Multi-Image Matching using Multi-Scale Oriented Patches[1]

Matthew Brown[2], Richard Szeliski, and Simon Winder

December 2004

Technical Report
MSR-TR-2004-133

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

[1] A shorter version of this report will appear in CVPR'2005

[2] The University of British Columbia

Abstract

This paper describes a novel multi-view matching framework based on a new type of invariant feature. Our features are located at Harris corners in discrete scale-space and oriented using a blurred local gradient. This defines a rotationally invariant frame in which we sample a feature descriptor, which consists of an 8×8 patch of bias/gain normalised intensity values. The density of features in the image is controlled using a novel adaptive non-maximal suppression algorithm, which gives a better spatial distribution of features than previous approaches. Matching is achieved using a fast nearest neighbour algorithm that indexes features based on their low frequency Haar wavelet coefficients. We also introduce a novel outlier rejection procedure that verifies a pairwise feature match based on a background distribution of incorrect feature matches. Feature matches are refined using RANSAC and used in an automatic 2D panorama stitcher that has been extensively tested on hundreds of sample inputs.

1 Introduction

Early work in image matching fell into two camps – feature-based methods and direct methods. Feature-based methods attempt to extract salient features such as edges and corners, and then to use a small amount of local information e.g. correlation of a small image patch, to establish matches [F8̈6, Har92]. In contrast to feature-based methods, which use only a small amount of the available image data, direct methods attempt to use all of the pixel values in order to iteratively align images [LK81, Ana89]. Other approaches to matching and recognition have used invariants to characterise objects, sometimes establishing canonical frames for this purpose [RZFM92].

At the intersection of these approaches are invariant features, which use large amounts of local image data around salient features to form invariant descriptors for indexing and matching. The first work in this area was by Schmid and Mohr [SM97] who used a jet of Gaussian derivatives to form a rotationally invariant descriptor around a Harris corner. Lowe extended this approach to incorporate scale invariance [Low99, Low04]. Other researchers have developed features that are invariant under affine transformations [Bau00, TG00, BL02]. Interest point detectors vary from standard feature detectors such as Harris corners or DOG maxima to more elaborate methods such as maximally stable regions [MCUP02] and stable local phase structures [CJ03].

Generally, interest point extraction and descriptor matching are considered as two basic steps, and there has been some progress in evaluating the various techniques with respect to interest point repeatability [SMB00] and descriptor performance [MS03]. Other researchers have suggested that interest points should be located such that the solutions for matching position [ST94], orientation and scale [Tri04] are stable. Compelling applications of invariant features based matching have been demostrated for object recognition [Low99], structure from motion [SZ02] and panoramic imaging [BL03].

While a tremendous amount of progress has been made recently in invariant feature matching, the final word has by no means been written. In this paper, we advance the state of the art in several directions. First, we develop a novel adaptive non-maximal suppression algorithm that better distributes features across the image than previous techniques (section 2.2). Second, we show that with suitable modifications, a direct patch-based sampling of the local image structure can serve as a useful invariant feature descriptor (section 2.7). Third, we develop a feature space outlier rejection strategy that uses all of the images in an n-image matching problem to give a background distribution for incorrect matches (section 3).

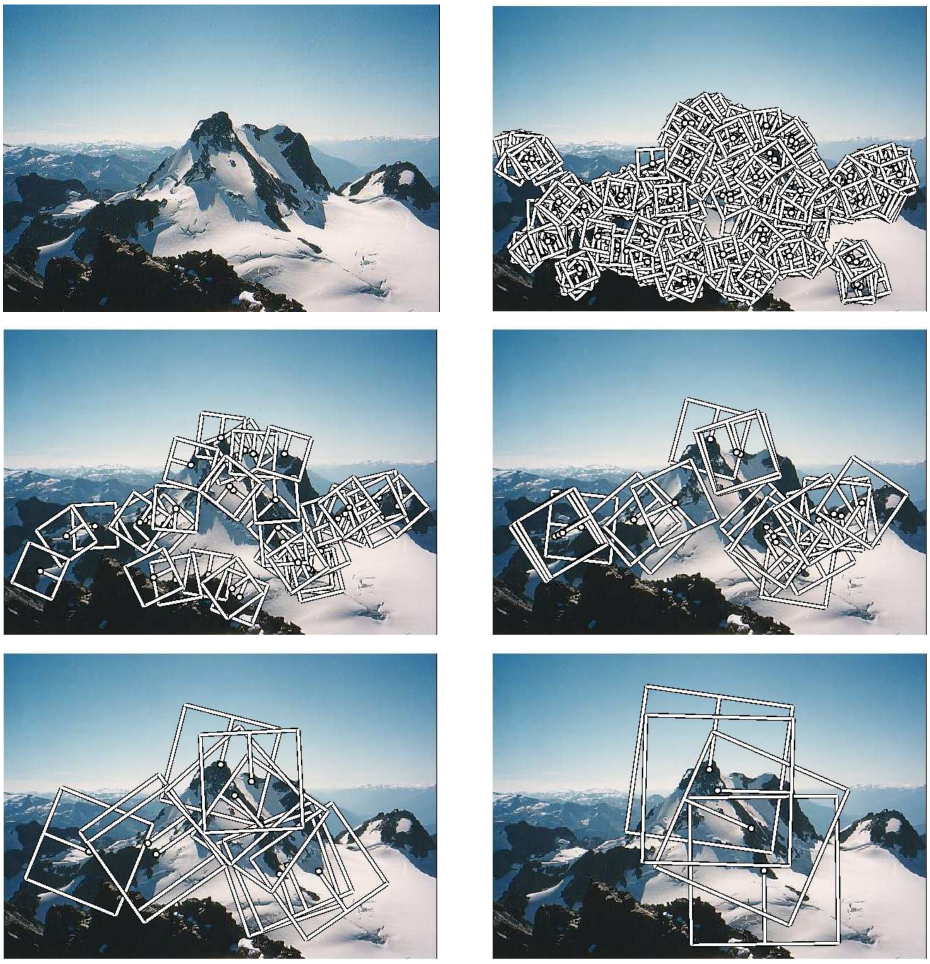


Figure 1: *Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels. The boxes show the feature orientation and the region from which the descriptor vectors are sampled.*

Fourth, we develop an indexing scheme based on low-frequency Haar wavelet coefficients that greatly speeds up the search for feature correspondences with minimal impact on matching performance (section 3.4). We close with a discussion of our results and ideas for future work in this area.

2 Multi-scale Oriented Patches

In general, the transformation between corresponding regions in a pair of images is a complex function of the geometric and photometric properties of the scene and the cameras. For the purposes of this work, we reduce this to a simple 6 parameter model for the transformation between corresponding image patches

$$I'(\mathbf{x}') = \alpha I(\mathbf{x}) + \beta + n(\mathbf{x}) \tag{1}$$

where

$$\begin{aligned} \mathbf{x}' &= \mathbf{A}\mathbf{x} + \mathbf{t} \qquad (2) \\ \mathbf{A} &= s \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \qquad (3) \end{aligned}$$

The four geometric parameters are t_1, t_2, θ, s—position, orientation and scale. The remaining two photometric parameters are α, β—gain and bias. The error $n(\mathbf{x})$ represents imaging noise and modelling error. Features are located at points where this transformation is well defined i.e. the autocorrelation of $I(\mathbf{x})$ is peaked [F̈86, Har92, ST94, Tri04]. To compare features, one could in principle compute the maximum likelihood estimates for the transformation parameters between a pair of image locations. Assuming Gaussian noise, this can be done iteratively by solving a non-linear least squares problem. However, for efficiency, we instead associate with each feature a set of model parameters, which implies a set of transformation parameters between each pair of features. We then use the statistics of the matching error $n(\mathbf{x})$ to verify whether a match is correct or incorrect.

2.1 Interest Points

The interest points we use are multi-scale Harris corners [F̈86, Har92, Tri04]. For efficiency, we work with greyscale images $I(x, y)$. For each input image $I(x, y)$, we form an image pyramid with the lowest level $P_0(x, y) = I(x, y)$ and higher levels related by smoothing and subsampling operations

$$\begin{aligned} P'_l(x, y) &= P_l(x, y) * g_{\sigma_p}(x, y) \qquad (4) \\ P_{l+1}(x, y) &= P'_l(sx, sy) \qquad (5) \end{aligned}$$

l denotes the pyramid level, and $g_\sigma(x, y)$ denotes a Gaussian kernel of standard deviation σ. We use a subsampling rate $r = 2$ and pyramid smoothing $\sigma_p = 1.0$. Interest points are extracted from each level of the pyramid. Other authors use sub-octave pyramids [SMB00, Low04]. Since we are mostly concerned with matching images that have the same scale, this is left for future work.

The Harris matrix at level l and position (x, y) is the smoothed outer product of the gradients

$$\mathbf{H}_l(x, y) = \nabla_{\sigma_d} P_l(x, y) \nabla_{\sigma_d} P_l(x, y)^T * g_{\sigma_i}(x, y) \tag{6}$$

∇_σ represents the spatial derivative at scale σ i.e.

$$\nabla_\sigma f(x, y) \triangleq \nabla f(x, y) * g_\sigma(x, y) \tag{7}$$

We set the integration scale $\sigma_i = 1.5$ and the derivative scale $\sigma_d = 1.0$ and use the corner detection function

$$f_{HM}(x, y) = \frac{\text{det } \mathbf{H}_l(x, y)}{\text{tr } \mathbf{H}_l(x, y)} = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \tag{8}$$

which is the harmonic mean of the eigenvalues (λ_1, λ_2) of $\mathbf{H}$. Interest points are located where the corner strength $f_{HM}(x, y)$ is a local maximum of a 3×3 neighbourhood, and above a threshold $t = 10.0$.

The reason for this choice of interest point detection function can be understood in terms of the relationship between $\mathbf{H}$ and the local autocorrelation function. For an image $I(\mathbf{x})$, the first order Taylor expansion gives an expression for the local autocorrelation

$$e(\mathbf{x}) = |I(\mathbf{x}) - I_0|^2 = \mathbf{x}^T \frac{\partial I}{\partial \mathbf{x}} \frac{\partial I}{\partial \mathbf{x}}^T \mathbf{x} = \mathbf{x}^T \mathbf{H} \mathbf{x} \tag{9}$$

Interest points are located at peaks in the autocorrelation function. This means that $e(\mathbf{u})$ is large for all unit vectors $\mathbf{u}$, which is equivalent to requiring that both eigenvalues of $\mathbf{H}$ are large[1]. Figure 2 compares isocontours of our interest point detection function (Harmonic mean) with the common Harris and Shi-Tomasi detectors. Note that all the detectors require both eigenvalues to be large.

$$\begin{array}{rcccccc} \text{Harris} & f_H & = & \lambda_1\lambda_2 - 0.04(\lambda_1+\lambda_2)^2 & = & \text{det } \mathbf{H} - 0.04(\text{tr } \mathbf{H})^2 \\ \text{Harmonic mean} & f_{HM} & = & \lambda_1\lambda_2/(\lambda_1+\lambda_2) & = & \text{det } \mathbf{H}/\text{tr } \mathbf{H} \\ \text{Shi-Tomasi} & f_{ST} & = & \min(\lambda_1, \lambda_2) & & \end{array}$$

[1] Note that in practice $\mathbf{H}$ is integrated over a range as in equation 6 (otherwise it would be rank 1).

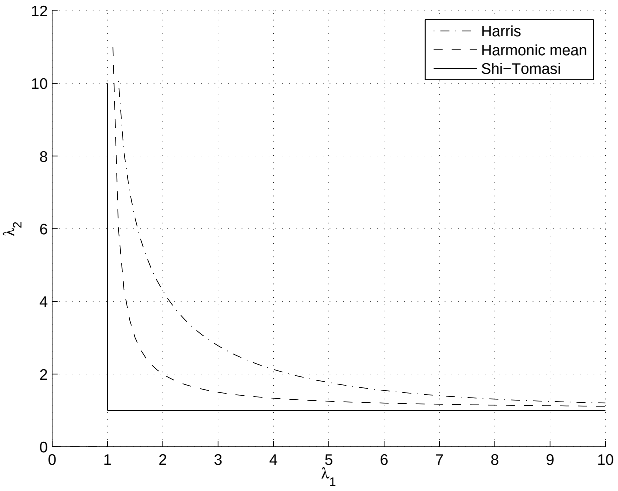


Figure 2: *Isocontours of popular interest point detection functions. Each detector looks for points where the eigenvalues* λ_1, λ_2 *of* $\mathbf{H} = \int_{\mathcal{R}} \nabla I \nabla I^T d\mathbf{x}$ *are both large.*

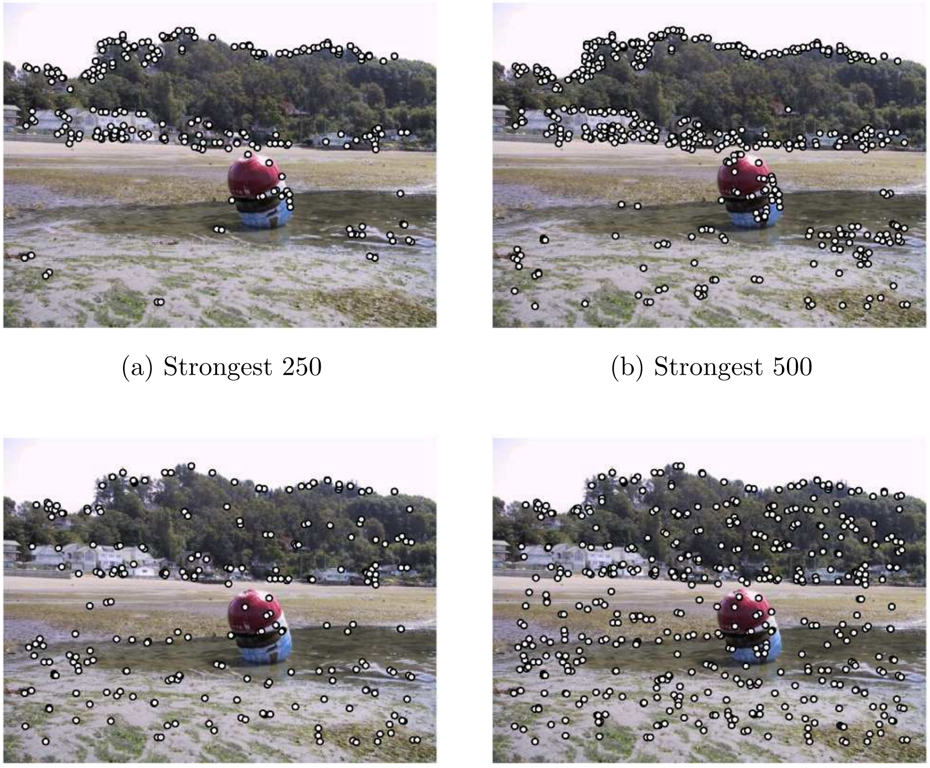
Preliminary experiments suggest each of these detectors give roughly the same performance, although one could compute repeatability statistics to confirm this.

2.2 Adaptive Non-maximal Suppression

Since the computational cost of matching is superlinear in the number of interest points, it is desirable to restrict the maximum number of interest points that are extracted from each image. At the same time it is important that the interest points that are generated are spatially well distributed over the image, since the area of overlap between a pair of images may be small. To satisfy these requirements, we developed an adaptive non-maximal suppression (ANMS) strategy to select a fixed number of interest points from each image.

