Robust Pairwise Matching of Interest Points With Complex Wavelets

Ee Sin Ng and Nick G. Kingsbury

***Abstract*— We present a matching framework to find robust correspondences between image features by considering the spatial information between them. To achieve this, we define spatial constraints on the relative orientation and change in scale between pairs of features. A pairwise similarity score, which measures the similarity of features based on these spatial constraints, is considered. The pairwise similarity scores for all pairs of candidate correspondences are then accumulated in a 2-D similarity space. Robust correspondences can be found by searching for clusters in the similarity space, since actual correspondences are expected to form clusters that satisfy similar spatial constraints in this space. As it is difficult to achieve reliable and consistent estimates of scale and orientation, an additional contribution is that these parameters do not need to be determined at the interest point detection stage, which differs from conventional methods. Polar matching of dual-tree complex wavelet transform features is used, since it fits naturally into the framework with the defined spatial constraints. Our tests show that the proposed framework is capable of producing robust correspondences with higher correspondence ratios and reasonable computational efficiency, compared to other well-known algorithms.**

***Index Terms*— Dual-tree wavelet transform (DTCWT), object matching, pairwise spatial constraints, polar matching, scale-invariant feature transform (SIFT).**

I. Introduction

THE SEARCH for robust and accurate correspondences between images is an important problem in computer vision. Many computer vision and image processing tasks, such as wide baseline matching, object detection, classification and recognition require accurate correspondences to achieve good performance. Thus, designing algorithms that produce more accurate and robust correspondences should lead to systems with better performance.

One common approach to solve the correspondence problem is to consider only local correspondences, using interest points and feature descriptors. A comprehensive comparison of commonly used interest point detectors and descriptors can be found in [1]–[3]. However, considering local feature appearance alone is often insufficient when searching for robust correspondences, due to various challenging factors, such as occlusion and changes in viewpoint and illumination. Other information, such as spatial information, can potentially be used to produce more robust correspondences. For example, groups of matching features should approximately have the same orientation and distance relative to each other between interest points. These additional pieces of information can be used to define spatial constraints on the features that match.

The main aim of this paper is to develop a robust interest point matching framework that considers spatial constraints defined by the relative orientation and change in scale between *pairs* of features, rather than single features. An additional benefit from this framework is that the orientation and scale of individual features do not need to be determined at the interest point detection stage. This differs from conventional methods of finding correspondences [such as scale-invariant feature transform (SIFT) [4]], which typically need the orientation and scale of features to be estimated by interest point detectors. With the proposed pairwise framework, the relative scale change and orientation change are determined only at the interest point matching stage, and our results demonstrate that this framework is capable of producing more robust correspondences.

Our main contributions are as follows.

1) We define a set of pairwise spatial constraints that can be used to select the relative orientation and change in scale between features when searching for correspondences.
2) We develop a robust matching framework that searches for clusters of pairwise correspondences, satisfying the defined spatial constraints, by using a 2-D similarity space. Robust and accurate correspondences can be found as clusters in this similarity space.
3) As a comparison, we study the performance of several common matching algorithms on 3-D objects under the effects of geometric distortion and viewpoint change. These algorithms also make use of spatial information to find correspondences. We demonstrate that our proposed framework compares well with these algorithms.

II. Related Work

Spatial constraints provide important information on the layout of features, which can be used to search for robust correspondences. In this section, we review prior work related to the use of spatial information for matching, which can be broadly classified into two approaches.

A. Graph-Based Approach

Graphs provide a flexible way of representing the features in images, thus image matching can be considered as a

Manuscript received April 20, 2011; revised March 16, 2012; accepted March 19, 2012. Date of publication April 17, 2012; date of current version July 18, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mary Comer.

