation for using feedback after bundle adjustment is to avoid applying the correction mechanism to all available correspondences, which would result in $10-20x$ slower processing times during this phase. While it is not the objective of the work presented here to explicitly solve for the occlusion problem in reconstruction or detect moving objects, the end goal is to achieve the best possible correspondence accuracy in such problem areas, even if it implies a higher computational expense, which makes it important to apply only where neces-

Hess-Flores M., A. Duchaineau M., J. Goldman M. and I. Joy K. (2010).
ITERATIVE DENSE CORRESPONDENCE CORRECTION THROUGH BUNDLE ADJUSTMENT FEEDBACK-BASED ERROR DETECTION.
In *Proceedings of the International Conference on Computer Vision Theory and Applications*, pages 400-405
DOI: 10.5220/0002816104000405

sary. Experimental results on real and synthetic data sets indeed show an overall improvement in the accuracy of the dense correspondence set after applying the procedure.

Error detection for dense correspondences has been done in the past, but either under simplifying assumptions or with respect to ground-truth data. In (Xiong and Matthies, 1997), matching errors are identified and corrected, but only one specific scene type is handled. The algorithm in (Mayoral and Aurnhammer, 2004) evaluates matching algorithms by introducing an error surface from matching errors. In both cases, the simplifying assumption of searching for disparity along scanlines is made. An exhaustive overview and evaluation of dense correspondence algorithms is given in (Scharstein and Szeliski, 2002), though the comparisons are done with respect to ground-truth values. As for error correction, an algorithm known as optimal triangulation (Hartley and Zisserman, 2004) makes an attempt to correct correspondences based on the pre-computed epipolar geometry between the scenes. However, such a correction, while mathematically correct and obtained by minimizing a geometrically meaningful criterion, does not necessarily produce matches that are correct in reality; it also reduces reprojection error after reconstruction to zero, thus preventing error detection using such a criteria.

An initial reconstruction of the scene from the two input views is needed as part of the algorithm, so a brief overview of the relevant literature on this subject is now given. In general, a reconstruction pipeline consists of obtaining matches (correspondences) between the images, then computing the relative camera poses between them and finally computing the structure of the scene. The matches used for the initial pose estimation can either be sparse features (for example corners) or dense correspondences, which assign a correspondence in a destination image to each source image position, and can be computed through a variety of methods (Scharstein and Szeliski, 2002). For two views, the epipolar geometry between them, encapsulated by the fundamental matrix F (Hartley and Zisserman, 2004), can be computed from the initial matches. This matrix can be computed through direct methods, such as in (Stewénius et al., 2006; Hartley and Zisserman, 2004) as well as through non-linear methods (Hartley and Zisserman, 2004). The RANSAC algorithm can be coupled with these methods to help obtain more robust estimates for F. Using the computed epipolar constraints, more matches can be generated across the images to obtain dense correspondences (details can be found in (Hartley and Zisserman, 2004)). Again, an issue with such constrained correspondences is that the new matches depend directly on the quality of the estimated epipolar geometry, making them mathematically valid but not necessarily correct.

Once matches are available, either sparse or dense, the relative pose (rotation and translation) between the cameras viewing the scene can be computed. Several methods exist, and an overview of different pose estimators is given in (Rodehorst et al., 2008). In the particular case that the F matrix is available or has been computed from matches, and if the camera's intrinsic parameters (such as the focal length, skew and principal point) are assumed known, the essential matrix E can be computed and decomposed into the relative rotation and translation. Finally, the scene's 3D structure can be obtained using the available sparse or dense matches. Typically, linear or optimal triangulation (Hartley and Zisserman, 2004) is applied on each correspondence pair to generate a 3D position corresponding to the scene structure. Once pose and structure estimates are available, a common fine-tuning step for both estimates is to carry out a *bundle adjustment*, where the total reprojection error of all computed 3D points in all cameras is minimized using non-linear techniques (Hartley and Zisserman, 2004). Fortunately, sparsity in the data has allowed for great speed-ups in this process (Lourakis and Argyros, 2000).

By coupling the use of unconstrained dense correspondences in a bundle-adjusted reconstruction pipeline, a novel mechanism to identify the most inaccurate dense correspondences and correct them using an iterative method can be achieved. The entire procedure will be described in detail in Section 2, followed by experimental results (Section 3) and conclusions (Section 4).

