Solving the Correspondence Problem by Finding Unique Features

Peter Biber, Wolfgang Strasser
University of Tuebingen, WSI/GRIS
email: biber@gris.uni-tuebingen.de

Abstract

Undoubtedly the correspondence problem is one of the most important problems in computer vision. This paper describes a simple modification of one standard algorithm to identify pairs of feature points in two images which belong to the same scene point. Without directly using any standard constraints, typically 90 to 100 percent of the obtained matches are correct. Even more impressive, almost no correspondences are found in images not showing the same scene. Key to the success of the new method is the introduction of the uniqueness of a feature. The uniqueness of a feature is a global attribute of a feature and together with the corresponding local description creates a powerful discrimination tool. It is proposed to replace the standard uniqueness and symmetry constraint by this measure. We also give a general derivation of our approach, where the proposed algorithm is divided into an algorithm and a "meta algorithm". This meta algorithm simulates the first algorithm and gains thereby insights of its applicability and its optimal parameters.
Keywords: Image registration, feature matching

1 Introduction

Undoubtedly the correspondence problem is one of the most important problems in computer vision. For many applications it is important to identify corresponding points in a pair or a set of images. Uncalibrated stereo or 2D image mosaicing (e.g [15], [16]) are not the only examples, others include also object recognition ([8],[3]).

One problem of correspondence algorithms is typical of many computer vision algorithms: Parameters and thresholds have to be determined heuristically. Further, the optimal parameters are dependent on the input.

The algorithm presented here will determine its parameters to a certain degree automatically dependent on the input. We will also give a recipe, how automatic threshold and parameter selection could be included in other algorithms.

Finding corresponding features at first requires extraction of features. One well known approach is to extract small windows (templates, patches) of a given size around points found by an interest operator. Such an interest point operator ensures, that the selected points are locally distinct. The hope is then: These features are distinct enough, such that they can be discriminated from other ones also globally.

The problem is now to find the correspondences between the features of two images. First, a pair of features is said to form a candidate pair, if a similarity measure between them exceeds a certain threshold. As experience shows, a lot of these candidate matches are wrong, if the threshold is too large. But if the threshold is too small, a lot of correct matches don't pass the test.

A lot of research has been spent in developing robust methods to determine the right matches from a candidate set. These methods can be subdivided into imposing constraints, local support and robust estimation of the scene geometry.

The most important constraints are uniqueness (one feature can have only one match) and symmetry (if X matches Y, then Y has to match X). These constraints are of course physically well founded. They are not explicitly contained in our approach, although in a certain sense they are the base of it. In fact, we propose to replace these constraints by the method presented in this paper.

The second method (local support) assumes, that other compatible matches have to exist in the neighborhood of a correct match. A relaxation procedure can be used to discard matches, which are isolated in this respect.

The third method, robust estimation of the scene geometry (e.g. the epipolar geometry) is used to check the consistency of the matches and then to establish many more matches. It is not quite clear,

whether this step still belongs to the feature matching pipeline, since often the estimation of the scene geometry is the goal per se. Nevertheless, methods like RANSAC can estimate the geometry despite a large fraction of outliers. But undoubtedly, they perform better and faster with a low percentage of outliers.

The literature on this topic is immense, all three methods are covered in [15]. A recent empirical evaluation of the mentioned constraints and other ones is given in [14]. Other research contains methods working on the candidate pairs or methods for extracting features being invariant to e.g. rotation or affine transformations (e.g. [9],[5],[10],[3],[13]).

The method proposed in this paper describes a step between feature extraction and establishing candidate pairs. It consists of assigning each feature a uniqueness with respect to the whole set of features of one image. This measure can then be used in two ways. First, an individual threshold can be determined for each feature. Second, a representation of that feature can be chosen, such that the uniqueness is high.

The uniqueness of a feature will be introduced in section 2. A feature matching algorithm using the uniqueness is proposed in section 3. Section 4 defends the algorithm against only being a heuristic by giving an interpretation of the algorithm as one algorithm simulating another one and yielding a context-dependent algorithm as result. Finally we present some results and compare them to the state-of-the-art technique.