The authors are with the Department of Engineering, Signal Processing and Communications Laboratory, University of Cambridge, Cambridge CB2 1TN, U.K. (e-mail: esn21@cam.ac.uk; ngk10@cam.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2012.2195012

graph matching problem. Generally, graph matching can be formulated as an assignment problem with certain mapping constraints, and solving it is usually NP-hard [5]–[9]. Thus, there is a need to design efficient algorithms to find approximate solutions for the problem. One such approach is the use of spectral methods. Umeyama [10] proposed an analytic solution based on the eigendecomposition of adjacency matrices to find the permutation matrix. The algorithm requires the graphs to have the same number of nodes, which is not practical in most computer vision applications. Shapiro and Brady [11] proposed an algorithm, which compared eigenvectors obtained from the adjacency matrices of individual images. Correspondences were found by minimizing the Euclidean distance between rows of the modal matrices. Generally, spectral methods are sensitive to outliers, and modal representations alone may not be sufficient to produce robust correspondences when matching complicated objects [12].

Another approach is to formulate the assignment problem as an integer quadratic program (IQP) [5], [6], [8], [13], and approximate solutions can be obtained by solving the optimization problem. In [5], a graduated assignment approach that iteratively refined previous matches using mapping constraints on the permutation matrix was proposed to find partial matches between attributed graphs. Even though, the algorithm produced good results, the algorithm is computationally costly.

In [12], a point matching algorithm that made use of the thin-plate spline for modeling the nonrigid spatial mapping of points was proposed, and softassign was used for correspondences. Similar to [5] and [12], Belongie *et al.* [14] proposed using shape context descriptors to solve for correspondences between different objects as a graph matching problem. The correspondences were then fitted with a thin-plate spline transformation model, and the results were refined based on the derived model. Berg *et al.* [13] minimized a cost function formulated as an IQP for feature similarity and the geometric distortion between candidate correspondences. After solving the optimization problem, a model between the points was estimated and used to refine the correspondences. Torresani *et al.* [15] solved the problem as an energy minimization graph matching problem using a dual decomposition technique.

Since graph matching is generally computationally costly, Leordeanu and Hebert [8] proposed an efficient spectral relaxation method to solve the IQP by finding the best matching clusters in the graphs. The affinities between pairs of points were considered, and the algorithm was shown to produce good approximate solutions efficiently. In [16], a discriminative algorithm was proposed, which uses the technique in [8] for object recognition and localization based on geometric constraints. In [17], the spectral matching technique was extended to include affine constraints along with bistochastic normalization. Improved matching performance was achieved at the tradeoff of increased computational complexity.

B. Geometric Approach

Spatial information can also be used for matching by considering different ways of representing the features' spatial information and local appearance. Generally, these approaches model the spatial relationships between features directly. One common approach is to use spatial information to derive the parameters of a pre-defined geometric model, assumed to represent the relationship between two sets of features. In particular, algorithms that sample the space of candidate correspondences to remove outliers, which do not follow the defined model have been used to produce robust correspondences, such as RANSAC [18]. Gold *et al.* proposed a point matching algorithm [19] based on pose estimation using a geometric affine model for finding correspondences. A cost function, which modeled the affine mapping of points, along with the defined mapping constraints on the matches was solved using an optimization technique. In [20], Lowe extended the basic SIFT matching procedure by fitting an affine model to the matches using a Hough transform and solving for the parameters iteratively with an optimization algorithm. The matches have been shown to be accurate, and these matches are then clustered into different models of a single object from various viewpoints, resulting in an effective object recognition system [4].

Lazebnik *et al.* proposed an object recognition algorithm in [21] based on groups of local affine regions to model 3-D objects. Spin images and a variant of SIFT are used as features to find correspondences between images by identifying sets of three regions that match. Objects are then represented by semi-local affine parts, which are learned using the correspondences found with additional training images. Carneiro and Jepson proposed a pairwise clustering algorithm for finding correspondences using semi-local constraints in [22], along with a semi-local feature based on an extension of the shape context descriptor in [14] for matching. Geometric prediction models are also used to improve the performance of the algorithms. The pairwise clustering algorithm in [22] defines a pairwise similarity score using the distance, orientation, and appearance of feature pairs, which is then collected in an affinity matrix. A connected component analysis is performed on the matrix to find the correspondences. This algorithm has similarities to the spectral matching algorithm in [8] since both algorithms consider the similarity between pairs of features in one image to pairs of features in another, and define a pairwise similarity score between the feature pairs in the affinity matrix. More importantly, both the algorithms in [8] and [22] search for correspondences by finding strongly connected clusters in the affinity matrix. Even though the pairwise similarity scores are defined differently in [8] and [22], the underlying approach of finding strongly connected clusters in the defined affinity matrices is similar. Likewise, our proposed algorithm has similarities to [22], by considering the relative orientation and change in scale between features. However, our approach is different since we perform a search for correspondences in a defined pairwise similarity space, instead of the affinity matrix directly.