Interest points are suppressed based on the corner strength f_{HM} and only those that are a maximum in a neighbourhood of radius r pixels are retained. Conceptually, we initialise the suppression radius $r = 0$ and then increase it until the desired number of interest points n_{ip} is obtained. In practice, we can perform this operation without search as the set of interest points that are generated in this way form an ordered list.

The first entry in the list is the global maximum, which is not suppressed at any radius. As the suppression radius decreases from infinity, interest points are added to the list. However, once an interest point appears, it will always remain in the list. This is true because if an interest point is a maximum in radius r then it is also a maximum in radius $r' < r$.



(a) Strongest 250

(b) Strongest 500

(c) ANMS 250, $r = 24$

(d) ANMS 500, $r = 16$

Figure 3: *Adaptive non-maximal suppression (ANMS). The two upper images show interest points with the highest corner strengths, while the lower two images show interest points selected with adaptive non-maximal suppression (along with the corresponding suppression radius r). Note how the latter features have a much more uniform spatial distribution across the image.*

In practice we robustify the non-maximal suppression by requiring that a neighbour has a sufficiently larger strength. Thus the minimum suppression radius r_i is given by

$$r_i = \min_j |\mathbf{x}_i - \mathbf{x}_j|, \text{ s.t. } f(\mathbf{x}_i) < c_{\text{robust}} f(\mathbf{x}_j), \ \mathbf{x}_j \,\epsilon\, \mathcal{I} \tag{10}$$

where $\mathbf{x}_i$ is a 2D interest point image location, and $\mathcal{I}$ is the set of all interest point locations. We use a value $c_{\text{robust}} = 0.9$, which ensures that a neighbour must have significantly higher strength for suppression to take place. We select the $n_{ip} = 500$ interest points with the largest values of r_i. Experiments on a large database of panoramic images suggest that distributing interest points spatially in this way, as opposed to selecting based on max corner strength, results in fewer dropped image matches.

$f_{-1,1}$	$f_{0,1}$	$f_{1,1}$
$f_{-1,0}$	$f_{0,0}$	$f_{1,0}$
$f_{-1,-1}$	$f_{0,-1}$	$f_{1,-1}$

Figure 4: *For subpixel accuracy, derivatives are computed from finite pixel differences in a* 3×3 *neighbourhood according to the equations of section 2.3*

2.3 Sub-pixel Accuracy

Interest points are located to sub-pixel accuracy by fitting a 2D quadratic to the corner strength function in a local 3×3 neighbourhood (at the detection scale) and finding its maximum.

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x} \tag{11}$$

where $\mathbf{x}$ denotes position (x, y), and $f(\mathbf{x}) = f_{HM}(\mathbf{x})$ is the corner strength measure. Derivatives are computed from the 3×3 neighbourhood using finite pixel differences i.e.

$$\begin{aligned}
\frac{\partial f}{\partial x} &= (f_{1,0} - f_{-1,0})/2 \\
\frac{\partial f}{\partial y} &= (f_{0,1} - f_{0,-1})/2 \\
\frac{\partial^2 f}{\partial x^2} &= f_{1,0} - 2f_{0,0} + f_{-1,0} \\
\frac{\partial^2 f}{\partial y^2} &= f_{0,1} - 2f_{0,0} + f_{0,-1} \\
\frac{\partial^2 f}{\partial x \partial y} &= (f_{-1,-1} - f_{-1,1} - f_{1,-1} - f_{1,1})/4
\end{aligned}$$

See figure 4. The subpixel location is given by

$$\mathbf{x}_m = \mathbf{x}_0 - \frac{\partial^2 f}{\partial \mathbf{x}^2}^{-1} \frac{\partial f}{\partial \mathbf{x}} \tag{12}$$

2.4 Repeatability

The fraction of interest points whose transformed position is correct up to some tolerance epsilon is known as repeatability [SMB00]. We use a slightly different definition of repeatability to that defined in [SMB00] (which is not symmetric) as follows. Let I_M denote the set of all points belonging to image M, and $\mathcal{I}_M$ denote the set of interest points in image M. The set of points from image M that project to image N is given by $\mathcal{P}_{MN}$

$$\mathcal{P}_{MN} = \{\mathbf{x}_i : \mathbf{H}_{NM}\mathbf{x}_i \,\epsilon\, I_N\} \tag{13}$$

where $\mathbf{H}_{NM}$ is the homography between images M and N. The set of points from image M that are repeated in image N (within tolerance ϵ) is given by $\mathcal{R}_{MN}(\epsilon)$

$$\mathcal{R}_{MN}(\epsilon) = \{\mathbf{x}_i : \exists j : |\mathbf{x}_i - \mathbf{H}_{MN}\mathbf{x}_j| < \epsilon,\ \mathbf{x}_i \,\epsilon\, \mathcal{I}_M,\ \mathbf{x}_j \,\epsilon\, \mathcal{I}_N\} \tag{14}$$

The repeatability is the number of interest points that *are* repeated as a fraction of the total number of interest points that *could* be repeated. It is useful to adopt a symmetrical definition

$$r(\epsilon) = \min(\frac{|\mathcal{R}_{MN}|}{|\mathcal{P}_{MN}|}, \frac{|\mathcal{R}_{NM}|}{|\mathcal{P}_{NM}|}) \tag{15}$$

The repeatability of our interest points with and without sub pixel localisation is shown in figure 5. Note that sub-pixel localisation gives approximately a 5% improvement in repeatability.

2.5 Orientation

Each interest point has an orientation θ, where the orientation vector $[\cos\theta, \sin\theta] = \mathbf{u}/|\mathbf{u}|$ comes from the smoothed local gradient

$$\mathbf{u}_l(x, y) = \nabla_{\sigma_o} P_l(x, y) \tag{16}$$

The integration scale for orientation is $\sigma_o = 4.5$. A large derivative scale is desirable so that the motion field $\mathbf{u}_l(x, y)$ varies smoothly across the image, making orientation estimation robust to errors in interest point location. The orientation estimate is poorly conditioned if the first derivative is close to zero, in which case it may be favourable to look at higher order derivatives [SF95]. This is left for future work.

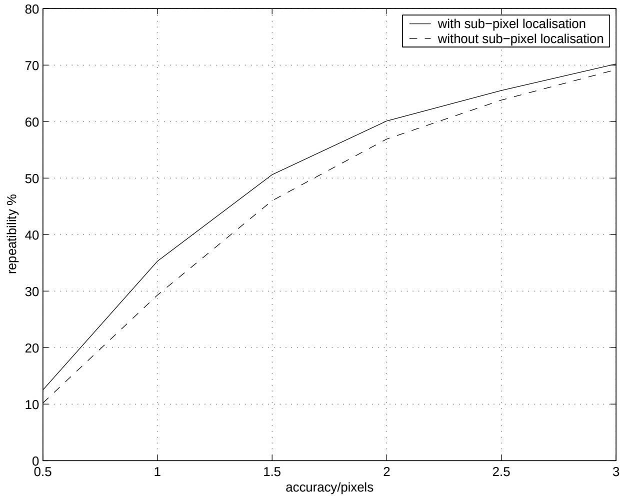


Figure 5: *Repeatability of interest points with and without sub-pixel correction. These results were computed from the Matier dataset.*

2.6 Analysis of Feature Extraction

Figure 6 compares the errors introduced in four stages of feature extraction: position, scale and orientation measurement, and descriptor matching. These experiments were conducted using the Matier dataset (see appendix C). Features were extracted and matched between all 7 images, and the top 2 image matches for each image were selected. The maximum number of matches per feature was 5. For each of these (14) image matches, the number of features in the area of overlap was found, and the number of features with consistent position, scale and orientation measurements computed. Consistent postition means that the interest point was detected within ϵ pixels of the projected position using the homographies computed from bundle adjustment. Consistent scale means that the interest point was detected at the same scale in the two images. Consistent orientation means that the transformed orientations differ by less than 3 standard deviations (= 3 × 18.5 degrees). To an accuracy of 3 pixels, 72% of interest points are repeated (have correct position), 66% have the correct position and scale, 64% also have correct orientation, and in total 59% of interest points are correctly matched (meaning they are one of the top 5 matches in terms of Euclidean distance in feature space). That is, given that an interest point overlaps another image, the probability that it will be correctly matched is 59%.

Whilst figure 6 shows combined results for all levels, figure 7 shows separate results for interest points extracted at each level of the pyramid. Note that contrary to the popular

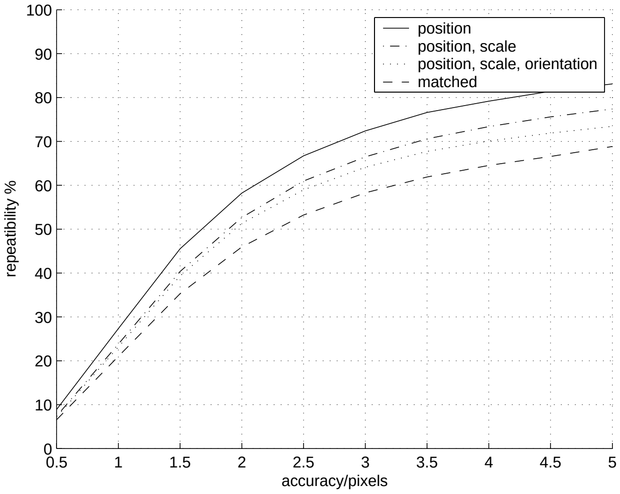
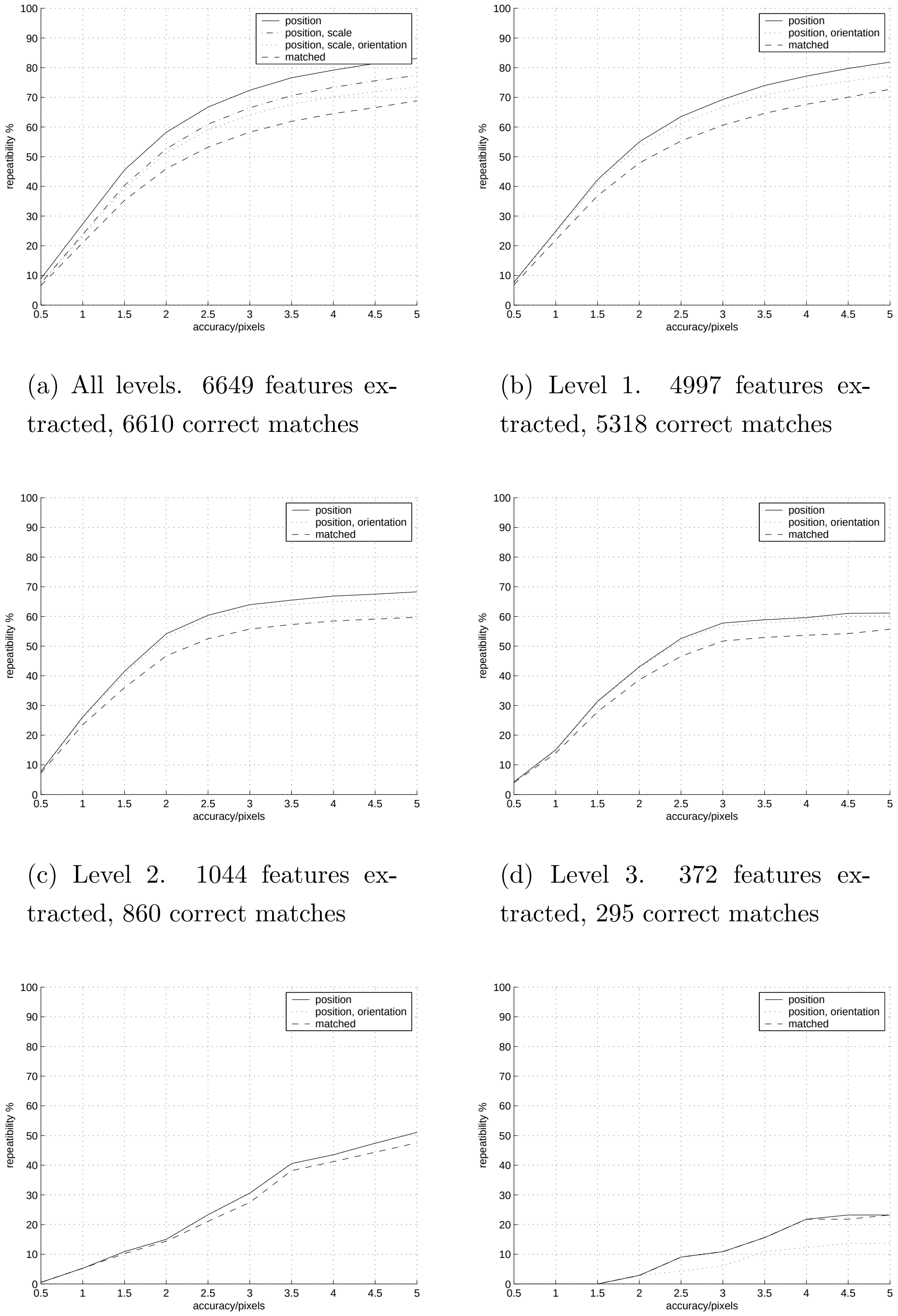


Figure 6: *Repeatability vs accuracy for Multi-scale Oriented Patches. To an accuracy of 3 pixels, 72% of interest points in the overlap region have consistent position, 66% have correct position and scale, 64% also have correct orientation, and in total 59% of interest points in the overlap region are correctly matched.*

perception that Harris corners are sub-pixel accurate, the majority of interest points have location errors in the 1-3 pixel range, even at the finest scale of detection. Also note that interest points at higher levels of the pyramid are less accurately localised relative to the base image than those at a lower level, due to the larger sample spacing. Although less useful for accurate localisation, these higher level features are still useful in verifying an image match or a coarse RANSAC hypothesis. Also, the orientation estimate improves as the level increases. As expected, features at levels 4 and 5 generally have poor accuracy, and their distributions show many features have accuracy worse than 3 pixels. However, it is slightly counter intuitive that features at levels 2 and 3 tend to have accuracies of 3 pixels or better.

Figure 8 show the same results as computed for the Van Gogh sequence. This is a pure rotation sequence, with no projective distortion. As compared to the Matier sequence, which has perspective distortion, matching is improved. Note in particular that the orientation repeatability curves and the matched curves are very close, indicating that if feature orientation is correctly estimated, then it is very likely that the feature will also be correctly matched. This is not the case for the Matier dataset due to perspective distortion.