2 The uniqueness of a feature

We will now introduce the uniqueness of a feature independent of a special representation or a special measure. It is assumed, that a distance d between two features is given, whereas a small distance denotes a high similarity. Given a set of features $S = \{f_i\}$ the uniqueness of a feature $f \in S$ with respect to S is defined as:

$$\mathtt{unique}(f, S) = \mathtt{min}\{\mathtt{d}(\mathtt{f}, \mathtt{f_i}) | \mathtt{f_i} \in \mathtt{S}, \mathtt{f} \neq \mathtt{f_i}\}. \quad (1)$$

The uniqueness is therefore the minimum distance to another feature of the set. A feature with a high uniqueness is intuitively easier matched than a feature with a low uniqueness, because it is not only salient in a small neighborhood, but also in the whole set.

A simple strategy to establish matches is now to consider only feature pairs, where the distance is smaller than the uniqueness. This way, a lot of features can be discarded even before trying to match them by requiring them to have a high uniqueness value. Further, if there are several representations available, the representation with the best uniqueness can be chosen as appropriate.

3 The algorithm

In the matching algorithm presented here, the features are small image patches of size $(2n+1)\times(2n+1)$ around interest points. In the following all sums are taken over such a patch, whereas each pixel's intensity is denoted by I. Interest points are identified in the following way([6],[11]): For each possible image patch, the values of the spatial derivatives of the patch are collected in the matrix

$$\begin{pmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{pmatrix}$$

If the smaller eigenvalue of this matrix is large enough, the patch is selected. We simply take the 500 patches having the largest values.

Two patches can be compared using the zero-mean normalized cross-correlation (NCC): First, the intensity values of the patches are replaced by $I - \bar{I}$ to have an average of zero, then the vector of the values is normalized to have a length of one and then

$$NCC = \sum I_1 I_2. \quad (2)$$

This measure is extended canonically to work on RGB-values.

Since the NCC returns a value between 1 and -1 with higher values meaning lower distance, we use $(1 - NCC)$ as distance:

$$\mathtt{unique}_{NCC}(f, S) = \mathtt{min}\{1 - \mathtt{NCC}(f, f_i) | f_i \in S, f \neq f_i\}. \quad (3)$$

A pair of features $f_1 \in S_1, f_2 \in S_2$ of two different sets S_1, S_2 now match if their distance is both smaller than an absolute threshold and a certain amount smaller than their uniqueness.

We use minimum NCC of 0.7 here, which is a relative low value (Values between 0.8 and 0.9 are standard values for matching). The only requirement is here, that as many as possible correct matches pass this test. The second requirement can be expressed by the following two conditions:

$$1 - \mathtt{NCC}(f_1, f_2) < \mathtt{unique}_{\mathtt{NCC}}(f_1, S_1) - \tau \quad (4)$$

$$1 - \mathtt{NCC}(f_1, f_2) < \mathtt{unique}_{\mathtt{NCC}}(f_2, S_2) - \tau \quad (5)$$

A good choice for τ is around 0.2, as will be shown in the experimental results. The algorithm allows therefore some features to match with a very low threshold (0.7), while some features (e.g. a feature with a uniqueness of 0.1) cannot match at all.

We call the difference between the uniqueness and the distance the *confidence* of a match:

$$\texttt{conf}(f_1, f_2) = \quad (6)$$
$$\texttt{min}(\texttt{unique}_{\texttt{NCC}}(f_1, S_1), \texttt{unique}_{\texttt{NCC}}(f_2, S_2)) - (1 - \texttt{NCC}(f_1, f_2)).$$

Equations 4 and 5 can then be expressed by

$$\texttt{conf}(f_1, f_2) > \tau \quad (7)$$

Our results will show, that this confidence score has much more influence than the NCC-score.

This algorithm is now executed for several choices of the window size n and even for several resolutions of the image by down-scaling the images. By calculating the uniqueness, the best features of every combination will be determined automatically.

Our approach can be seen as a natural extension to the interest point operator. This operator guarantees the uniqueness of a feature in a small neighborhood, our notation describes the uniqueness with respect to the whole image.

Some experimental results will be given in section 5. But first, we give an explanation of the algorithm as one algorithm simulating another one.

4 The meta algorithm - Context-dependent reasoning about algorithms

We would like to see the algorithm in the sense of a meta algorithm simulating another algorithm. To show the idea, we first divide the algorithm in two parts. One algorithm (A), that is known to be able to do some task in some context. Second, an algorithm (B), that simulates the first one and determines from the result of the simulation, whether:

1. The algorithm is appropriate in the global context.
2. If yes, which parameters / thresholds / representations are appropriate.