C. Our Approach

Generally, geometric approaches assume that correct correspondences follow a pre-defined geometric model, such as the geometric affine model [4], [19], [20]. This assumption

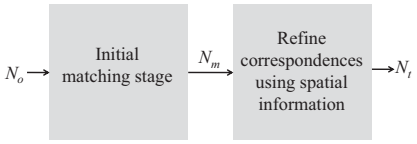


Fig. 1. Generic framework for matching with spatial information. For N_o, the number of interest points to be matched, an initial matching stage produces N_m correspondences. Spatial information, such as the orientation and scale of features, is then used to produce N_t robust correspondences as the output.

restricts the correspondences that can be found, and the pre-defined model may also be an inadequate representation of the relationship between complicated objects under large viewpoint changes. Even though graph matching approaches consider the relationship between features, solving the combinatorial optimization problem is NP-hard, and obtaining an approximate solution is still computationally costly, while not necessarily producing accurate solutions [8], [9]. In addition, the use of invariant features in different matching algorithms implies that they tend to rely on the scale and orientation of features estimated by the feature detector and descriptor.

In this paper, we consider the pairwise relationship between pairs of features, since the distortion between them can generally be modeled as a rotation and scale change under large viewpoint changes. By mapping the pairwise relationship into a similarity space, using a set of pairwise spatial constraints defined on the *relative* orientation and *change* in scale between pairs of features, we are able to find robust correspondences, which satisfy these constraints. Using these constraints, the proposed matching framework does *not* depend on interest point detectors to estimate orientation and scale.

III. Pairwise Matching Using SIFT

In this section, we describe our basic framework for matching with pairwise spatial constraints. For simplicity, this employs a pairwise matching algorithm, which uses the well-known SIFT interest-point detector and descriptor [4], and makes effective use of spatial information between pairs of candidate correspondences to produce good matching performance [23]. In later sections, we extend the pairwise matching ideas to a system based on complex wavelet methods. Note that in the following discussion, a *pair of candidate correspondences* refers to two interest points in one image being matched to two interest points in another image.

A. Framework for Using Spatial Constraints

In general, matching algorithms that make use of spatial information, such as [8] and [20] have two main stages, as shown in Fig. 1. The initial matching stage finds a set of candidate correspondences between individual features of the images based on a similarity score, such as the Euclidean distance, distance ratio threshold [4] or correlation score [24] of feature descriptors. The subsequent stage refines these correspondences using spatial information, such that the output consists of more robust correspondences. This is a good approach for designing these matching algorithms, since the initial matching stage eliminates features that are poor correspondences, such that the number of correspondences

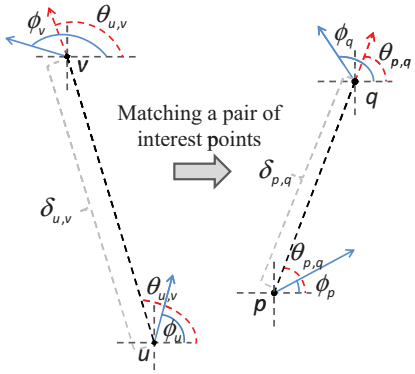


Fig. 2. Matching a pair of interest points u, v to a second pair p, q. $\theta_{u,v}$ is the direction of the vector between u, v, and $\delta_{u,v}$ is the distance between u, v (similarly for $\theta_{p,q}$ and $\delta_{p,q}$). ϕ_u, ϕ_v, ϕ_p, ϕ_q are feature orientations at interest points u, v, p, q.