(a) All levels. 6649 features extracted, 6610 correct matches

(b) Level 1. 4997 features extracted, 5318 correct matches

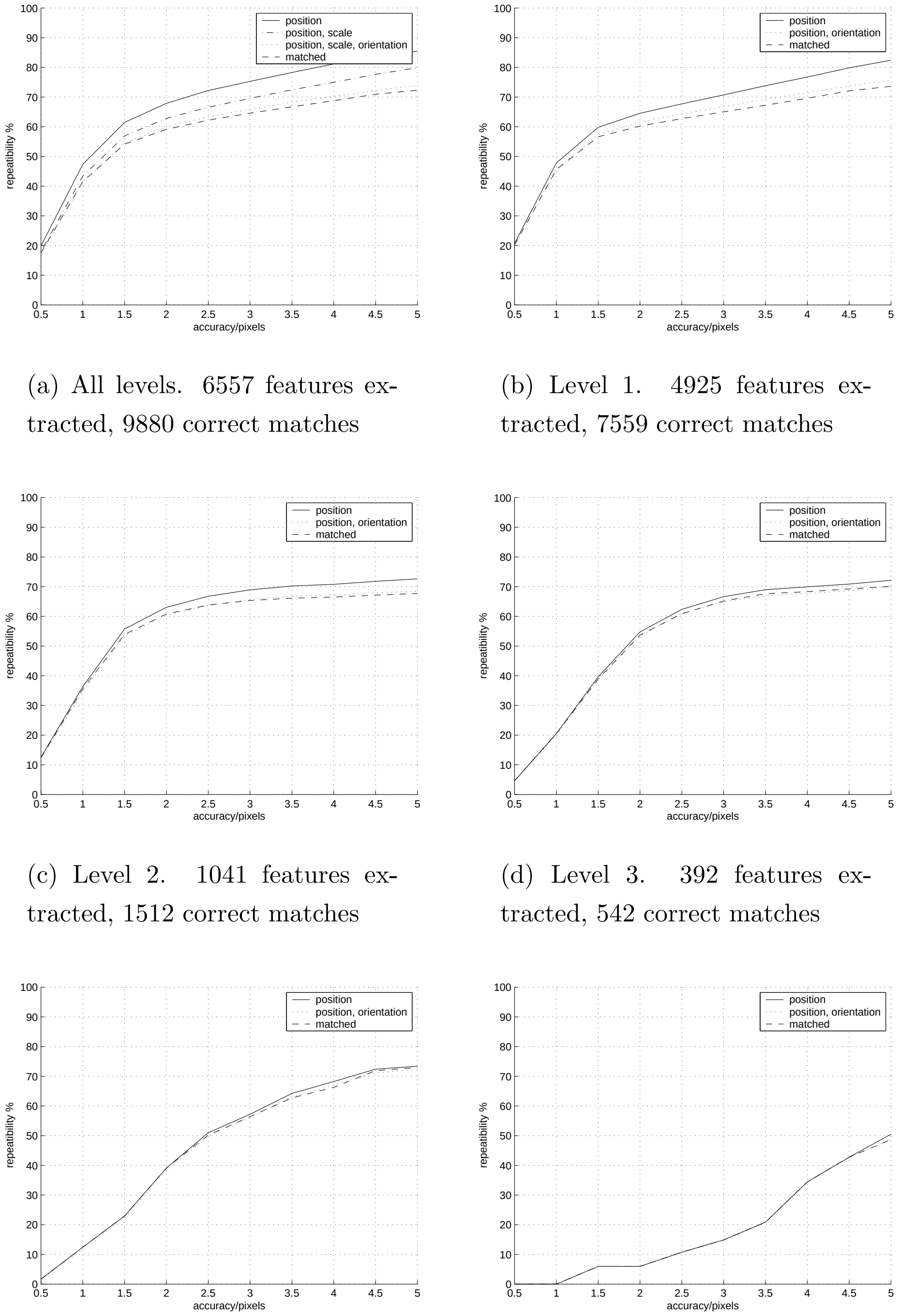
(c) Level 2. 1044 features extracted, 860 correct matches

(d) Level 3. 372 features extracted, 295 correct matches

(e) Level 4. 180 features extracted, 120 correct matches

(f) Level 5. 56 features extracted, 17 correct matches

Figure 7: *Repeatability of interest points, orientation and matching for multi-scale oriented patches at 5 pyramid levels (Matier dataset). The top left figure is a combined result for all levels.*



(a) All levels. 6557 features extracted, 9880 correct matches

(b) Level 1. 4925 features extracted, 7559 correct matches

(c) Level 2. 1041 features extracted, 1512 correct matches

(d) Level 3. 392 features extracted, 542 correct matches

(e) Level 4. 158 features extracted, 212 correct matches

(f) Level 5. 41 features extracted, 55 correct matches

Figure 8: *Repeatability of interest points, orientation and matching for multi-scale oriented patches at 5 pyramid levels (Van Gogh dataset). This dataset consists of pure rotations with no perspective distortion. The top left figure is a combined result for all levels.*

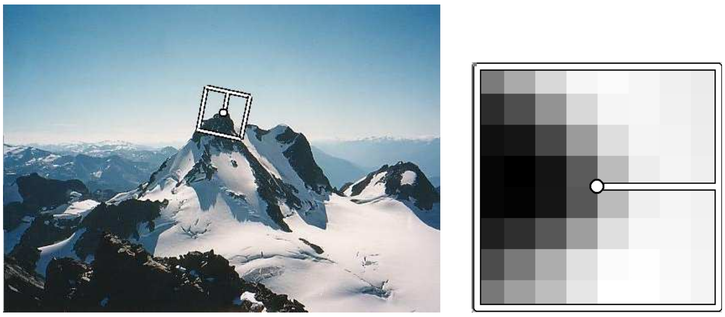


Figure 9: *Descriptors are formed using an* 8×8 *sampling of bias/gain normalised intensity values, with a sample spacing of 5 pixels relative to the detection scale. This low frequency sampling gives the features some robustness to interest point location error, and is achieved by sampling at a higher pyramid level than the detection scale.*

2.7 Feature Descriptor

Once we have determined where to place our interest points, we need to extract a description of the local image structure that will support reliable and efficient matching of features across images. A wide range of such local feature vectors have been developed, including local intensity patches [F̈86, Har92], Gaussian derivatives [SM97], shift invariant feature transforms [Low04], and affine-invariant descriptors [Bau00, TG00, BL02]. In their comparative survey, Mikolajczyk and Schmid [MS03] evaluated a variety of these descriptors and found that SIFT features generally perform the best. Local patches oriented to the dominant local orientation were also evaluated, but found not to perform as well. In this section, we show how such patches can be made less sensitive to the exact feature location by sampling the pixels at a *lower frequency* than the one at which the interest points are located.

Given an interest point (x, y, l, θ), the descriptor is formed by sampling from a patch centred at (x, y) and oriented at angle θ from pyramid level l. We sample an 8×8 patch of pixels around the sub-pixel location of the interest point, using a spacing of $s = 5$ pixels between samples (figure 9). Figure 10 shows how varying the sample spacing s affects the reliability of feature matching. We have found that performance increases up to a value $s = 5$, with negligible gains thereafter.

To avoid aliasing, the sampling is performed at a higher pyramid level, such that the sampling rate is approximately once per pixel (the Nyquist frequency). This means sampling the descriptor from a level l_s levels above the detection scale, where

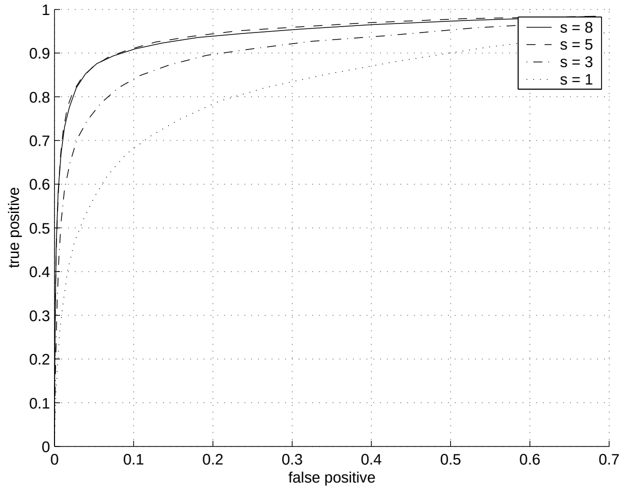


Figure 10: *Effect of changing the descriptor sample spacing on performance. These ROC curves show the results of thresholding feature matches based on normalised match distance as in section 3.1. Performance improves as the sample spacing increases (larger patches), but gains are minimal above a sample spacing of 5 pixels.*

$$l_s = \text{floor}\left(\frac{\log s}{\log 2} + 0.5\right) \tag{17}$$

The descriptor vector is sampled using bilinear interpolation. In practice, $s = 5$ so the descriptor vectors are sampled at $l_s = 2$ levels above the detection scale.

Suppose the interest point was detected at level l. This suggests sampling the descriptor from $P_{l+l_s}(x, y) = P_{l+2}(x, y)$. However, we have found better results by instead sampling the descriptor from $P'_{l+1}(x, y)$, where $P'_{l+1}(x, y) = P_{l+1}(x, y) * g_{\sigma_p}(x, y)$, i.e. blurring but not downsampling. Further (smaller) gains are made by sampling from $P''_l(x, y) = P_l(x, y) * g_{2\times\sigma_p}(x, y)$. Note that bilinear resampling corresponds to impulse sampling after convolution with a linear hat function. The extra blurring stage corresponds to sampling with a Gaussian blurred hat function at a finer scale. This is a better resampling kernel, resulting in less aliasing in the descriptors. This is discussed in section 4.3.

2.8 Normalisation

The descriptor vector is normalised so that the mean is 0 and the standard deviation is 1, i.e.

$$d_i = (d'_i - \mu)/\sigma \tag{18}$$

where $d'_i, \ \ i\epsilon\{1..d^2\}$ are the elements of the descriptor vector, with $\mu = \frac{1}{d^2}\sum_{i=1}^{d^2} d'_i$ and $\sigma = \sqrt{\frac{1}{d^2}\sum_{i=1}^{d^2}(d'_i - \mu)^2}$. This makes the features invariant to affine changes in intensity (bias and gain).

2.9 Haar Wavelet Transform

Finally, we perform the Haar wavelet transform on the 8×8 descriptor patch d_i to form a 64 dimensional descriptor vector containing the wavelet coefficients c_i. Due to the orthogonality property of Haar wavelets, distances are preserved

$$\sum_i (d_i^1 - d_i^2)^2 = \sum_i (c_i^1 - c_i^2)^2 \tag{19}$$

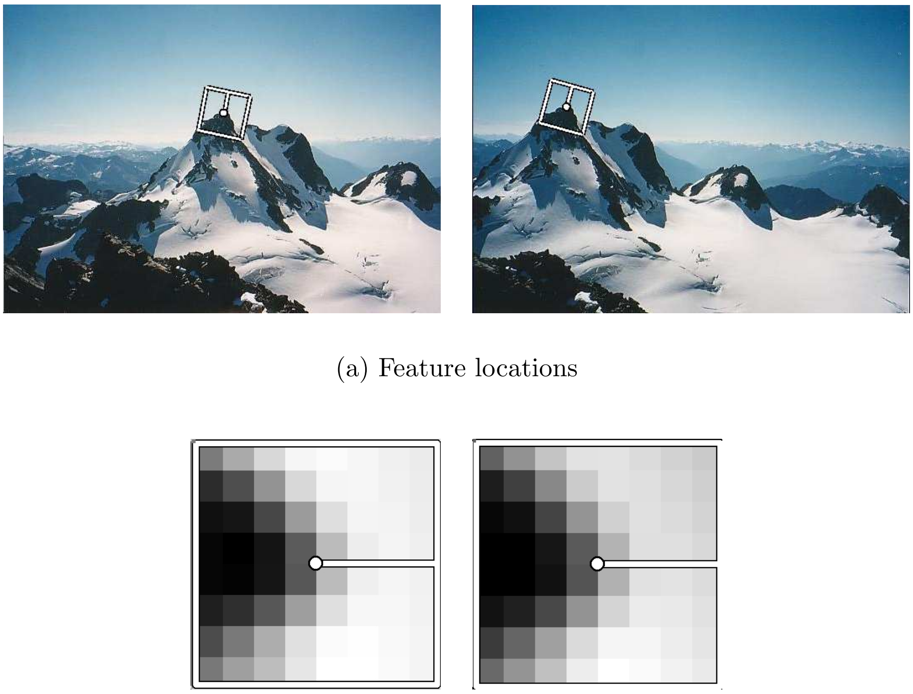
Therefore, nearest neighbours in a sum-squared difference sense are unchanged. The first 3 non-zero wavelet coefficients c_1, c_2, c_3 are used in the indexing strategy described in section 3.4.

3 Feature Matching

Given Multi-scale Oriented Patches extracted from all n images, the goal of the matching stage is to find geometrically consistent feature matches between all images. This proceeds as follows. First, we find a set of candidate feature matches using an approximate nearest neighbour algorithm (section 3.4). Then, we refine matches using an outlier rejection procedure based on the noise statistics of correct/incorrect matches. Finally we use RANSAC to apply geometric constraints and reject remaining outliers.

3.1 Feature-Space Outlier Rejection

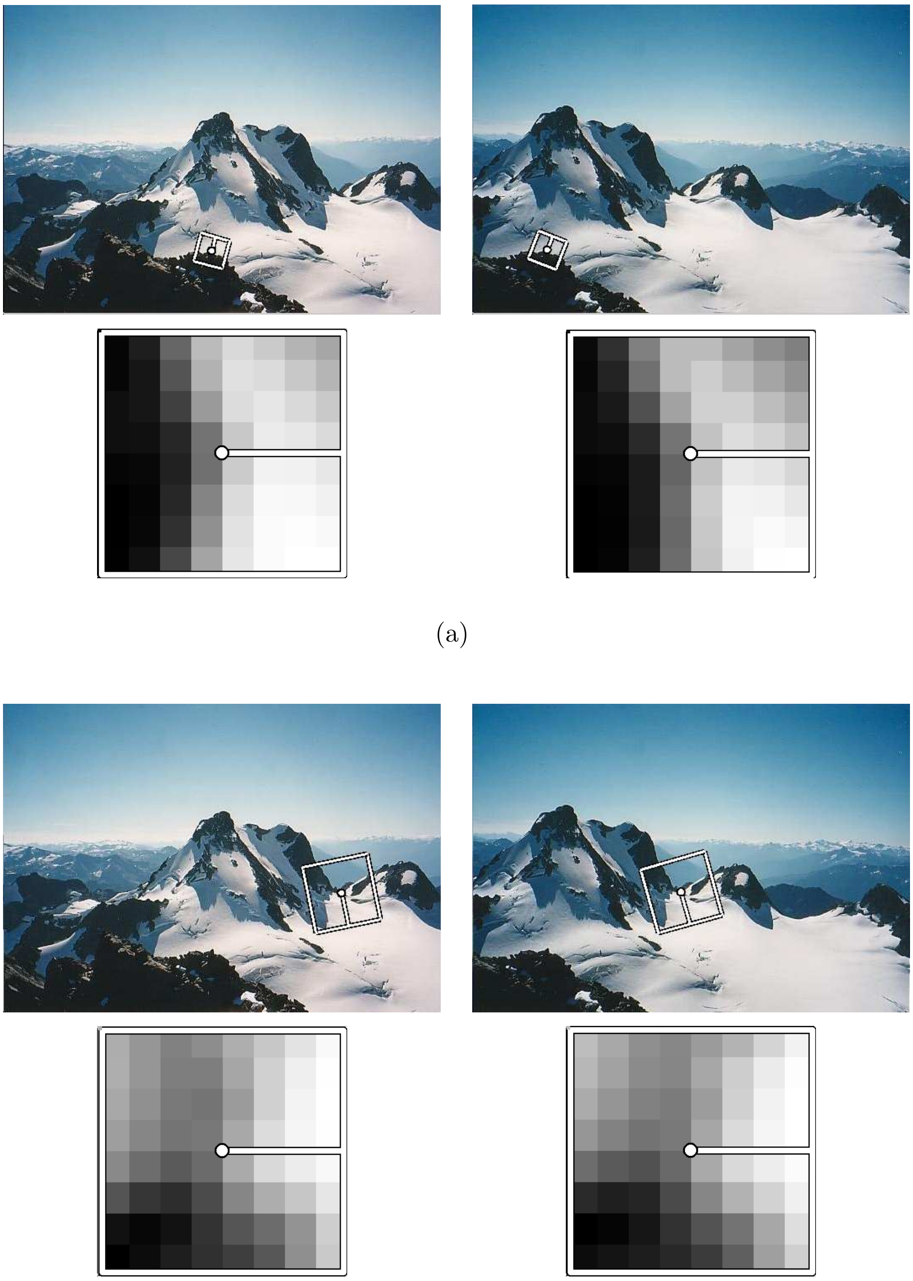
Our basic noise model assumes that a patch in one image, when correctly oriented, located and scaled, corresponds to a patch in the other image modulo additive Gaussian noise:



(a) Feature locations

(b) Feature descriptors

Figure 11: *Corresponding features in a pair of images. For each feature, a characteristic scale and orientation is established and an* 8×8 *patch of pixels sampled for the feature descriptor. Since the reference frame and the image undergo the same transformation between the images, the descriptor vector is the same in both cases (up to noise and modelling error).*



(a)

(b)

Figure 12: *Examples of corresponding features from different images in the Matier dataset. For each image, MOPS are extracted and descriptor vectors are stored in an indexing structure. Feature descriptors are indexed and matched as described in section 3.*

$$\begin{aligned} I'(\mathbf{x}') &= \alpha I(\mathbf{x}) + \beta + n(\mathbf{x}) && (20) \\ \mathbf{x}' &= \mathbf{A}\mathbf{x} + \mathbf{t} && (21) \\ \mathbf{A} &= s \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} && (22) \end{aligned}$$

$$n(\mathbf{x}) \sim \mathcal{N}(0, \sigma_n^2) \tag{23}$$

where $I(\mathbf{x})$ and $I'(\mathbf{x})$ are the corresponding patches, and $n(\mathbf{x})$ is independent Gaussian noise at each pixel. However, we have found this model to be inadequate for classification, since the noise distributions for correctly and incorrectly matching patches overlap significantly (see figure 14(a)). Hence, it is not possible to set a global threshold on the matching error $e = \sqrt{\sum_{\mathbf{x}} n(\mathbf{x})^2}$ in order to distinguish between correct and incorrect matches.

Note that the above results apply to the noise in the image plane, after correcting for the brightness changes α, β between patches. To compute α, β we assume that e is small so that taking the mean and variance of equation 20 gives

$$\begin{aligned} \alpha &= \frac{\sigma'}{\sigma} && (24) \\ \beta &= \mu' - \alpha\mu && (25) \end{aligned}$$

where $\mu, \sigma, \mu', \sigma'$ are the means and variances of patches $I(\mathbf{x})$ and $I'(\mathbf{x}')$ respectively.

We have also repeated this experiment using a Gaussian noise model in feature space

$$n(\mathbf{x}) = \left| \frac{I_1(\mathbf{x}_1) - m_1}{\sigma_1} - \frac{I_2(\mathbf{x}_2) - m_2}{\sigma_2} \right| \tag{26}$$

and found similar results.

This behaviour has also been observed by Lowe [Low04], who suggested thresholding instead on the ratio e_{1-NN}/e_{2-NN}. Here e_{1-NN} denotes the error for the best match (first nearest neighbour) and e_{2-NN} denotes the error for the second best match (second nearest neighbour). As in Lowe's work, we have also found that the distributions of e_{1-NN}/e_{2-NN} for correct and incorrect matches are better separated than the distributions of e_{1-NN} alone (figure 14(b)).

The intuition for why this works is as follows. For a given feature, correct matches always have substantially lower error than incorrect matches. However, the overall scale of errors varies greatly, depending upon the appearance of that feature (location in feature space). For this reason it is better to use a discriminative classifier that compares correct and incorrect matches for a particular feature, than it is to use a uniform Gaussian noise model in feature space.

Lowe's technique works by assuming that the 1-NN in some image is a potential correct match, whilst the 2-NN in that same image is an incorrect match. In fact, we have observed that the distance in feature space of the 2-NN and subsequent matches is almost constant[2]. We call this the *outlier distance*, as it gives an estimate of the matching distance (error) for an incorrect match (figure 15).

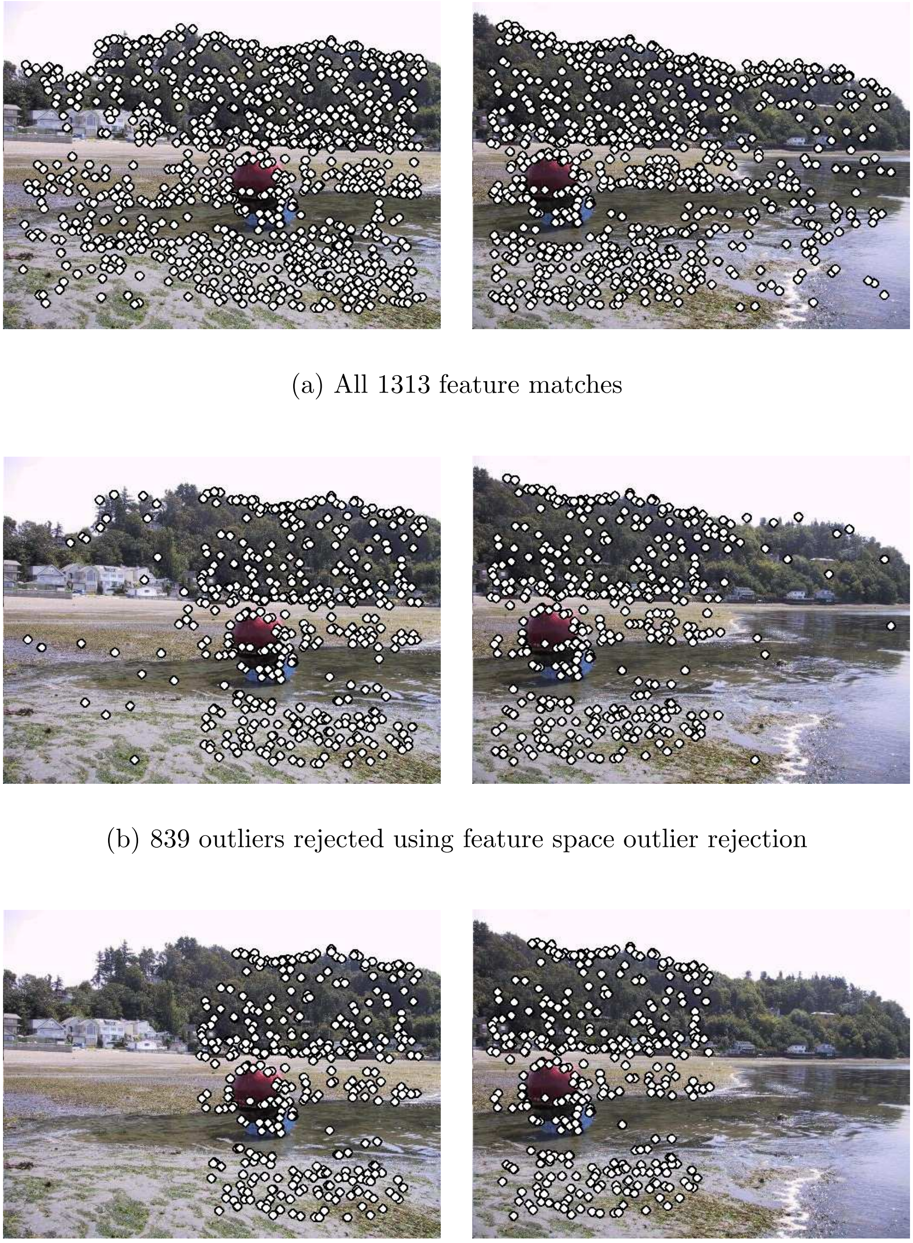
We have found that in the n image matching context (e.g., in panoramic image stitching), we can improve outlier rejection by using information from all of the images (rather than just the two being matched). Using the same argument as Lowe, the 2-NN from each image will almost certainly be an incorrect match. Hence we *average* the 2-NN distances from all n images, to give an improved estimate for the outlier distance. This separates the distributions for correct and incorrect matches still further, resulting in improved outlier rejection (figure 14(d)). Note that the extra overhead associated with computing 2-nearest neighbours in every image is small, since if we want to consider every possible image match, we must compute all of the 1-nearest neighbours anyway.

In practice we use a threshold $f = 0.65$ i.e. we accept a feature match with error e iff $e < f \times e_{outlier}$ where $e_{outlier}$ is the outlier distance. In general the feature-space outlier rejection test is very powerful. For example, we can eliminate 80% of the false matches with a loss of less than 10% correct matches. This allows for a significant reduction in the number of RANSAC iterations required in subsequent steps (see figure 13). These results are computed for Matier (7 images, 6649 features, 5970 correct matches), Van Gogh (7 images, 6557 features, 9260 correct matches) and Abbey (20 images, 18567 features, 15558 correct matches).

3.2 Spatial Variation of Errors

In actual fact, the errors between corresponding patches are not uniform across the patch as suggested in equation 23. We have also computed the error variance assuming a diagonal covariance model

[2] This is known as the shell property: the distances of a set of uniformly distributed points from a query point in high dimensions are almost equal. See appendix B.

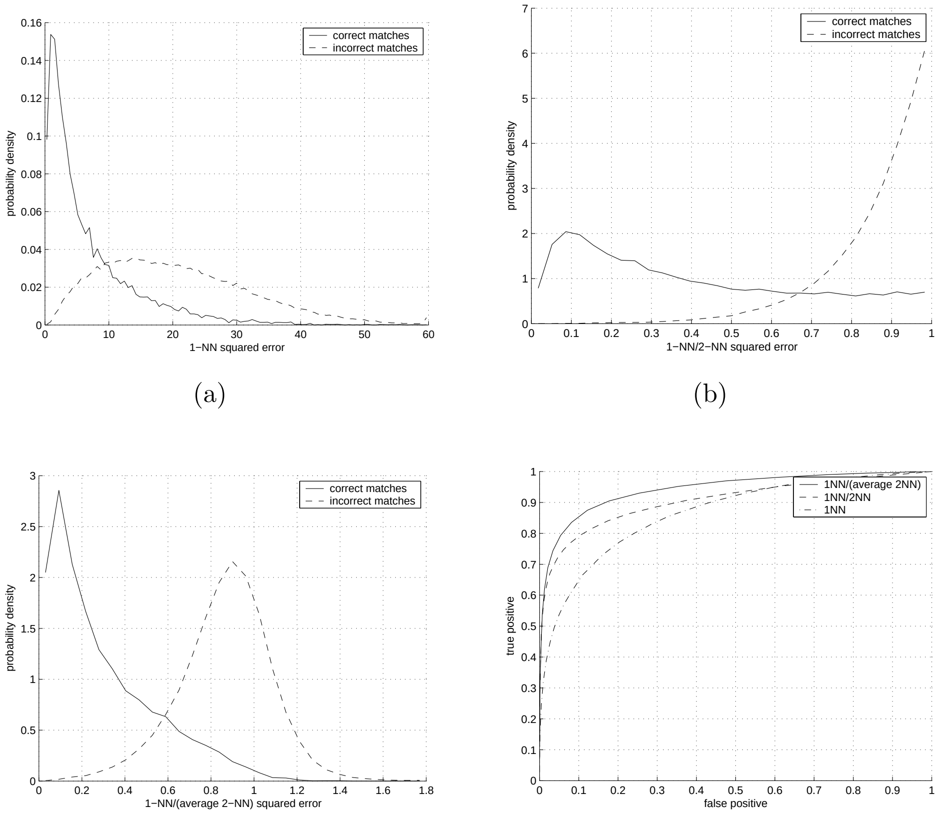


(a) All 1313 feature matches

(b) 839 outliers rejected using feature space outlier rejection

(c) A further 96 matches rejected using geometrical constraints

Figure 13: *Outlier rejection using b) feature space outlier rejection c) geometric constraints. The raw matches are a). There were 1313 initial matches, of which 839 were rejected without geometric constraints by thresholding based on the outlier distance, and a further 96 were rejected using geometric constraints. The input images are 385 × 512 and there were 378 matches in the final solution.*



(a)

(b)

(c)

(d)

Figure 14: *Distributions of matching error for correct and incorrect matches. Note that the distance of the closest match (the 1-NN) is a poor metric for distinguishing whether a match is correct or not (figure (a)), but the ratio of the closest to the second closest (1-NN/2-NN) is a good metric (figure (b)). We have found that using an average of 2-NN distances from multiple images (1NN/(average 2-NN)) is an even better metric (figures (c)-(d)). These results were computed from 18567 features in 20 images of the Abbey dataset, and have been verified for several other datasets.*

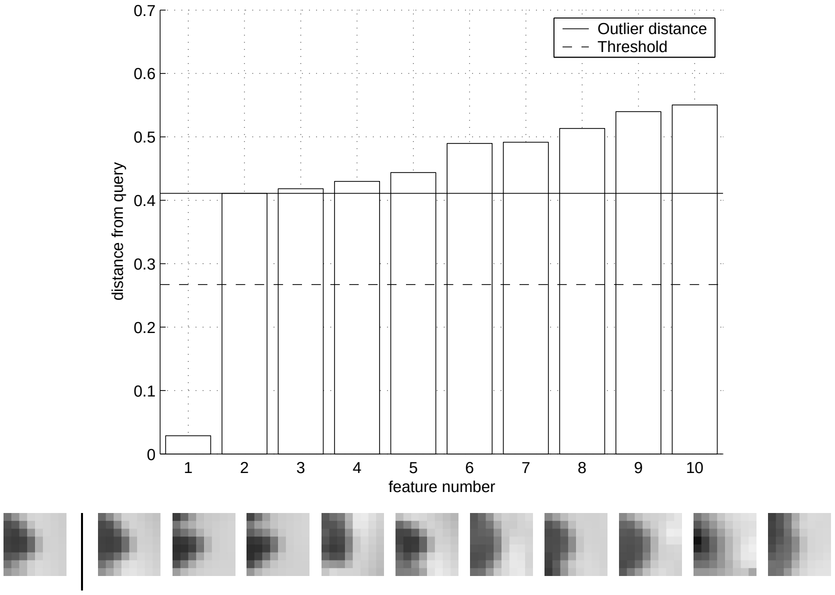


Figure 15: *Thresholding based on outlier distance. This figure shows the best 10 matches for a sample feature. The first is a correct match, and the rest are incorrect matches. Thresholding based purely on matching error gives poor results, since matching errors vary greatly depending upon the position in feature space. However, thresholding at a fraction of the outlier distance gives better results.*