The algorithm A is in this case simply the following:

if $\texttt{NCC}(f_1, f_2) > a$ then (f_1, f_2) is a match.

where a is a threshold. Possible parameters/representations are the window size and the resolution of the image. To allow the simulating algorithm B to simulate A, B needs somehow a model, on which to run A. The acquisition of such a model may be difficult in general, but is easy in this case. The image itself is a good model of what is to be expected. And one thing is sure: Each pixel in the image is the projection of a different world point. This insight can be seen as a variant of the uniqueness constraint applied to only one image. So if B detects, that A finds a match for a pair of features for a certain threshold a, this match is for sure wrong and therefore the combination of threshold/window size/resolution is not appropriate in this context. On the other hand, if B detects, that A finds no match even for a very low threshold a, it can state, that this feature is likely to be matched correctly, even with a low threshold . For some images, no features at all will be classified as potentially good, so the result is: The algorithm is not appropriate at all for this image.

This simulation naturally depends on the quality of the internal model. While the simulation will be better, the better the model is, a model will never represent all aspects of reality. One solution of that problem is to represent the result of the algorithm by a continuous measure like the confidence. But this is not possible for discrete decisions, that are part of the algorithm. To overcome this problem, the validity and the usefulness of the result can be improved by simulating the algorithm with softer overall conditions. In this way the simulating algorithm has more possibilities to produce errors. Errors, which could in reality also be caused by imperfect models. We apply this idea the following way: The extraction of interest points is a step, where such discrete decisions are taken. If one point is selected in one image (because his interest point score is just above the threshold), and the corresponding point in a second image is not selected (because his point of interest score is a little bit under the threshold), errors are likely to occur. To model this error source to some extent, we use now two thresholds: A lower one for the calculation of the uniqueness, and a larger one for establishing matches.

5 Experimental results

The algorithm has been tested on a collection of several hundred images. The result of the algorithm is used in a larger framework to stitch a set of images fully automatically (as far as possible) to a panorama. For this reason most of the test image sets consist of images taken from a common viewpoint.

In our application, no prior information about the connectivity of the images is given. A key goal for the algorithm is therefore to deliver a lot of matches when applied to images with significant overlap, and

fewer or no matches when applied to images that do not overlap.

Figure 1: Images 4,5 and 11 of the sequence *City*

Our test set presented here (23 images, Figure 1) shows a city scene containing a large building with a lot of similar structures like windows. The images have here a size of 320×256.

The algorithm was run with image patch sizes of 7×7, 9×9 and 11×11 at five different resolutions. Each resolution was calculated by down-sampling the previous resolution by a factor of 1.5. The maximal number of features for each combination of patch size and resolution was set to 500. All features were used to calculate the uniqueness, but only the 80 percent with the highest interest point score were used to establish matches (as explained in the last section).

To count the correct matches, the following method was used. We registered the images using the results of the algorithm, intensity-based optimization ([12]) and manual interaction (if necessary) using a pinhole camera model. The test sequence mapped onto a cylinder is shown in figure 2. This registration was used to subdivide the matches into correct, inexact and wrong.

Since the photos were taken handheld and sometimes small camera movements occurred, some matches, which are visually correct, are some pixels away from the position calculated by the estimated mapping; these are the inexact matches. We count correct matches therefore this way: All matches with distances smaller than 2 pixels in their corresponding resolution are counted as correct. Otherwise, if the distance is smaller than 5 pixels in total, the match is not counted at all (neither as correct nor as wrong). All other matches are counted as wrong.

Results are shown in three formats answering the three important question: How many of the matches are good in percent, how many matches are good in total and how many matches are wrong. For the following plots, the minimum NCC-score (NCC) was varied from 0.7 to 0.95 and τ was varied from -0.3 to 0.4. A value of τ of -0.3 is nearly equivalent to not using the uniqueness of the features at all.

Figure 3 gives an impression of the results of the algorithm. Here $\tau = 0.2$ and nearly all matches are correct. Figures 4 show the results of applying the algorithm to the complete image sets. That is, each image is compared with each other, in total 276 comparisons. The majority of the comparisons is therefore between images not showing the same scene. Here the impact of the algorithm is striking: With a minimum NCC of 0.9, and $\tau = -0.3$, there are only 35 percent of correct matches, whereas with a minimum NCC of 0.7 and $\tau = 0.17$ 90 percent are correct. Figure 5 shows the effect of τ for two different settings for the minimum NCC (0.7 and 0.9) in this test.