N_m is small enough to keep the subsequent matching stage computationally efficient, instead of having to consider all N_o features directly. We also define N_t, the total number of output correspondences produced by the subsequent matching stage. We adopt this two-stage approach for all the matching algorithms considered in this paper. Next, we describe an algorithm for this second stage, that can produce robust correspondences by defining spatial constraints on the relative orientations between pairs of candidate correspondences.

B. Spatial Constraints Using Orientation of SIFT Features

Our SIFT-based pairwise matching algorithm, first presented in [23], requires that groups of interest points which are true correspondences to satisfy certain spatial constraints. For example, interest points in image Y should have the same *relative* orientation within image X when they are true correspondences. These constraints are defined between pairs of candidate correspondences, and robust correspondences can be found by searching for clusters in a similarity space.

Consider interest points, u and v, in an image X, as shown in Fig. 2. The line vector $\hat{x}$ between them can be defined as

$$\hat{x}_{u,v} = \delta_{u,v} \exp(j\theta_{u,v}) \tag{1}$$

where $\delta_{u,v}$ is the length and $\theta_{u,v}$ the orientation of $\hat{x}_{u,v}$. For a second pair of interest points, p and q, in another image Y, a second line vector $\hat{y}_{p,q}$ is defined similarly. The pairwise spatial relationship between these line vectors can be defined as the complex log-ratio

$$\begin{aligned} \kappa + j\omega &= \ln \frac{\hat{x}_{u,v}}{\hat{y}_{p,q}} \\ &= \ln \frac{\delta_{u,v} \exp(j\theta_{u,v})}{\delta_{p,q} \exp(j\theta_{p,q})} \\ &= \ln \frac{\delta_{u,v}}{\delta_{p,q}} + j(\theta_{u,v} - \theta_{p,q}) \end{aligned} \tag{2}$$

where ω is the difference in orientation of the vectors (i.e., rotation), and κ is the log-ratio of vector lengths (i.e., scale change). We define a more convenient scale

parameter, $\lambda = \kappa/\ln 2$, which is the ratio of the vector lengths on a $\log_2$ scale. A pairwise similarity space $\mathcal{K}(\omega, \lambda)$ can then be defined for pairs of candidate correspondences in X and Y. For each of these, a pairwise similarity score $\psi_{\{(u,p),(v,q)\}}$, which measures the orientation consistency and feature similarity, is stored in $\mathcal{K}$ at location $\{\omega, \lambda\}$ and is given by

$$\psi_{\{(u,p),(v,q)\}} = \frac{\chi_{u,p}\gamma_{u,p} + \chi_{v,q}\gamma_{v,q}}{2} \tag{3}$$

where $\chi_{u,p}$ is the orientation consistency of u and p, and $\gamma_{u,p}$ is the feature similarity score. These are defined as

$$\chi_{u,p} = \frac{\cos(\phi_u - \theta_{u,v} - \phi_p + \theta_{p,q}) + 1}{2} \tag{4}$$

$$\gamma_{u,p} = \exp\left(-\frac{\|f_u - f_p\|^2}{2\sigma^2}\right) \tag{5}$$

where f_u and f_p are the feature vectors at u and p, respectively, with orientations ϕ_u and ϕ_p. $\chi_{v,q}$ and $\gamma_{v,q}$ can then be defined similarly, with f_v and f_q the feature vectors at interest points v and q, with orientations ϕ_v and ϕ_q.

An illustration of a pair of candidate correspondences is shown in Fig. 2. Note that $\phi_u - \theta_{u,v}$ is the difference between the dominant orientation of the feature f_u and the orientation of the line vector $\hat{x}_{u,v}$. Pairs of true correspondences will give $\psi \approx 1$, since they will satisfy the orientation consistency while also having similar feature appearance. Hence, we accumulate in $\mathcal{K}(\omega, \lambda)$ the ψ values of all pairs of candidate correspondences between X and Y, which have values larger than a threshold τ_0. True correspondences can then be found by searching for modes or regions of high density in $\mathcal{K}(\omega, \lambda)$, since corresponding groups of interest points with the same relative spatial information (and hence from the same object) will tend to be tightly clustered in (ω, λ) space. In [23], this algorithm was shown to produce more robust correspondences compared to [8] using the CALTECH database [25].