2 PROPOSED ALGORITHM

2.1 Pose and Structure Estimation based on Dense Correspondences

The first step in our algorithm is to compute unconstrained dense correspondences between two images, for which a sub-pixel accuracy direct method which solves coarse-to-fine on $4-8$ mesh image pyramids with a $5x5$ local affine motion model was used, as outlined in (Duchaineau et al., 2007). There are several reasons for starting out with such a general-purpose dense correspondence algorithm. First of all, our intended applications, such as dense scene reconstruction and image stitching, call for the use of dense

as oppposed to sparse matching. Now, by not using epipolar constraints as in guided matching, it allows for errors in the dense correspondences to be unmasked in later stages. Additionally, it is a more general approach that adequately samples the scene; for example sparse feature matchers could fail to find a significant amount of features in regions with little intensity variation, whereas dense correspondences could still be obtained. However, as mentioned earlier dense correspondences are prone to errors resulting from occlusions, moving objects, texture-less regions and repetitive patterns. For now, the next steps (pose and structure estimation) must proceed despite these errors, but it will be explained in Section 2.2 how these issues can be respectively detected and corrected through a novel mechanism based on feedback.

The first step in estimating the relative pose between the two cameras is to estimate the 3*x*3 fundamental matrix F, which encapsulates the epipolar geometry between the two views. The direct and robust 5-point method (Stewénius et al., 2006) embedded in RANSAC is currently being used. It is important to mention that even though a large amount of correspondences are available for estimating F, only a small number are actually needed. Even if the minimal amount is used, the use of RANSAC coupled with random sampling ensures that a reliable F can be estimated in a computationally-efficient yet accurate manner. Now, the essential matrix E is obtained from the fundamental matrix, assuming known intrinsic parameters for the camera, and factorized into the rotation and unit translation (R,t) pair representing the pose. To obtain the scene structure as a set of 3D points for each correspondence pair, linear triangulation was used. A dense scene structure must be computed, since it will be used as part of the error detection and correction mechanism based on feedback that will be described later on.

The objective of the next step, bundle adjustment, is to adjust pose and structure estimates in such a way that the total reprojection error of the 3D points with respect to their corresponding 2D correspondences in each camera is minimized (Hartley and Zisserman, 2004). The cost function which is traditionally minimized can be expressed as the sum of squares of the geometric (reprojection) error between each 3D point and the correspondences which yielded it, as shown in Equation 1 for the general case of N 3D points seen in M cameras, though it must be kept in mind that in this work we use only two cameras.

$$min(a_j,b_i)\sum_{i=1}^{N}\sum_{j=1}^{M}v_{ij}(d(Q(a_j,b_i),x_{ij}))^2 \qquad (1)$$

Here, x_{ij} is the position of the i_{th} correspondence on image j. The binary variable v_{ij} equals '1' if point i is visible in image j ('0' otherwise). The vectors a_j and b_i parameterize each camera j and 3D point i, respectively, with $Q(a_j,b_i)$ as the reprojection of point i on image j. Finally, d is the Euclidean distance in each image between each original correspondence and its associated reprojection. This minimization involves a total of $3N+11M$ parameters, and can be achieved using the Levenberg-Marquardt algorithm. An implementation that exploits the sparse block structure of the normal equations solved at each iteration to greatly speed up the process was used; details are presented in (Lourakis and Argyros, 2000). Bundle adjustment must be applied to the entire structure, in order to allow for detection of high-error correspondences, as outlined next.

2.2 Outlier Correspondence Detection and Correction

Once bundle adjustment has been applied on the structure and two cameras, all correspondences are now classified based on the reprojection error of the 3D point each pair generated; those classified as having low reprojection errors will be referred to as *inliers*, and high-error ones as *outliers*. Since bundle adjusment is the maximum-likelihood estimator for zero-mean Gaussian noise, the optimized pose and structure estimates (plus the known intrinsic parameters) allow for the 'unmasking' of errors purely in the correspondences in this step. If very erroneous initial pose and structure estimates arise from a very inaccurate input dense correspondence set, optimization may actually guide the estimates away from the global optimum in such cases, thus failing to unmask pure correspondence errors, but it is assumed that a reasonable amount of correspondences are accurate enough such that initial pose and structure estimates are in the vicinity of their optimal values.