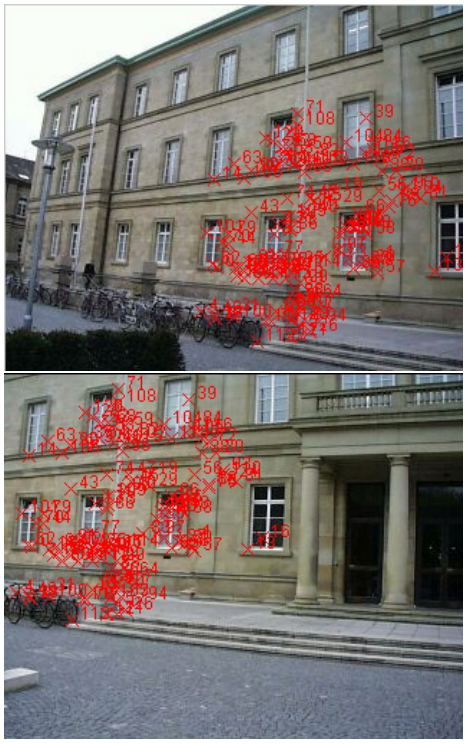

Figure 3: Found matches for $\tau = 0.2$.

Figure 6 shows the result of running the algorithm on image pair 4/11 of the sequence *city*. These images have no overlap and so all found matches are wrong. For a minimum NCC of 0.9, there are still 176 matches. In contrast, the highest occurring confidence score is 0.17, and for $\tau = 0.1$, there are only

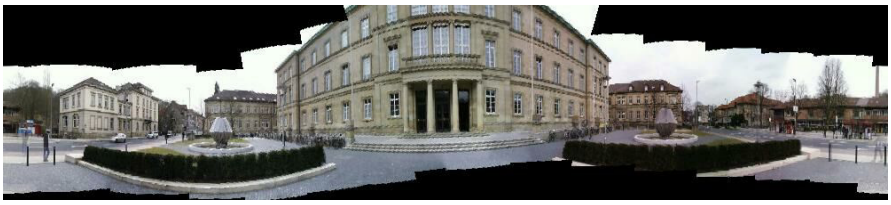

Figure 2: Stitched sequence *City* mapped onto a cylinder

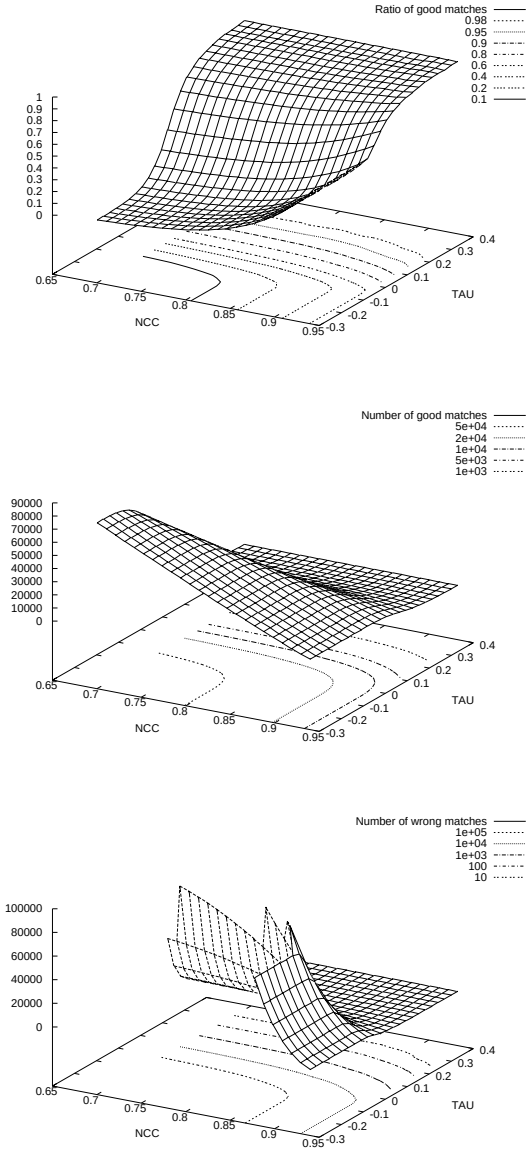

Figure 4: Result of comparing each image of the sequence *city* with each other.