Conventionally, the scale and orientation of features used for matching are estimated by the interest point detector. However, it is often challenging to estimate these well, since the exact spatial extent of a feature is usually unknown, and a feature can potentially have several dominant orientations. Previous works on the estimation of scale include the discrete scale space theory developed in [26] and automatic scale selection for feature and edge detection [27], [28]. Others include the search for peaks in 3-D space of spatial location and scale to determine the location and scale of interest points. They usually use the Laplacian-of-Gaussian or other differential filters [29], or the difference-of-Gaussians filter (DoG) [4], [20] to form the metric for interest point detection and scale estimation. Orientation can then be estimated, using the detected scale to define the size of local region around each interest point. In [20] and [4], a histogram of the gradient orientations in each local region is formed, and peaks in the histograms are then assigned as dominant orientations.

In view of the difficulty in achieving reliable and consistent estimates of scale and orientation, we have decided to develop an alternative approach to the algorithm in [23], which is based on the flexibility afforded by the dual-tree wavelet transform (DTCWT). Feature descriptors that are multiscale and multiorientation in nature can thus be efficiently produced, and hence we do not require such parameters from the detector.

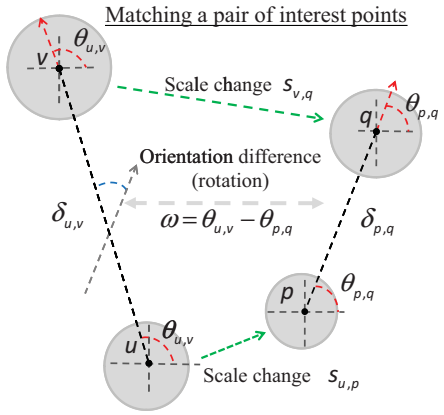


Fig. 3. Matching a pair of features u, v to a second pair p, q by considering the relative orientation and change in scale between them. $\theta_{u,v}$ is the direction of the vector between u, v, and $\delta_{u,v}$ is the distance between u, v (similarly for $\theta_{p,q}$ and $\delta_{p,q}$). The relative orientation between the features is determined as the rotation ω between the line vectors based on the defined spatial constraint. $s_{u,p}$ and $s_{v,q}$ can be approximated as λ, the log ratio of the line vectors' lengths.

IV. Pairwise Matching Using Complex Wavelets

In theory, we do not require the interest point detector to estimate orientation and scale, since we can consider the relative orientation R and change in scale s between features directly during matching. More specifically, based only on the line vectors $\hat{x}_{u,v}$ and $\hat{y}_{p,q}$, we can estimate R and s between pairs of correspondences, $\{u, v\}$ and $\{p, q\}$, and define spatial constraints with them, since groups of matching features are expected to satisfy certain spatial relationships such as having approximately the same relative orientation and change in scale. These constraints can then be used directly to select R and s between clusters of true correspondences. We now extend the formulation of orientation-based pairwise matching from Section III-B to incorporate both scale and orientation into our matching framework, such that it can produce robust correspondences without requiring the orientation and scale to be estimated by the interest point detector.

A. Spatial Constraints Between Pairs of Correspondences

Consider again the pairs of interest points, $\{u, v\}$ and $\{p, q\}$, from Fig. 2. $\chi_{u,p}$ may now be re-defined as

$$\begin{aligned}\chi_{u,p} &= \frac{\cos(\phi_u - \phi_p - (\theta_{u,v} - \theta_{p,q})) + 1}{2} \\ &= \frac{\cos(R_{u,p} - \omega) + 1}{2}\end{aligned} \tag{6}$$

where $R_{u,p}$ is the relative orientation between the features at u and p, and ω is the rotation between the line vectors of the candidate correspondence pair, as defined in (2). Similarly, we may define $\chi_{v,q}$ to depend on $R_{v,q}$ and ω, where $R_{v,q}$ is the relative orientation between the features at v and q. We can then define a spatial constraint on the relative orientations between pairs of candidate correspondences, such that