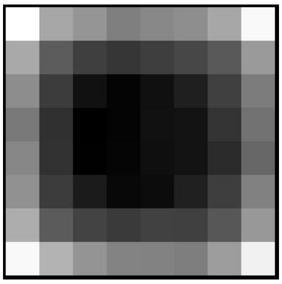


Figure 16: *Spatial variation of errors across the patch (for correct feature matches). Lighter tones indicate larger values of variance. The variance of the errors at the edge of the patches are larger than those in the centre. This is consistent with making small errors in scale / orientation selection.*

$$\mathbf{n}(\mathbf{x}) \sim \mathcal{N}(0, \mathbf{\Sigma}_n) \tag{27}$$

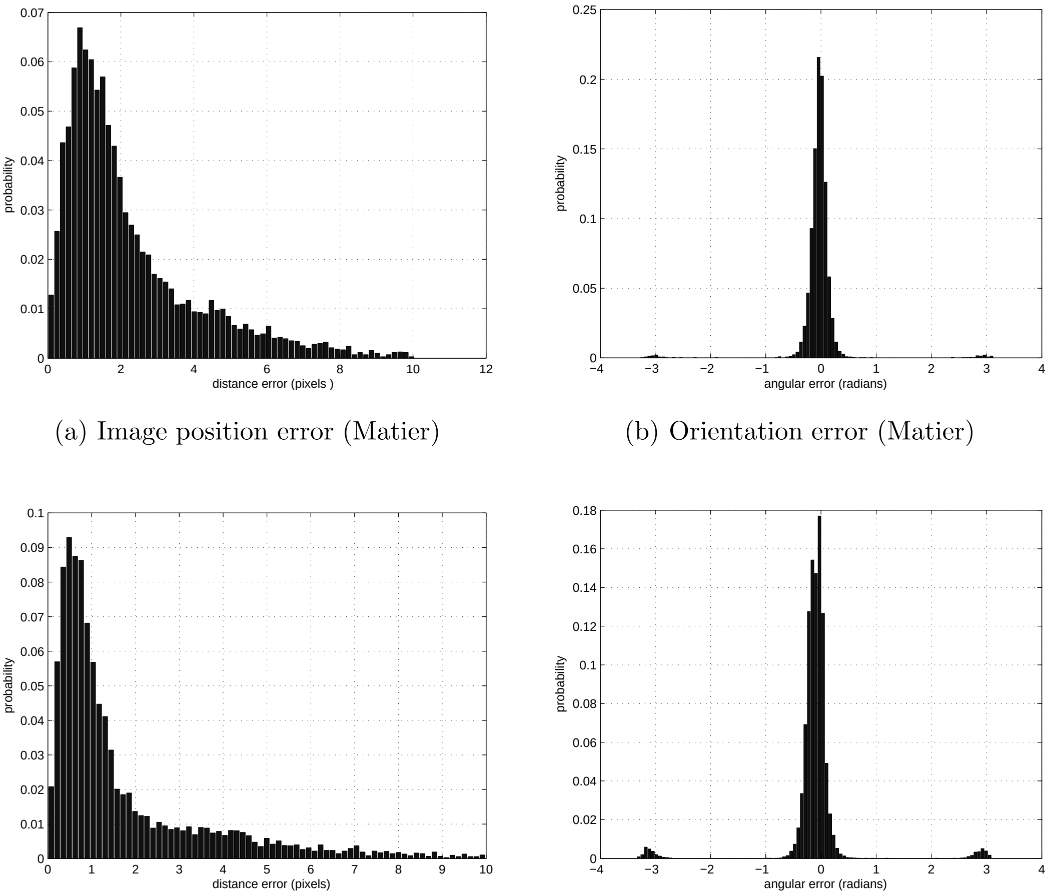
where

$$\mathbf{\Sigma}_n = \begin{bmatrix} \sigma_{11}^2 & 0 & 0 & \\ 0 & \sigma_{22}^2 & 0 & \dots \\ 0 & 0 & \sigma_{33}^2 & \dots \\ & \vdots & & \ddots \end{bmatrix} \tag{28}$$

If we assume that the error variance is constant across the patch ($\mathbf{\Sigma}_n = \sigma_n^2\mathbf{I}$) we find that the standard deviation of intensity errors for correct matches is $\sigma_n = 0.0334$ (for brightness values in the range $0 < I < 1$). That is, the error for correct matches is around 3 %. However, with the diagonal covariance model of equation 28 we find that the standard deviation of errors at the edge of the patch is approximately 2 times that in the centre (figure 16). Note that this is consistent with small errors in scale / orientation estimation for each patch, as these would generate larger errors at the edge of the patch. Preliminary experiments have shown that weighting the errors by their inverse standard deviation i.e. minimising $|\Sigma_n^{-\frac{1}{2}}\mathbf{n}(\mathbf{x})|^2$ does not give much improvement over simply minimising $|\mathbf{n}(\mathbf{x})|^2$. These results were computed using 7572 correctly matching features (RANSAC inliers with $\epsilon = 10$ pixels) from the Matier dataset.

3.3 Position and Orientation Errors for Correct Matches

Figure 17 shows the residual image position errors and errors in rotation estimates for correctly matching features. Note that the features from the (rotation only) Van Gogh dataset are more accurately located than those in the Matier dataset. This suggests that perspective distortion can adversely affect the feature location process. Also, there are a significant number of features that match correctly when the rotation estimate is 180° out. This suggests that some the of the features might be rotationally symmetric e.g. the 2×2 checkerboard pattern has the properties of ambiguous gradient and rotational symmetry. Features were projected between images using the 'gold standard' homographies after bundle adjustment of panoramic image sequences, before position and orientation errors between corresponding features were computed.



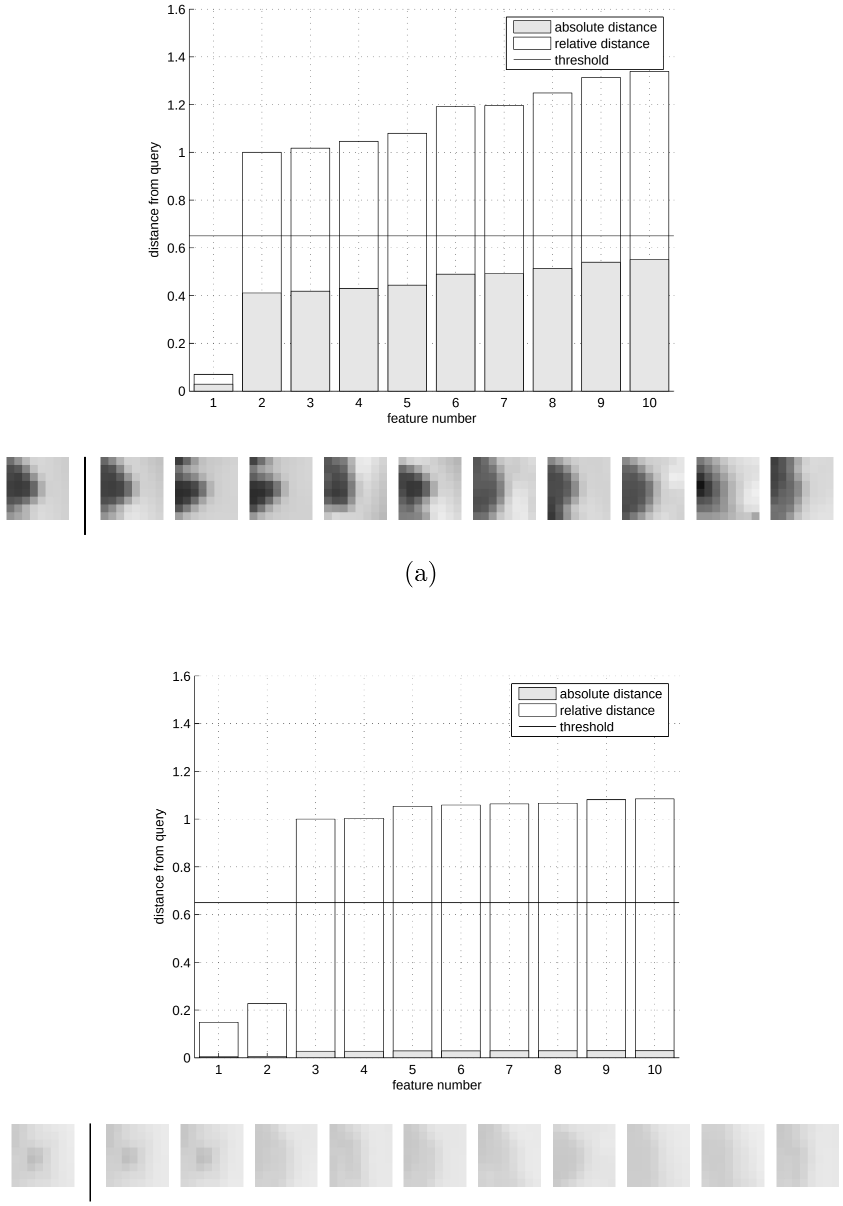
(a) Image position error (Matier)

(b) Orientation error (Matier)

(c) Image position error (Van Gogh)

(d) Orientation error (Van Gogh)

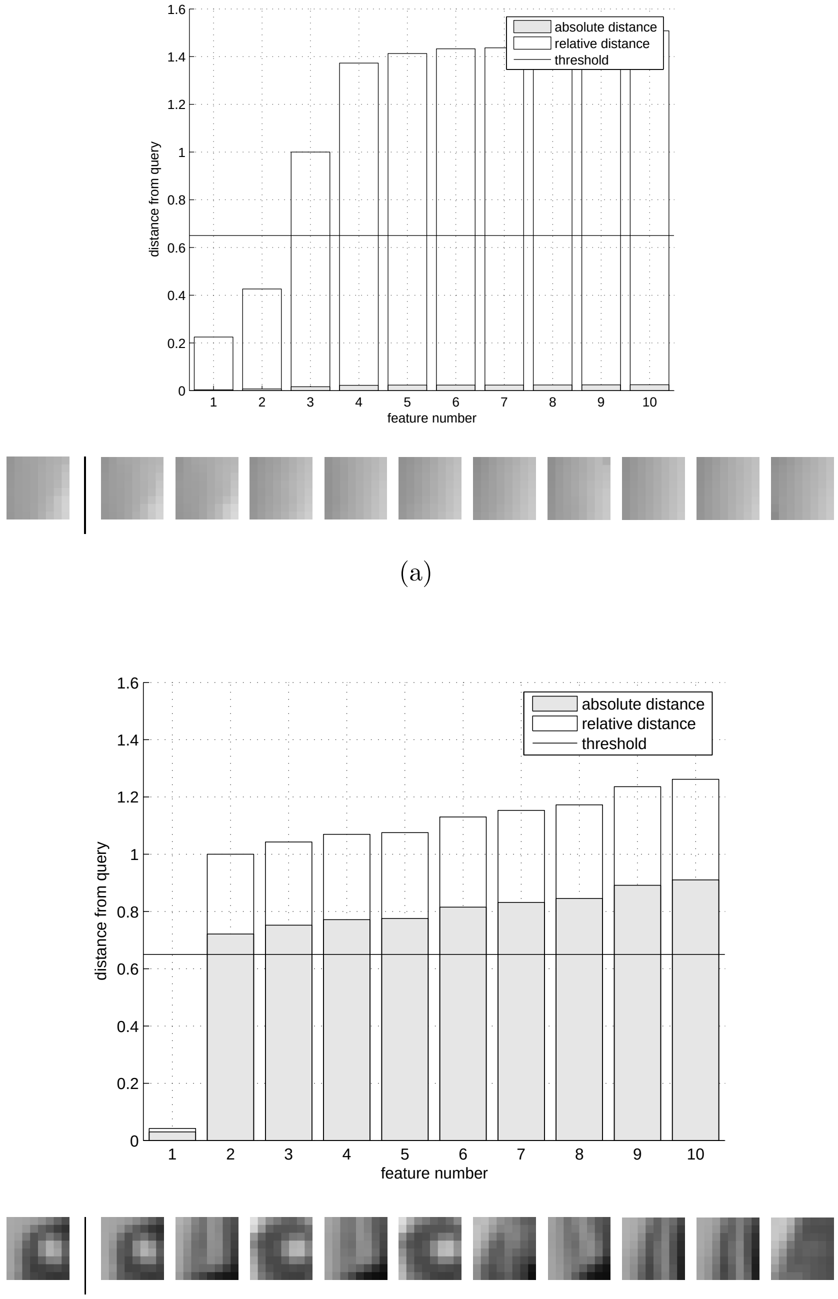
Figure 17: *Distributions of image location error and feature orientation error for correctly matching features. Note that features from the Van Gogh dataset are more accurately located. Also, there are a significant number of features that match correctly when the rotation estimate is* 180° *out.*



(a)

(b)

Figure 18: *Distances of correct and incorrect matches for high and low contrast features. Absolute distance is the square root of the sum-squared error between brightness normalised patches. Relative distance is the absolute distance relative to the outlier distance (closest 2-NN match from all other images). Note that for the high contrast features, the absolute distances are all much larger than for the low contrast features.*



(a)

(b)

Figure 19: *Distances of correct and incorrect matches for high and low contrast features. Absolute distances are larger for high contrast features than low contrast features. Hence, thresholding based on absolute match distances is a poor test, but thresholding on relative distances is better.*

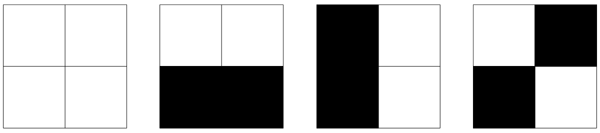


Figure 20: *Indexing is performed on the first 3 non-zero wavelet coefficients (the mean is 0). These represent the first derivatives in x and y and the second order cross derivative.*

3.4 Fast Approximate Nearest Neighbours using Wavelet Indexing

To efficiently find candidate feature matches, we use a fast nearest-neighbour algorithm based on wavelet indexing. Features are indexed in a three-dimensional lookup table with dimensions corresponding to the first 3 non-zero wavelet coefficients c_1, c_2, c_3 (estimates of $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial^2 I}{\partial x \partial y}$ over the patch) (see figure 20). The lookup table has $b = 10$ bins per dimension, which cover $\pm n_\sigma = 3$ standard deviations from the mean of that dimension. Note that the means are typically around zero except for the first derivative, which is aligned with the feature orientation and is hence significantly positive.

The bins are overlapped so that data within half a bin width, i.e. $\frac{2n_\sigma}{b-1}\frac{1}{2} = \frac{\sigma}{3}$, are guaranteed to be matched against the query. These are approximate nearest neighbours as it is possible (but unlikely) that the true nearest neighbour lies outside $\frac{\sigma}{3}$ in one of the 3 dimensions. The query is exhaustively matched to all features in the query bin, and k approximate nearest neighbours are selected. We then apply the outlier distance constraint as described in section 3.1 to verify correct matches and eliminate outliers. Indexing with b bins on 3 dimensions gives a speedup of $b^3/2^3$ (assuming features are evenly distributed in the histogram) at the expense of some lost feature matches.

Table 1 shows the percent recall for three indexing methods:

Wavelet low freq Indexing uses the first 3 non-zero wavelet coefficients

Pixel random Indexing uses 3 random grey values from the descriptor

Wavelet random Indexing uses 3 random wavelet coefficients from the descriptor

It is clear from the results that using the low frequency wavelet coefficients for indexing is most effective. Choosing 10 bins per dimension gives a speedup of $10^3/2^3 = 125$ compared to exhaustive nearest neighbour matching, with the loss of less than 10% of the matches.