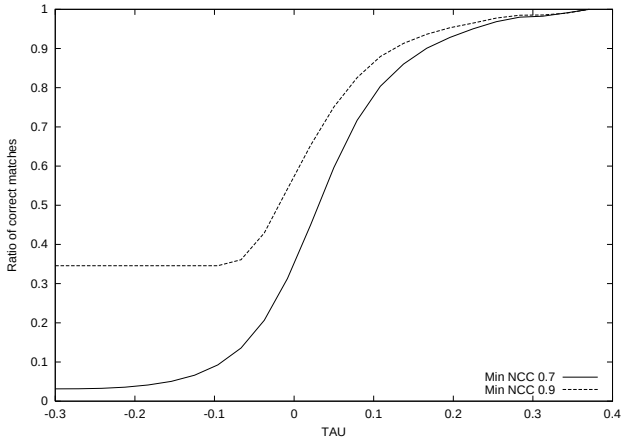

Figure 5: Dependence of correct matches ratio from τ.

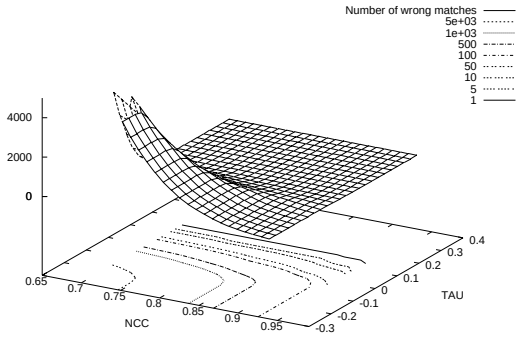

Figure 6: Result of comparing images 4 and 11 of the sequence *city*, which have no overlap, so all matches are wrong.

eight matches.

	Correct	Wrong	Ratio
U+C	52591	128657	0.29
U+C+D	43462	20874	0.68
Min Conf 0.1	20640	3472	0.85
Min Conf 0.2	7095	393	0.95
Min Conf 0.3	1681	41	0.98
Min Conf 0.1 + U+C+D	19781	1508	0.93
Min Conf 0.2 +U+C+D	7095	393	0.95
Min Conf 0.3 +U+C+D	1681	41	0.98

Figure 7: Comparison of the proposed algorithm to other aproaches. U and C stand for applying Uniqueness and Symmetry constraint. D stands for applying a Disparity gradient constraint. Opposed are the results of requiring a minimum confidence. For this comparison, we used a minimum NCC of 0.8.

The outcomes of our experiments show, that the confidence score is dominant for the probability of a correct match. Certainly, from figure 4 or 5 it can also be seen, that the correlation score has an influence. For sorting the matches, both values should be taken into account. A possibility is here to take the probability values of the results our test sets (e.g. simply the probability distribution of 4) as a measure. Such a distribution is different for different test sets, but the differences are small and the structures are similar.

We have shown up to now, that the results are good, but how do we perform compared to traditional methods? For a comparison, we have selected the classical symmetry and uniqueness constraint (as mentioned in the introduction). Additionally, we have implemented a disparity gradient constraint with parameters as proposed in [14]. The idea here is, that proximate points in the image should have similar disparities. The disparity gradient of two matches (f_1, f_2) and (f_1', f_2') is defined as

$$\Delta = \frac{|d(f_1, f_2) - d(f_1', f_2')|}{d(m_1, m_2)}, \qquad (8)$$

where d is the euclidian distance and m_1 (m_2) is the midpoint of f_1 and f_1' (f_2 and f_2'). This measure is now used in a constraint, that accepts a match if it shares the disparity gradient with at least 3 of its 5 closest neighbors using a thresholds of 0.4.

We ran the algorithm again on the whole image set using a minimum NCC of 0.8 and the mentioned constraints. The results are shown in figure 7. Application of all three constraints yields a result of 68 percent correct matches. In contrast, requiring a minimum confidence of 0.1 yields a result of 85 percent correct matches, but at the cost of having only half the number of correct matches in total. Application of the uniqueness and symmetry constraint together with requiring a minimum confidence does not improve the result, because these constraints are already contained implicitly. However, the disparity gradient helps for matches with confidences smaller than 0.1, showing that it is a concept being relativly orthogonal to our approach.

Though we have tested our approach only on a small set of stereo pairs, the algorithm should be equally useful in this case. Figure 8 shows the results on an example stereo pair.