$$R_{u,p} \approx \omega \qquad R_{v,q} \approx \omega \tag{7}$$

if $\{f_u, f_p\}$ and $\{f_v, f_q\}$ are true correspondences, as shown in Fig. 3. This is a valid constraint, since we expect the relative orientations between pairs of true correspondences to be approximately the same as the difference in orientation of the line vectors joining them. In addition, we can define a second spatial constraint on the change in scale between pairs of candidate correspondences, such that

$$s_{u,p} \approx \log_2 \frac{\delta_{u,v}}{\delta_{p,q}} = \lambda \qquad s_{v,q} \approx \log_2 \frac{\delta_{u,v}}{\delta_{p,q}} = \lambda \tag{8}$$

if $\{f_u, f_p\}$ and $\{f_v, f_q\}$ are true correspondences, and $s_{u,p}$ and $s_{v,q}$ are the changes in scale between them. $\delta_{u,v}$ and $\delta_{p,q}$ are the lengths of the line vectors, and λ is the $\log_2$ version of κ in (2). Equation (8) is a valid constraint since the change in scale between a pair of true correspondences should remain approximately unchanged under the effects of geometric distortions and viewpoint change, as shown in Fig. 3 and should approximately match the change in lengths of the vectors joining the pairs of points if they are from similar rigid objects in the two images. (Note that there will be some pairs of points where viewpoint perspective effects on 3-D objects will invalidate either the scale or rotation constraints to some extent, but it is expected that these will be in the minority.)

Having defined spatial constraints on R and s between pairs of candidate correspondences, the absolute orientation and scale of individual features are no longer required and we can now design a matching framework based on these relative rotation and scale parameters. To include these constraints, the similarity between a pair of features should vary as a function of both R and s. The feature similarity score defined in (5) is modeled as the Gaussian function of the Euclidean distance between features. For a pair of interest points u and p, (5) can be re-defined as

$$\begin{aligned} \gamma_{u,p} &= \exp\left(-\frac{\|f_u - f_p\|^2}{2\sigma^2}\right) \\ &= \exp\left(-\frac{\|f_u\|^2 + \|f_p\|^2 - 2f_u f_p}{2\sigma^2}\right) \\ &= \exp\left(-\frac{1 - f_u f_p}{\sigma^2}\right) \\ &\approx \exp\left(-\frac{1 - \nu_{u,p}(\Theta)}{\sigma^2}\right) \end{aligned} \tag{9}$$

where f_u and f_p are assumed to be l_2-normalized (typically to reduce sensitivity to lighting variations), and $f_u f_p$ can be interpreted as the similarity score $\nu_{u,p}(\Theta)$ between features f_u and f_p based on a set of unknown parameters Θ. Since the spatial constraints are defined on the relative orientation and change in scale between pairs of correspondences, we assume that $\Theta = \{R, s\}$, and (9) can be defined as

$$\gamma_{u,p} \approx \exp\left(-\frac{1 - \nu_{u,p}(R_{u,p}, s_{u,p})}{\sigma^2}\right) \tag{10}$$

where the feature similarity score is now a function that varies with the assumed values of both relative orientation $R_{u,p}$ and the change in scale $s_{u,p}$ between features f_u and f_p. Based on the defined spatial constraints in (7) and (8), $R_{u,p}$ and $s_{u,p}$ can be approximated by ω and λ, respectively. Hence, assuming that $\{f_u, f_p\}$ and $\{f_v, f_q\}$ are true correspondences, we define

$$\begin{aligned} \gamma_{u,p} &= \exp\left(-\frac{1 - \nu_{u,p}(\omega, \lambda)}{\sigma^2}\right) \\ \gamma_{v,q} &= \exp\left(-\frac{1 - \nu_{v,q}(\omega, \lambda)}{\sigma^2}\right). \end{aligned} \tag{11}$$

We observe that SIFT is not very suitable as a feature descriptor for the feature similarity score in (11), since it only retains the most dominant orientation(s) for each descriptor and discards all the other orientations, and similarly for scale. SIFT patches are "de-rotated" and "rescaled" by the estimated orientation and scale before the feature vectors are calculated, so it is not easy to see how the similarity score γ would vary for different assumptions of orientation and scale.