Indexing Method	*Dataset*	*Number of bins / dimension*			
		1	*5*	*10*	*15*
Wavelet low freq	Matier	100	99.6	91.4	72.4
	Dash point	100	99.8	93.2	76.8
	Abbey	100	99.9	95.1	80.2
Pixel random	Matier	100	97.9	79.8	57.8
	Dash point	100	96.4	74.0	52.9
	Abbey	100	96.7	77.8	56.3
Wavelet random	Matier	100	84.4	49.2	28.1
	Dash point	100	81.5	42.8	25.6
	Abbey	100	83.0	45.4	24.6

Table 1: *Indexing on wavelet coefficients vs. pixel values - percent recall in database matching. Using 10 bins per dimension, indexing on the 3 non-zero low frequency Haar wavelet coefficients (x and y derivative and the cross derivative) gives about 10% better recall than indexing on random dimensions (pixels) of the descriptor.*

Indexing using low frequency wavelet coefficients is 10-20% better than the other methods at this operating point.

3.5 Outlier Rejection using Geometric Constraints

Once tentative matches have been established between the images, one can in principle apply any sort of geometrical matching constraint, depending on how the images were taken. If the images were taken from a point and the scene is static, a panoramic motion model (homography) is appropriate. If the images were taken with a moving camera and static scene, a full 3D motion model (fundamental matrix) is appropriate. It would also be possible to devise more elaborate motion models for multiple or moving objects. For automatic panorama stitching, we use the panoramic motion model and probabilistic model for matching similar to the one described in [BL03].

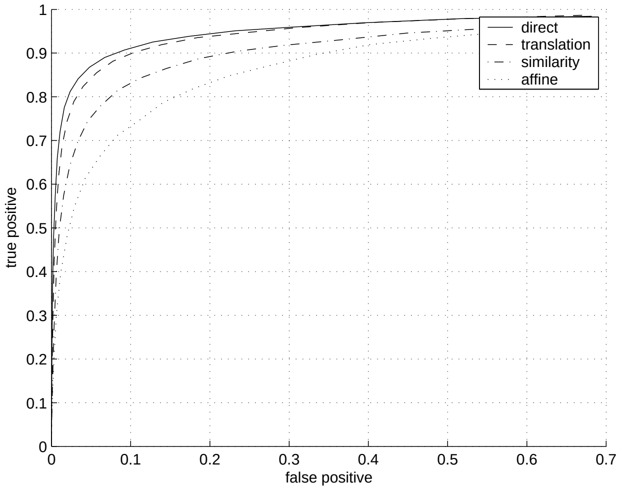


Figure 21: *ROC curves for patch refinement with different alignment models (Matier dataset). Each additional free parameter degrades the matching performance.*

4 Experimental Results

4.1 Panoramic Image Stitching

We have successfully tested our multi-image matching scheme on a dataset containing hundreds of panoramic images. We present results for the Matier, Abbey and Dash Point datasets in appendix C. See http://www.research.microsoft.com/∼szeliski/StitchingEvaluation for more examples.

4.2 Patch Refinement

In [MS03], Mikolajczyk and Schmid note that "It would be interesting to include correlation with patch alignment which corrects for these errors and to measure the gain obtained by such an alignment." Since sensitivity to localization errors has been touted as one of the weaknesses of pixel-based descriptors, we decided to implement this suggestion to see how much it would help. Rather than computing sum-squared error on pixel patches (or wavelet coefficients) directly, we included a stage of Lucas-Kanade [LK81] refinement to bring the patches more closely into spatial alignment before computing the pairwise descriptor distance. Since this has elements in common with the use of tangent distances [SLDV96] we expected that there might be an improvement in the separation of good and bad matches. Instead we found the opposite to be true.

Relative	*Extra*	*Number of feature matches*		
Sampling Level	*Smoothing*	*Matier*	*Dash Point*	*Abbey*
0	0	2620	2323	7467
-1	σ	3017	2988	9078
-2	2σ	3139	3058	9268

Table 2: *Effect of Pyramid Downsampling on Feature Matching. We found better results when sampling the feature descriptor at a smoothed version of a finer pyramid level, when using bilinear resampling.*

We used four motion models (no motion (direct), translation, similarity and affine) with 0, 2, 4 and 6 parameters respectively. The results are shown in figure 21. Note that matching performance is degraded for each new parameter that is added to the model.

Since correct matches are already fairly well aligned, but bad matches typically have large errors, refinement tends to overfit the incorrect matches, whilst making only small improvements to the correct matches. This means that Lucas-Kanade refinement actually makes it more difficult to distinguish between correct and incorrect matches than before.

4.3 Pyramid Downsampling

Table 2 shows the effect of pyramid downsampling on feature matching. ‘Relative level’ means the level relative to the ideal sampling level, where one would sample exactly once per pixel in the pyramid. ‘Extra smoothing’ is the standard deviation of the extra Gaussian smoothing applied i.e. instead of sampling at some level l, we sample at level $l - 1$, but introduce extra smoothing with a Gaussian kernel standard deviation σ. In each case, exactly the same interest points were extracted, so the total number of feature matches that result is a good indication of how well the descriptors are working. The results show that simple bilinear resampling is inferior to resampling using a Gaussian blurred hat function at a lower pyramid level. These gains must be balanced against the cost of the extra convolution operations.

4.4 Comparison to SIFT features

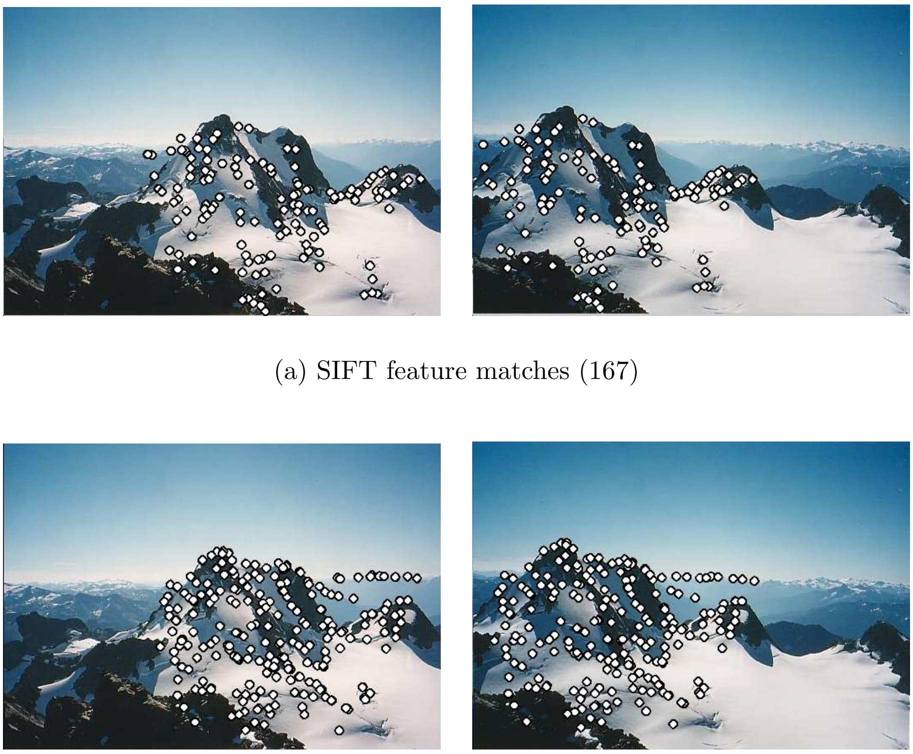
To compare Multi-Scale Oriented Patches (MOPS) and SIFT features, we used 3 datasets of panoramic images. For each method, we extracted approximately the same number of

Dataset		*MOPS*	*SIFT*
Matier	#interest points	3610	3678
	#matches	3776	4344
	#matches/interest point	1.05	1.18
Dash point	#interest points	32689	32517
	#matches	11750	22928
	#matches/interest point	0.36	0.71
Abbey	#interest points	15494	15710
	#matches	18659	21718
	#matches/interest point	1.20	1.38

Table 3: *Comparison of Multi-Scale Oriented Patches and SIFT feature matching. Note that SIFT features have a larger number of matches per interest point for each of the 3 datasets.*

interest points from each image, and then exhaustively matched them to find $k = 4$ exact nearest neighbour matches for each feature. I then used identical RANSAC algorithms to find the number of correct feature matches. In each case we have tabulated the number of correct matches per feature. The results are given in table 3. Note that both methods could potentially find more matches by using more scales / adjusting interest point strength thresholds etc.

From the results of table 3 it seems that in terms of number of matches per interest point SIFT features outperform MOPS. Why is this? Lowe [Low04] reports higher repeatability for his difference of Gaussian interest points (around 90% compared to 70% for our Harris corners), although this is highly dataset dependent. The rotation estimate used in SIFT features (maxima of a histogram of local gradients) is more robust since multiple orientations can be assigned if the histogram peaks are close. Lowe reports repeatability of around 80% for position (to an accuracy of 3 pixels), orientation and scale compared to our value of 58%. Another issue is that SIFT features are found to be located very close to the borders of the image, but since MOPS use relatively large image patches they are constrained to be at least 20 pixels from the image edge (this is the main reason SIFT performs much better on the dash point dataset). This is shown in figure 23. Finally, the SIFT descriptor is more robust to affine change than patch correlation due to accumulating measurements in spatial histograms, and this has been verified for various interest point detectors by Mikolajczyk [MS03].

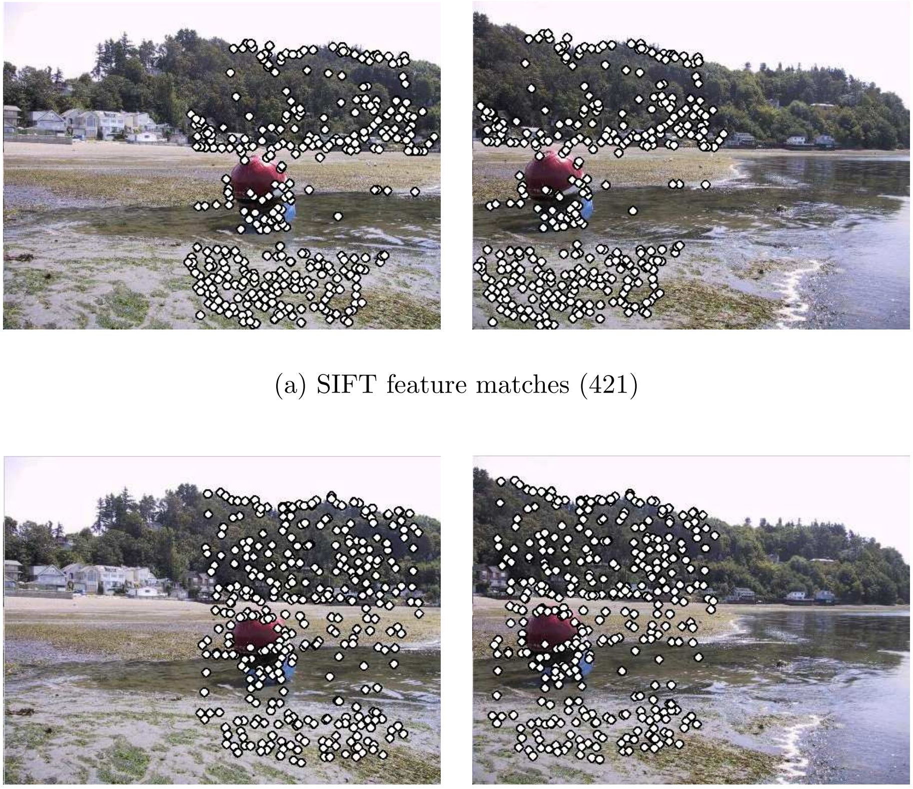


(a) SIFT feature matches (167)

(b) MOPS feature matches (238)

Figure 22: *Comparison of SIFT features and MOPS features. Note that MOPS features concentrate on edges/corners, whereas SIFT features concentrate on blobs.*

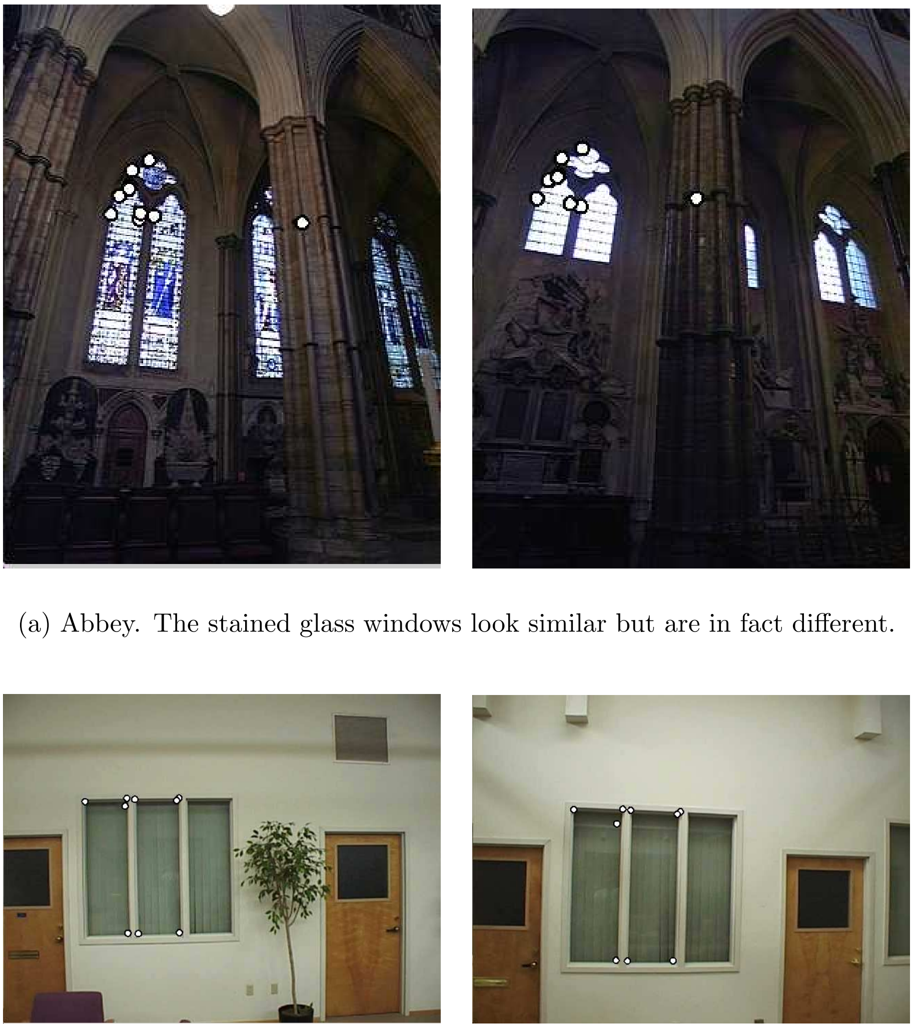
Note however, that MOPS and SIFT features tend to concentrate on different areas in the images. In particular, the DOG detector used for SIFT makes it more likely to find 'blobs' - bright areas surrounded by dark pixels or vice versa, whereas the autocorrelation detector used for MOPS makes it more likely to find edge or corner like features. This suggests that a combination of the two feature types might be effective. Also, it is important to note that MOPS features matches are by design well spatially distributed in the image. Sometimes SIFT feature matches are very close together. Are feature matches equally useful if they are very close together? It seems that some other criterion, such as registration accuracy, would be a better criterion for evaluating features. Another approach would be to compare the number of matched/dropped images in a panorama dataset.