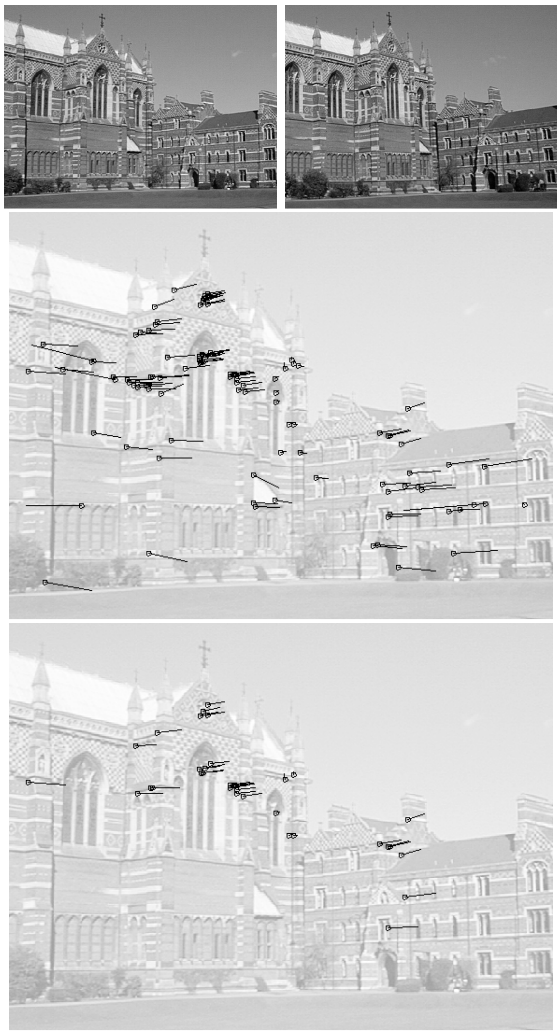

Figure 8: Results on a stereopair (from [7],chapter 10). Shown are the correspondences for $\tau = .2$ (middle) and $\tau = .3$ (bottom).

6 Conclusions

The results shown here are representative for all our test sets and show the robustness of the approach in

difficult environments with a lot of similar structures, where other algorithms could fail. The price to pay is, that sometimes few matches or no matches at all are found. For natural scenes, a really surprising number of features with high uniqueness are found (much more than in the test set presented here).

We want to emphasize two general advantages of our approach: First, our notion of uniqueness and confidence describes a continuous measure, opposed to a constraint, which is a discrete measure. This enables us to sort matches according to their confidence, which is useful for RANSAC. Secondly, the uniqueness can be calculated on a single image, opposed to the uniqueness and the symmetry constraint, which is calculated on a set of candidate matches. This is useful, if an image has to be compared with several other images, since the computations have to be done only once.

The meta-algorithm given in section 4 gives an explanation for the good results on image pairs showing the same scene. For the apparent ability of the algorithm to discriminate between different scenes, this is of course no explanation. The key to this success is maybe the combination of a local description (the RGB values of the image patch) with a global property (the uniqueness). Together with the uniqueness, the image patch codes not only a local appearance, but also the absence of a lot of similar patches. The more unique a feature is, the larger is the subspace of all possible image patches, which can occur only once. An interesting fact is, that the representation of the image patch is very high dimensional, while the representation of the global property is only one dimensional.

We implemented a tool to automatically stitch a panorama from a set of images without any prior information. The algorithm presented here is the central part of this tool. For example the test set shown in the experimental results was stitched automatically except for closing the panorama, which had to be done manually.

References

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

[2] L. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[3] C. and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[4] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *IJCV*, 10(2):101–124, 1993.

[5] Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *CVPR*.

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, 1988.

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

[8] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the ICCV*, pages 1150–1157, 1999.

[9] P. Montesinos, V. Gouet, and R. Deriche. Differential invariants for color images. In *Proceedings of 14 th International Conference on Pattern Recognition*, 1998.

[10] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *ICCV*, pages 754–760, 1998.

[11] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[12] H. Shum and R. Szeliski. Panoramic image mosaics. Technical Report MSR-TR-97-23, Microsoft Research, 1997.

[13] T. Tuytelaars, L. Van Gool, L. D'haene, and R. Koch. Matching affinely invariant regions for visual servoing. In *IEEE Conference on Robotics and Automation*, pages 1601–1606, may 1999.

[14] E. Vincent and R. Laganire. An empirical study of some feature matching strategies. In *Proc. of the International Conference on Vision Interface*, pages 139–145, 2002.

[15] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.

[16] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2d mosaic from a set of images. CVPR 97.