For the proposed matching framework, we require the feature similarity score γ to be a function of R and s, such that the defined spatial constraints can be used directly to determine γ when $R = \omega$ and $s = \lambda$. (Note that ω and λ will vary for a given match pair $\{u, p\}$, according to which other points $\{v, q\}$ are paired with them.) A more suitable choice of feature descriptor here is the polar matching matrix (p-matrix) [24], derived from complex wavelet coefficients. We give a brief overview of this material now.

B. Polar Matching as a Function of Rotation R

In general, wavelet transforms possess many attractive properties, which can be used for object matching. For example, the directional selectivity and invariance to shifts and rotations of Gabor wavelets have produced good performance for face recognition tasks [30]–[32]. Wavelets have also been used previously for object recognition [33]–[35], producing good results in general. However, computational complexity is a concern when wavelet features are used for object recognition, since over-complete wavelet transforms typically become computationally intensive when accounting for different scales and orientations, and this also leads to large wavelet feature vectors. The DTCWT [36], [37] possesses several qualities that are potentially useful for the task of object matching, while addressing the concerns mentioned above. The DTCWT has shift invariance and directional selectivity comparable to the Gabor wavelets, while having significantly lower redundancy and better computational efficiency, as discussed in [32]. The p-matrix, proposed in [24], is a feature descriptor based on DTCWT coefficients, that permits an efficient algorithm, called polar matching, to find correlations between image patches as a function of the angle of rotation between them.

At each level (octave scale), the 2-D DTCWT decomposes an image into six complex directional subbands. By considering also the complex conjugate of these subbands, the coefficients consist of 12 different directions spaced regularly at $(30k-15)^\circ$, for $k = 1, \ldots, 12$. The DTCWT descriptor is formed by assembling the coefficients from 12 points around a ring, together with those from the ring's center point (the interest point), into a p-matrix. The coefficients are arranged such that each 30° rotation of the image patch about the center of the ring corresponds to a cyclical shift by one element of each column of the p-matrix.

If two similar image features have a $n \times 30°$ rotation between them and we consider two p-matrices from equivalent interest points in them, a summation of the column-wise correlations of the two p-matrices will produce a response vector with a peak at a shift by n elements. Thus, the peak correlation score gives an estimate of the relative rotation between the two images ($n \times 30°$). However, the estimated rotation will only be at intervals of 30°. Fortunately the correlation, being cyclic over the columns, can be calculated efficiently in the Fourier domain, and the resolution can be improved by using zero-padding of the Fourier coefficients. This may be used to upsample the original 12 directions to 48 and give a 48-point correlation vector, now with rotation intervals of 7.5°. The rotation may now be estimated to 7.5° resolution or better. More details of the p-matrix can be found in [24]. The amount of information carried by the p-matrix can be increased by adding more columns to the matrix, corresponding to multiple levels and additional rings with the tradeoff being increased computational complexity. In this paper, a sampling ring from a chosen decomposition level of the DTCWT, along with the center points from this level and the next coarser level are used to form the p-matrix, resulting in a 12×8 matrix containing 96 complex coefficients (see typically 128 real coefficients in a SIFT feature vector).

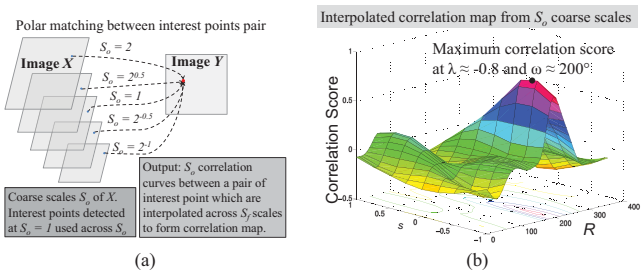
Polar matching efficiently produces a 48-point similarity score, which varies as a function of the relative orientation R between two p-matrices. Based on the spatial constraint in (7), the similarity score between pairs of corresponding p-matrices can be determined as the correlation score when $R \approx \omega$, rounded to 7.5° resolution. Next, we extend the above concepts and introduce a version of polar matching that is tolerant to changes in scale s, such that the correlation score varies as a function of both R and s.