(a) SIFT feature matches (421)

(b) MOPS feature matches (372)

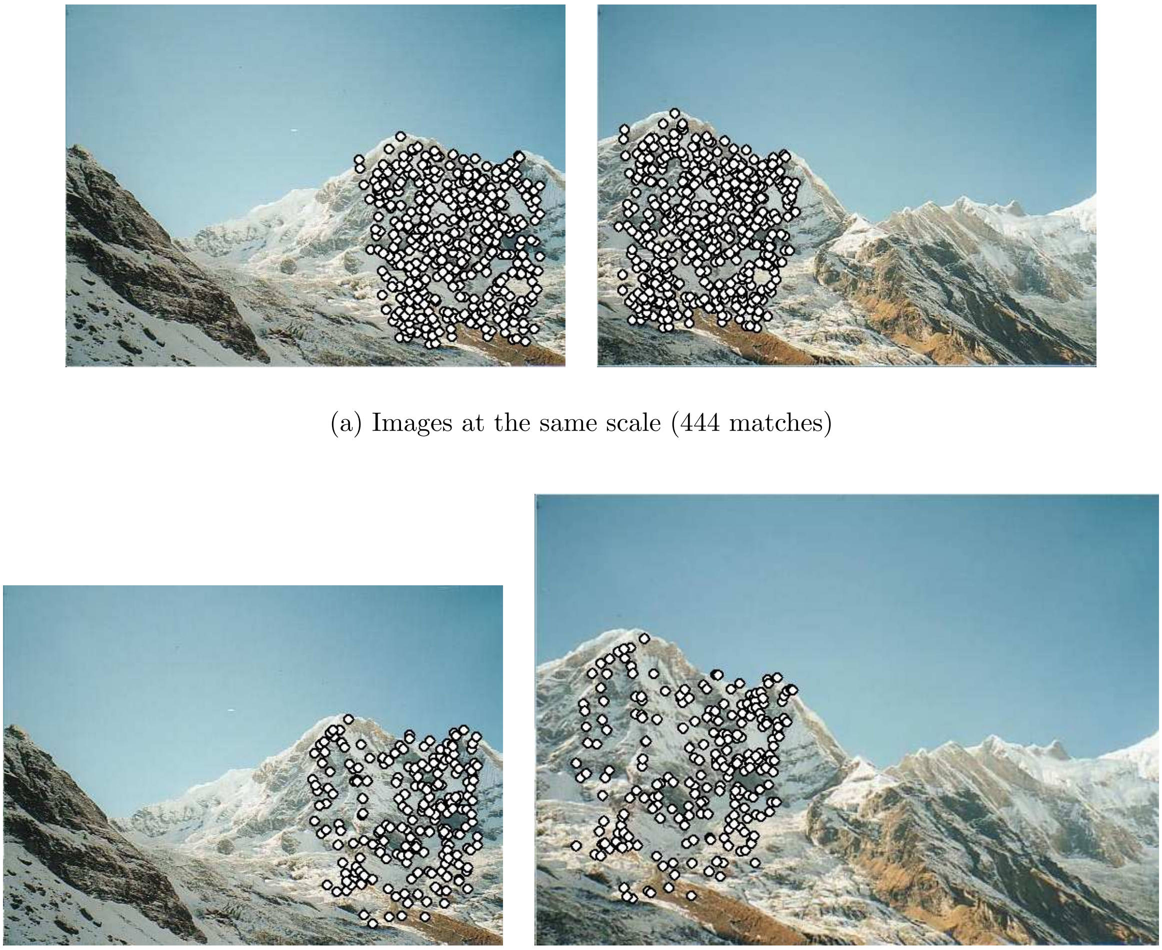
Figure 23: *Comparison of SIFT features and MOPS features. For the dashpoint dataset, the SIFT feature detector performed better in terms of number of matches. However, note that the MOPS feature matches are better spatially distributed e.g. there are many more matches in the sea/sand. Also note that the SIFT features are located right up to the edge of the image, but due to the large patches used in MOPS, the features are constrained to be at least 20 pixels away from an image edge.*



(a) Abbey. The stained glass windows look similar but are in fact different.

(b) Office. Here the tree gives away the fact that these images don't in fact match!

Figure 24: *Matching mistakes. These are often caused by repeating structures or similar looking objects that appear in multiple views.*



(a) Images at the same scale (444 matches)

(b) Images differing in scale by 1.25 (238 matches)

Figure 25: *Effect on matching of scale changes. Here we have use a pyramid subsampling of 1.5, and an image scaling of 1.25. Matching of MOPS features seems to be adversely affected by scale changes. One possible solution is to use true scale space features e.g. DOG maxima, located to sub-scale accuracy. Harris corners have the disadvantage that they tend to repeat across scales.*

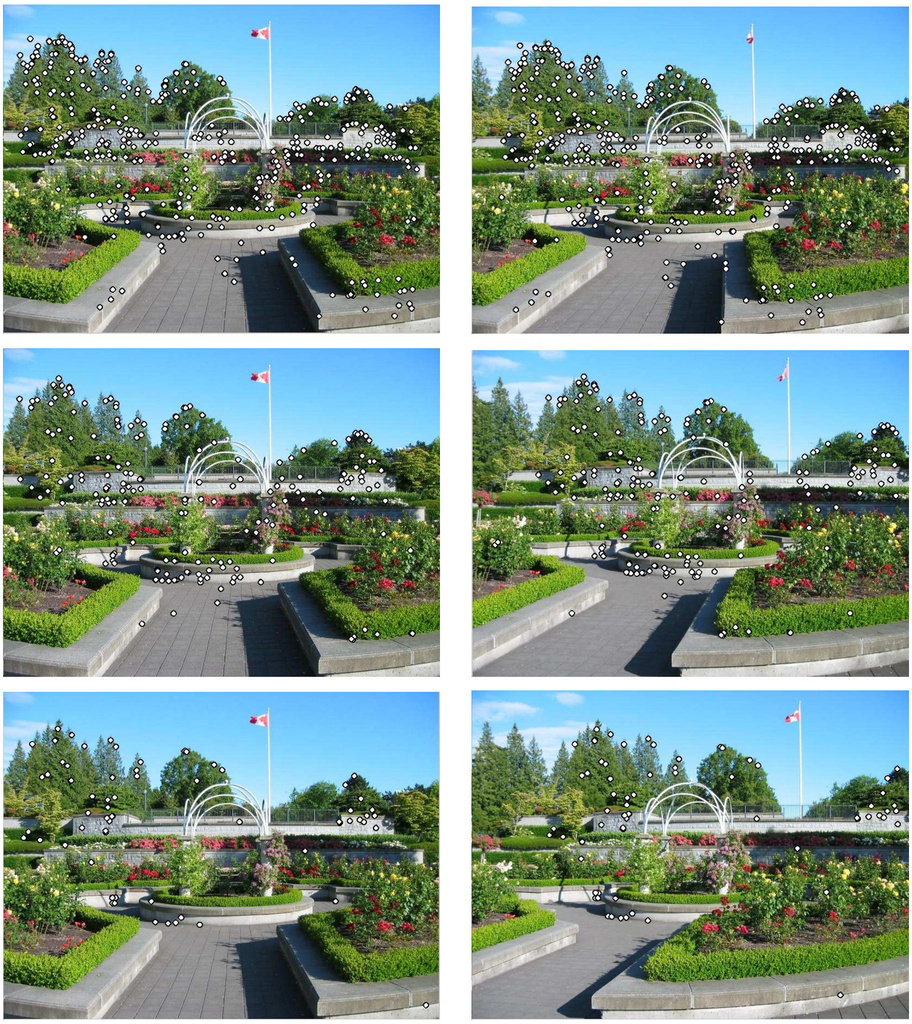


Figure 26: *Matching images taken from different viewpoints. Shown are inliers to the fundamental matrix estimated between each pair. The number of matches in each case are 433, 202, 70 with approximately equal increments of 15 degrees between the views. Note that feature matching works well under small viewpoint changes but drops off as the viewpoint change becomes large. This is probably due to affine deformations and partial occlusions.*

5 Conclusions

We have presented a new type of invariant feature, which we call Multi-Scale Oriented Patches (MOPs). These features utilise a novel adaptive non-maximal suppression algorithm for interest point location, and a simple sampling of the (oriented) local image intensity for the feature descriptor. We have also introduced two innovations in multi-image matching. First, we have demonstrated an improved test for verification of pairwise image matches that uses matching results from all n images. Second, we have shown that an indexing scheme based on low frequency wavelet coefficients yields a fast approximate nearest neighbour algorithm that is superior to indexing using the raw data values.

5.1 Future Work

In future work, we plan to explore a number of promising ideas for extending our basic technique:

Evaluation and Comparison of Image Stitching Algorithms. We plan to develop a principled framework for comparing image stitching algorithms by comparing stitching results to ground truth based on projection errors. This could be used to form more thorough comparisons between SIFT/MOPS features, to perform quantative evaluation of algorithm choices such as adaptive non-maximal suppression, and to tune algorithm parameters.

Orientation estimation. This is currently problematic if the gradient is not well defined or rapidly spatially-varying around the interest point. Alternative methods could include: largest eigenvector of $\mathbf{H}$ (degeneracy for rotational symmetry), histogram of theta (can be made robust by using multiple peaks) or steerable filters.

Colour. We could use RGB features instead of greyscale, or just add a few dimensions of colour information (e.g. the average [R, G, B]/(R + G + B) for the patch) to the descriptors with an appropriate weighting.

Interest operators. In addition to using autocorrelation maxima, we could form features using other interest operators e.g. difference of Gaussian maxima, watershed regions, edge based features.

Other matching problems. In order to better cope with multi-scale matching, we should use a scale-space interest operator. To cope better with 3D matching problems we should introduce more robustness to affine change and relative shifting of edge positions.

References

[Ana89] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[Bau00] A. Baumberg. Reliable feature matching across widely separated views. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR00)*, pages 774–781, 2000.

[Bis95] C. Bishop. *Neural Networks for Pattern Recognition.* Clarendon, Oxford, UK, 1995.

[BL02] M. Brown and D. Lowe. Invariant features from interest point groups. In *Proceedings of the 13th British Machine Vision Conference (BMVC02)*, pages 253–262, Cardiff, 2002.

[BL03] M. Brown and D. Lowe. Recognising panoramas. In *Proceedings of the 9th International Conference on Computer Vision (ICCV03)*, volume 2, pages 1218–1225, Nice, October 2003.

[CJ03] G. Carneiro and A. Jepson. Multi-scale local phase-based features. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR03)*, 2003.

[F8̈6] W. Förstner. A feature-based correspondence algorithm for image matching. *Int.l Arch. Photogrammetry & Remote Sensing*, 26(3):150–166, 1986.

[Har92] C. Harris. Geometry from visual motion. In A. Blake and A. Yuille, editors, *Active Vision*, pages 263–284. MIT Press, 1992.

[LK81] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[Low99] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision (ICCV99)*, pages 1150–1157, Corfu, Greece, September 1999.

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[MCUP02] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the 13th British Machine Vision Conference (BMVC02)*, 2002.

[MS03] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR03)*, 2003.

[RZFM92] C.A. Rothwell, A. Zisserman, D.A. Forsyth, and J.L. Mundy. Canonical frames for planar object recognition. In *Proceedings of the 2nd European Conference on Computer Vision (ECCV92)*, pages 757–772, 1992.

[SF95] Eero P. Simoncelli and William T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *International Conference on Image Processing*, volume 3, pages 444–447, 23-26 Oct. 1995, Washington, DC, USA, 1995.

[SLDV96] Patrice Simard, Yann LeCun, John S. Denker, and Bernard Victorri. Transformation invariance in pattern recognition-tangent distance and tangent propagation. In *Neural Networks: Tricks of the Trade*, pages 239–27, 1996.

[SM97] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997.

[SMB00] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, June 2000.

[ST94] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR94)*, Seattle, June 1994.

[SZ02] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organise my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision (ECCV02)*, pages 414–431, 2002.

[TG00] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proceedings of the 11th British Machine Vision Conference (BMVC00)*, pages 412–422, Bristol, UK, 2000.

[Tri04] B. Triggs. Detecting keypoints with stable position, orientation, and scale under illumination changes. In *Proceedings of the 8th European Conference on Computer Vision (ECCV04)*, volume 4, pages 100–113, Prague, May 2004.

Appendix

A Parameters

The following are the parameter values we currently use for our feature detector:

Features

$r = 2$	Pyramid subsampling
$\sigma_p = 1.0$	Pyramid smoothing
$\sigma_d = 1.0$	Derivative scale
$\sigma_i = 1.5$	Integration scale for harris corners
$t = 10.0$	Harris corner strength threshold
$n_{ip} = 500$	Maximum number of interest points per image
$\sigma_o = 4.5$	Integration scale for orientation measure
$c_{robust} = 0.9$	Threshold for adaptive non-maximal suppression
$d = 8$	Descriptor vector is $d \times d$ elements
$s = 5$	Sample spacing at detection scale (pixels)

Matching

$k = 4$	Maximum number of matches per feature
$n = 6$	Maximum number of matches per image
$b = 10$	Number of bins per dimension in histogram
$n_\sigma = 3$	Number of standard deviations represented in histogram
$f = 0.65$	Threshold (fraction of outlier distance) for outlier rejection

B The Shell Property

In high-dimensions, most of the volume of a hypersphere is concentrated in the outside shell. This means that for common distributions such as the Gaussian distribution, most of the probability mass is located in an outside shell ([Bis95] exercise 1.4). Consider a set of uniformly distributed points in a d-dimensional hypercube. The number of points within a distance r of the centre increases as r^d, until the boundary of the space is reached, whereupon it rapidly falls off. As d becomes large, this means that almost all points have the same distance from a given query point, i.e. they live in a thin shell at a constant distance from the query (the shell distance). Figure 27 illustrates. Note however, that the value of the shell distance depends upon the position of the query point in the space e.g. a point towards the edge has a larger shell distance than a point in the centre (see figure 28).

C Image Datasets

Figures 29–32 show three of the data sets (Matier, Dash Point, and Abbey) on which we tested our feature extraction, matching, and stitching algorithms, as well as the Van Gogh data set taken from [MS03]. Figure 33 shows the resulting stitched panoramas displayed on cylindrical compositing surfaces.