The reprojection error for the i_{th} correspondence pair is taken as the sum of the absolute values of the errors obtained by reprojecting its resulting 3D point into each individual image. Then, a threshold on the reprojection error r_i (Equation 2) given optimized cameras $\hat{a}_j$ and structure $\hat{b}_i$ is established, such that correspondence pairs whose error is above the threshold are deemed outliers, while the rest are inliers. Without this threshold, or with a low one, the procedure described in the next section (whose processing time is linear in the amount of pixels) would be applied to nearly every pixel in the image, which is expensive. On the other hand, a higher threshold would imply faster processing, but with the downfall that some correspondences with relatively substantial

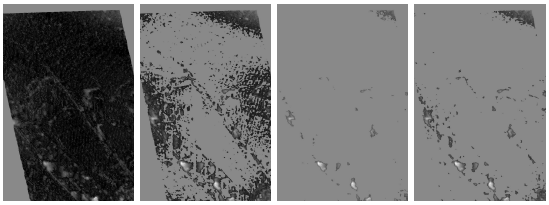

Figure 1: Applying a low threshold to detect outliers from a set of correspondences (left) results in unnecessary processing (middle left), while a high threshold erroneously yields very few outliers (middle right). An appropriate threshold must identify only the problematic regions (right).

errors are left uncorrected. This is shown in Figure 1, which shows the effect of the used threshold on the number of detected outliers. The algorithm should solely detect high-error correspondences in problematic regions. An analysis of the reprojection error histograms for different data sets reveals that the curves gradually taper off as the reprojection error grows. This observation is key towards determining an appropriate threshold. From visual observation of the detected outliers using different thresholds for different real and synthetic data sets, along with the corresponding histogram information, it was determined that a threshold t of an average (as defined in Equation 3) plus 1.5 standard deviations (Equation 4) of the reprojection errors results in an appropriate outlier detection.

$$r_i = \sum_{j=1}^{2} |d(Q(\hat{a}_j, \hat{b}_i), x_{ij})| \tag{2}$$

$$\mu_r = \frac{1}{N} \sum_{i=1}^{N} r_i \tag{3}$$

$$t = \mu_r + 1.5 \sqrt{\frac{1}{N} \sum_{i=1}^{N} (r_i - \mu_r)^2} \tag{4}$$

Next, for a given outlier correspondence pair, the objective is to correct the position of the match in the second (target) image to the information in the first (source) image, while keeping the position in the first image fixed, to find a better match than the one currently available. The algorithm works on a coarse-to-fine resolution pyramid, where a fixed amount of iterations (typically hundreds) is applied per resolution level, such that the pixel count doubles at each level. After constructing the hierarchy, a sub-pixel accurate iterative, three-phase algorithm is applied at successively finer levels. Each iteration consists of perturbation, matching (based on gradient descent) and affine-fitting phases. The resulting transformation for level i of the hierarchy is used as a starting prediction at level $i+1$.

Starting at the coarsest level, a fixed-size image chip from the source image is centered at the start position on the target image. The first phase of one iteration, perturbation, consists of adding noise to the source image chip in order to avoid local minima which could possibly occur in the next phase, which is based on gradient descent. In this matching phase, for each pixel of the source image chip, a local gradient is computed at its current position in the target image. This gradient is used to make a linear prediction of the direction and distance to move the source pixel in the target image to match its intensity (Duchaineau et al., 2007). Each pixel moves independently in this phase. For robustness, the movement step size is only a fraction of a pixel, and further modified according to the magnitude of the gradient. As the gradient magnitude becomes small (as determined by an adaptive threshold), the gradient direction becomes more noise than signal, and such pixels are eliminated from use in the next phase. In the final phase, a least-squares fit is applied to find an affine transformation to be applied to the source image chip. Only those pixels inside the chip that were not eliminated during the matching phase are used. The three-phase process is iterated a number of times at this coarsest level first and then at succesively higher resolutions, resulting in a new and more accurate correspondence position in the target image once completed. The process is illustrated in Figure 2. For an aerial view of a small section of a road with vehicles, the upper left image shows the initial position of the image chip, where gradients are color-coded such that the largest gradients are displayed in lighter colors. Results of the three-phase algorithm are also illustrated for a given iteration: the upper right image shows the result of noise perturbation followed by matching, where the image depicts (via tilts in the pixels) the direction and also the movement of each individual pixel, and the lower left image shows the affine fit computed from this information. The lower right image shows marked with an 'X' those pixels that were eliminated in the matching phase. Though this correction process is expensive, the goal is to achieve more accurate correspondences by taking into account the actual structure of the neighborhood around a given point, which is more strict than using just the pure epipolar constraint, which could be geometrically but not physically correct.

To determine the most appropriate fixed neighborhood size, the improvement percentage in the average reprojection error for detected outliers with respect to the average obtained before correction was tested for different sizes. It was concluded that similar results are obtained, which is quite remarkable and indicates that the correction process is very robust

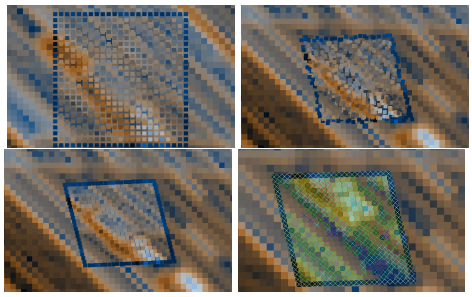

Figure 2: Affine correction process (see text for details).

even when using a relatively small neighborhood. For the *Aerial Views I* (Oxford Visual Geometry Group, 2009) data set, though slightly better results are obtained for a large 59*x*59 neighborhood (3.04% improvement), an 11*x*11 size (2.9% improvement) was chosen as it yields good results with only a fraction of the processing time. Results were actually worse (1.9% improvement) for a 35*x*35 neighborhood.

3 EXPERIMENTAL RESULTS

In this section the results of the presented approach are analyzed. Both real and synthetic data sets were used to test the algorithm. Tests on real imagery included an aerial imagery data set, which will be referred to as *Downtown*, and two publicly-available data sets: *Aerial Views I* (Oxford Visual Geometry Group, 2009) and *Rocks 2* (Hirschmüller and Scharstein, 2007). Figure 3 shows the resulting 3D structure obtained after correction using an image pair from the *Downtown* data set. Figure 4 shows reprojection errors after bundle adjustment (color-coded such that white means a high error and black a low one, over a uniform gray background) for the same image pair. It is clear that the higher errors in general are seen on structures that tend to have plain or repetitive patterns, for example on highways and train tracks (circled in red), near occlusion edges (green) and near the edges of the image (blue). After applying the proposed method, it can be seen that reprojection errors in these areas are generally lower, and the reconstruction is very accurate as seen in Figure 3 for such areas. The highest remaining errors are seen near occlusion edges, which makes sense since there is information missing in such areas (as opposed to texture-less regions, which can potentially be matched with enough neighborhood information). A synthetic scene, which will now be referred to as *Coneland* was also used to test the proposed algorithm. Figure 5 shows results of the outlier detection and correction from two images of this data

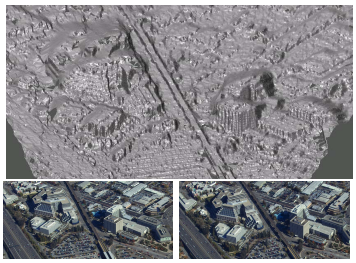

Figure 3: Closeups of a reconstruction (top) and its two input images from the *Downtown* data set (bottom), after outlier correction.

set. A ground-truth evaluation of the proposed algorithm was also performed. Table 1 shows the pose errors with respect to the ground-truth values when using the original set of dense correspondences versus the modified set after applying the proposed algorithm for the *Coneland* data set. Translational error is obtained as the angle in degrees of the dot product between the ground-truth translation and each estimate. Rotational error is obtained as the angle for the quaternion corresponding to the difference rotation matrix between ground-truth and estimated rotations for each case. It can be seen that pose estimates improve, even though the robust RANSAC is used to estimate F, showing that an overall more accurate set of correspondences is indeed achieved. Table 2 shows the outlier percentage and outlier reprojection error improvement percentage when applying the algorithm for some of the test data sets. At first glance the improvements may seem small, but when dealing with sub-pixel accuracy even small errors can result in large structural inaccuracies, so in practice the improvement is substantial.