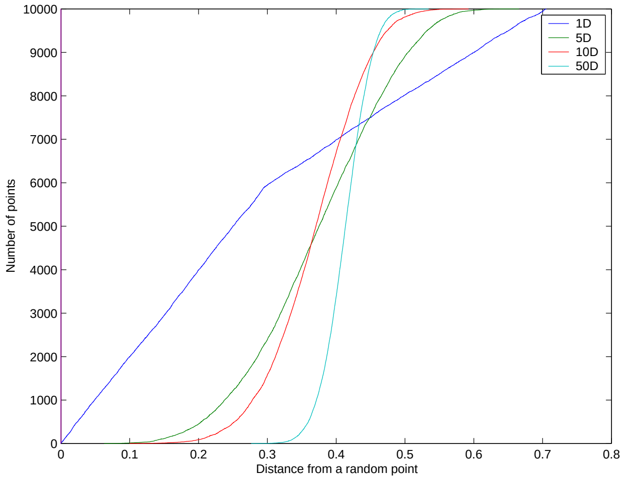


Figure 27: *The shell property. Distances of uniformly distributed points in a hypercube from a random point rapidly approach a single value as the number of dimensions increases. This is because all of the volume of the space is concentrated in a thin shell at the edge of the hypersphere.*

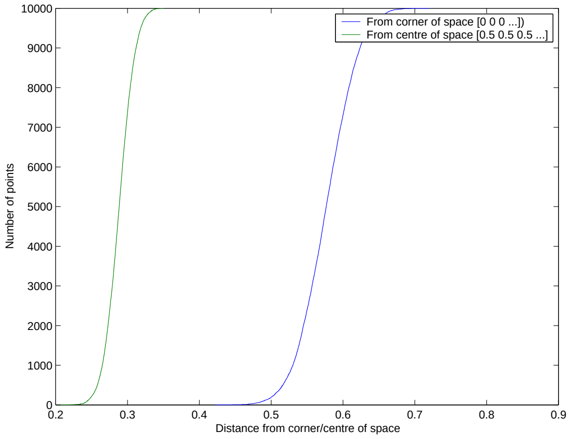


Figure 28: *The shell distance varies across the space. This figure shows distances of uniformly distributed points from the edge and centre of the space, in 50 dimensions. Note that though the shell property holds throughout the space, the shell distance is greater at the edge of the hypercube than in the centre.*

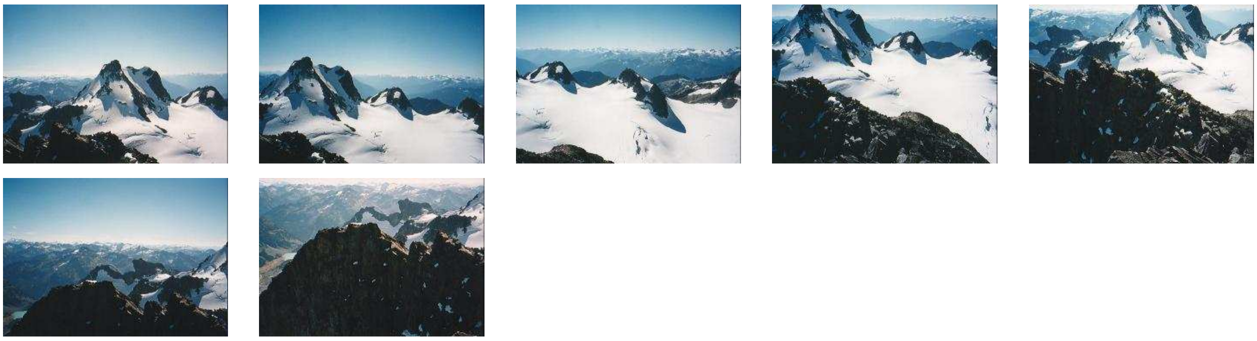


Figure 29: *Matier dataset*

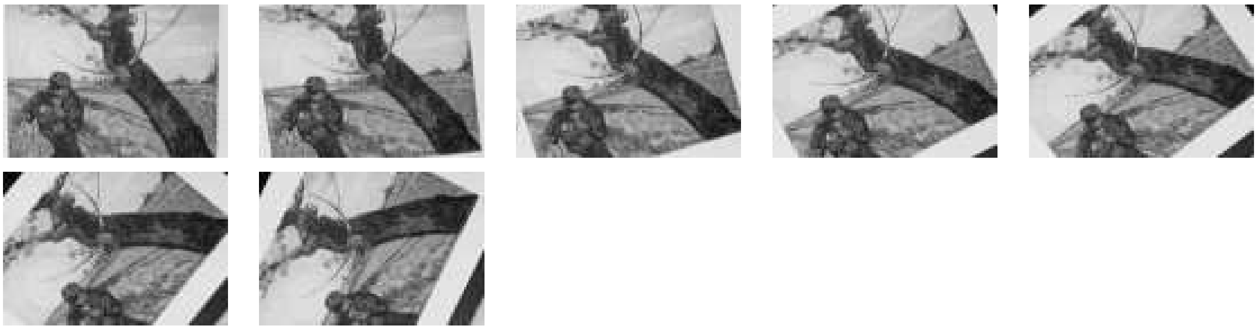


Figure 30: *Van Gogh dataset*

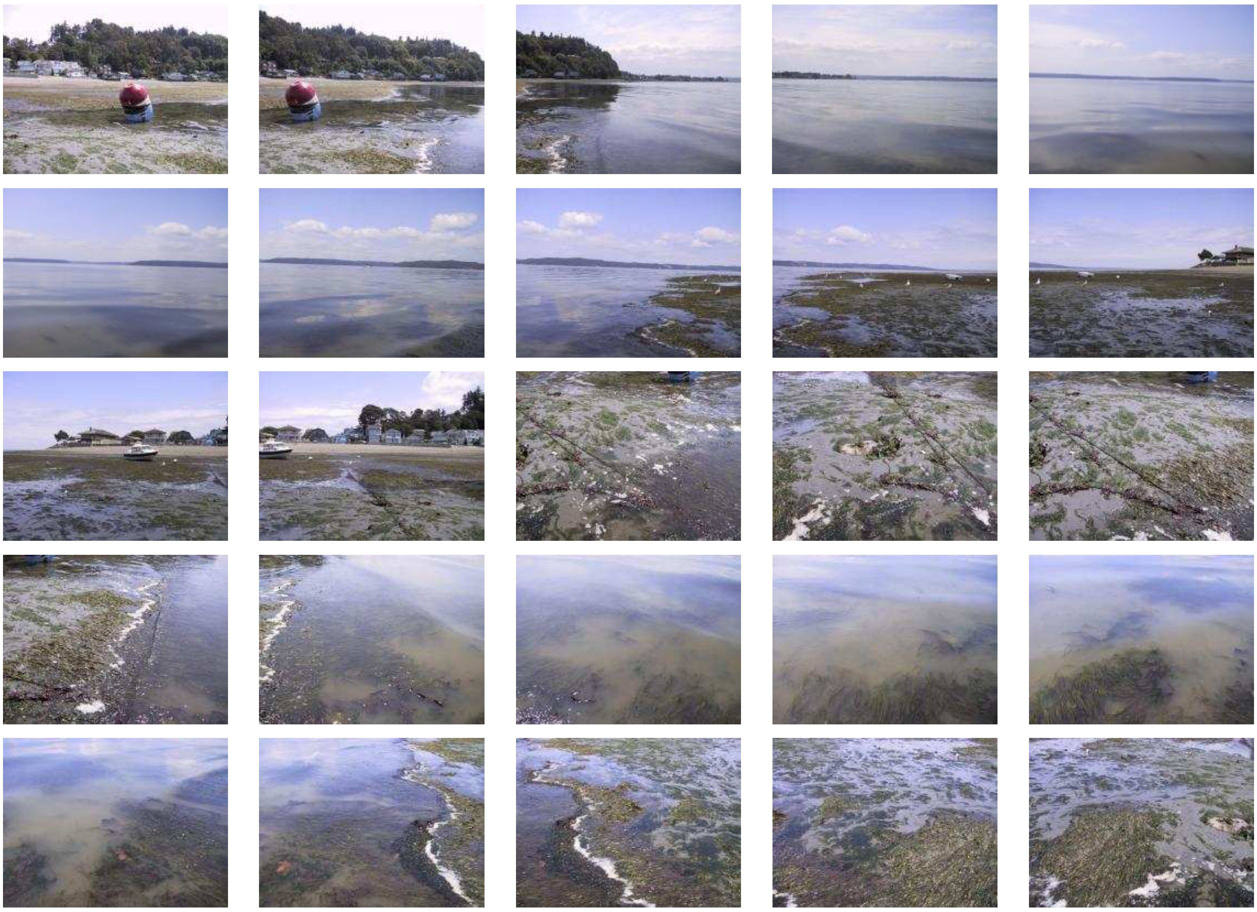


Figure 31: *Dash point dataset*

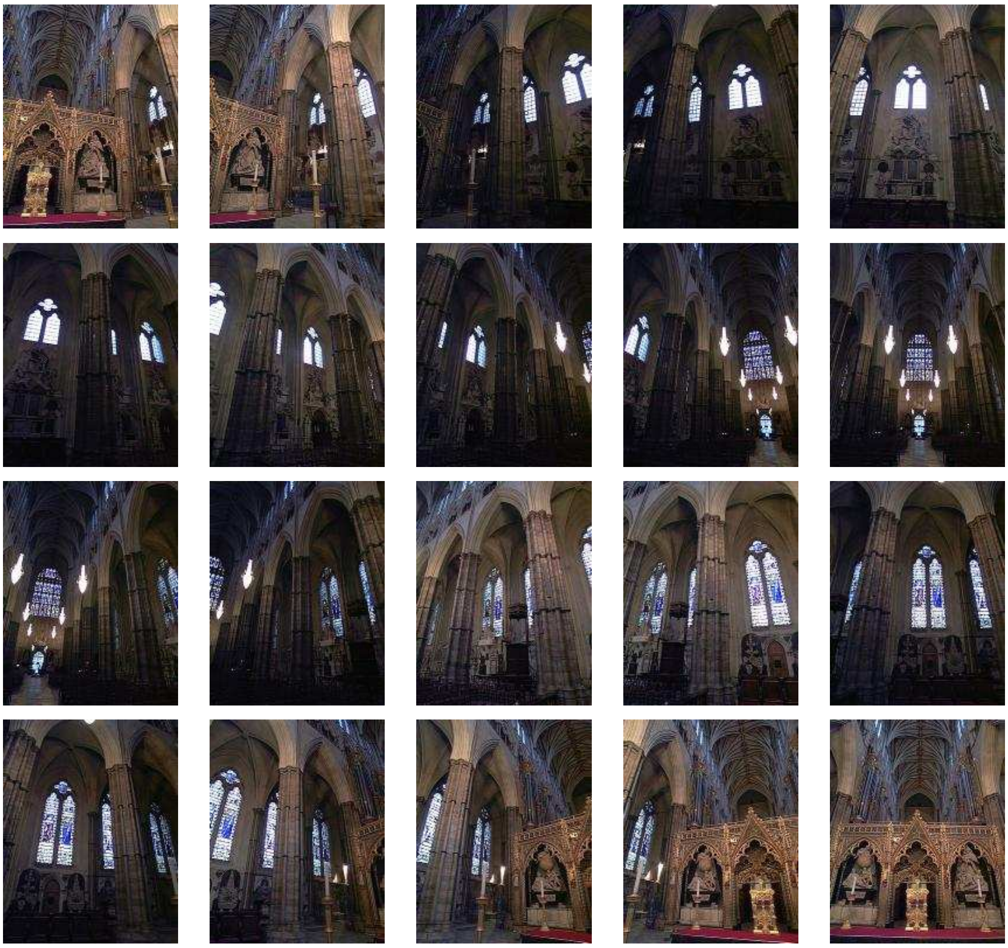
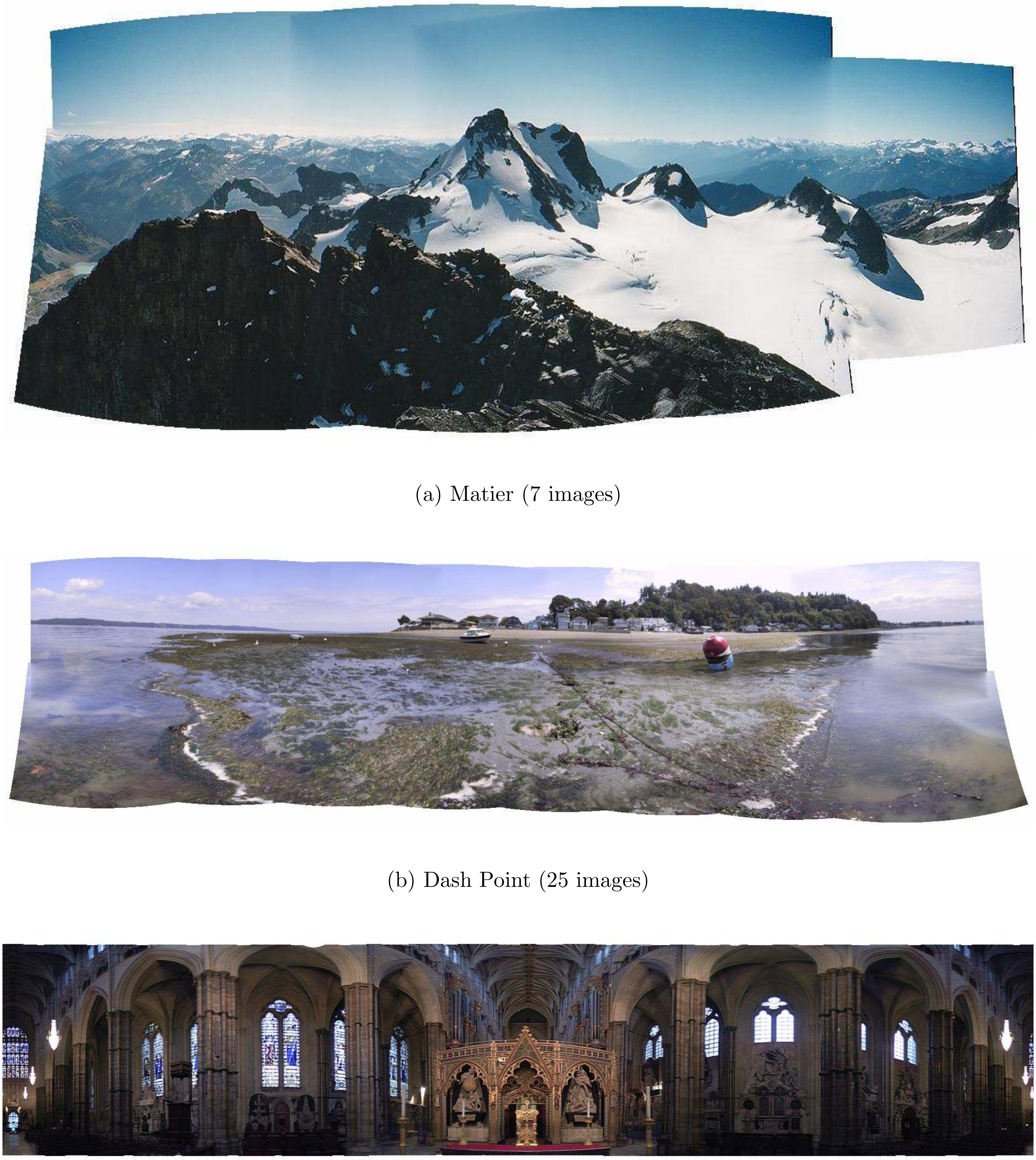


Figure 32: *Abbey dataset*



(a) Matier (7 images)

(b) Dash Point (25 images)

(c) Abbey (20 images)

Figure 33: *The stitched images used for the matching results found in this report.*