One possible improvement for the algorithm is to use adaptive neighborhood sizes for the outlier correction process, based on intensity variation statistics for a given chip position. Using larger chips could potentially yield more accurate results in texture-less regions. The use of hardware solutions (such as using GPU's) to speed up expensive processes must also be further analyzed.

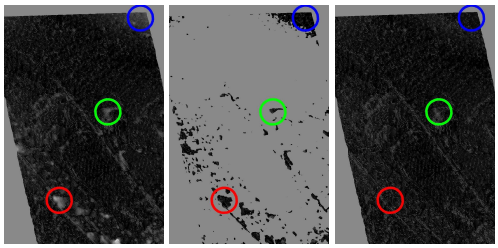

Figure 4: Reprojection errors after bundle adjustment (left) for an image of the *Downtown* data set. The detected and corrected outliers are shown (middle), along with errors for the resulting set of correspondences (right).

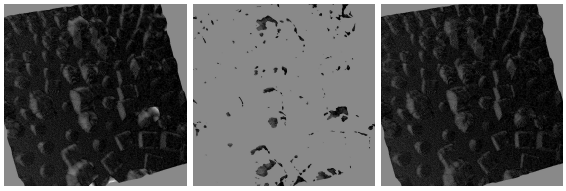

Figure 5: Reprojection errors after bundle adjustment (left) for an image of the *Coneland* data set. The detected and corrected outliers (middle) are shown, along with the errors for the resulting set of correspondences (right).

Table 1: Pose errors $\triangle_R$ for rotation and $\triangle_T$ for translation (in degrees) with respect to ground-truth values using original and modified dense correspondences, for the *Coneland* data set.

Correspondences	$\triangle_R$	$\triangle_T$
Original	9.818953°	2.443838°
Modified	0.167859°	0.418460°

Table 2: Outlier percentage, average outlier reprojection error μ_E (in pixels) before correction and error improvement percentage $\triangle_E$ for tested data sets.

Data set	Outliers	μ_E	$\triangle_E$
Downtown	5.442%	2.072	10.956%
Aerial Views I	8.475%	11.392	3.234%
Coneland	5.753%	2.759	3.404%

4 CONCLUSIONS

This paper presented a new method for detecting and correcting outlier dense correspondences between two images. Initial estimates for the pose and scene structure are obtained from the given dense correspondences, assuming known camera intrinsic parameters, and are then bundle-adjusted. The resulting reprojection errors per correspondence pair are used as a metric to separate high-error and low-error correspondences. Then, an affine neighborhood-based iterative algorithm operating on a coarse-to-fine resolution pyramid is used to correct outlier correspondences. Results on both real and synthetic scenes show that a more accurate set of dense correspondences is obtained after applying the proposed method, which results in an improvement in pose and structure estimates.

REFERENCES

Duchaineau, M., Cohen, J., and Vaidya, S. (2007). Toward fast computation of dense image correspondence on the GPU. In *Proceedings of HPEC 2007, High Performance Embedded Computing, Eleventh Annual Workshop*, pages 91–92, Lincoln Laboratory, Massachusetts Institute of Technology.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition.

Hirschmüller, H. and Scharstein, D. (2007). Evaluation of cost functions for stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 91–92, Minneapolis, MN.

Lourakis, M. and Argyros, A. (2000). The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg-Marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece.

Mayoral, R. and Aurnhammer, M. (2004). Evaluation of correspondence errors for stereo. In *17th International Conference on Pattern Recognition (ICPR'04)*, volume 4, pages 104–107.

Oxford Visual Geometry Group (2009). Multi-view and Oxford Colleges building reconstruction. http://www.robots.ox.ac.uk/ vgg/data/data-mview.html.

Rodehorst, V., Heinrichs, M., and Hellwich, O. (2008). Evaluation of relative pose estimation methods for multi-camera setups. In *International Archives of Photogrammetry and Remote Sensing (ISPRS '08)*, pages 135–140, Beijing, China.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal On Computer Vision*, 47(1-3):7–42.

Stewénius, H., Engels, C., and Nistér, D. (2006). Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294.

Xiong, Y. and Matthies, L. (1997). Error analysis of a real-time stereo system. In *IEEE Conference on Computer Vision and Patter Recognition (CVPR)*, pages 1087–1093.