C. Polar Matching as a Function of Scale s

Polar matching may be extended to tolerate changes in scale by considering correlation vectors between p-matrices from different scales (DTCWT levels) of the image pair. Scale increments of less than 2:1 may be achieved by interleaving parallel DTCWT decompositions, starting from different resized versions of the input image. This then leads to a similarity score, which varies smoothly with s as well as R, and is achieved as follows.

Given two images X and Y with M_X and M_Y interest points, respectively, X is sampled at a set of initial coarse scales S_o and the interest points are projected across all $|S_o|$ scales, resulting in M_X p-matrices *at each scale*. Polar matching then results in S_o correlation vectors for each pair of interest points in X and Y (see Section IV-E). Typically $S_o = 5$ and the scale interval is $\sqrt{2}:1$, requiring just two DTCWTs. These correlation vectors can then be interpolated across a fine set of scales S_f, resulting in a correlation map for each pair of features f_u and f_p in X and Y, respectively.

The correlation map is a $48 \times |S_f|$ matrix, which measures the similarity as a function of both the relative orientation R and change in scale s between p-matrices. More importantly, we observe that the correlation map takes the form



(a) (b)

Fig. 4. (a) Extension of polar matching to tolerate change in scale. (b) Example of a correlation map produced for an actual correspondence. Y is a rotated and scaled version of X, with a rotation of 200° and scale of $2^{-0.8}$. We observe that there is a corresponding peak at approximately $R = 200°$ and $s = -0.8$.

of $v_{u,p}(\omega, \lambda)$ and $v_{v,q}(\omega, \lambda)$ in (11), which can be used to determine the similarity of the p-matrices of points $\{u, v\}$ to those of $\{p, q\}$ at $R = \omega$ and $s = \lambda$, determined from the line vectors $\hat{x}_{u,v}$ and $\hat{y}_{p,q}$. An illustration of the interpolation is shown in Fig. 4. Y is only sampled at one scale, since we are considering s to be in the range of 0.5 and 2, and this can be produced by resampling X. However, a wider range of scales can possibly be considered by resampling Y.

The pairwise similarity score $\psi_{\{(u,p),(v,q)\}}$ in (3) can then be calculated as a function of R and s between pairs of correspondences. Following the spatial constraint in (7), the pairwise similarity score can be simplified to be:

$$\psi_{\{(u,p),(v,q)\}} = \frac{\gamma_{u,p} + \gamma_{v,q}}{2} \tag{12}$$

since the sampling of the correlation score at $\phi_u - \phi_p = \phi_v - \phi_q = \theta_{u,v} - \theta_{p,q}$ (i.e., $R_{u,p} = R_{v,q} = \omega$) ensures that $\chi_{u,p} = \chi_{v,q} = 1$ in (3). The ψ values for pairs of candidate correspondences are accumulated in $K(\omega, \lambda)$ and robust "object" correspondences can then be found by searching for clusters in K. The proposed framework does not rely on estimates of the orientation and scale of features.

D. Summary of Matching Framework With Spatial Constraints

A summary of the proposed matching framework can be found in Fig. 5. Similar to our earlier algorithm described in [23], local interest point groups are being considered for matching such that spatial constraints will be considered over a local neighborhood. To form these groups, we only consider pairs of interest points with a distance δ below a threshold τ_δ, such that the pairs of correspondences considered in our framework are all within a local neighborhood.

To find robust correspondences between two images X and Y, polar matching is performed across all S_o scales. Interest point pairs with maximum correlation scores larger than a threshold τ_c are considered as candidate correspondences. We consider $S_o = 2^{(-1,-0.5,0,0.5,1)}$, such that the sampled scales are logarithmically uniform. Local interest-point groups are then formed from the candidate correspondences. For each candidate correspondence, the interpolated correlation map $v_{u,p}(R, s)$ is formed. The scales are interpolated to $S_f = 2^{(-1,-0.75,...,0.75,1)}$ using bicubic interpolation, such that the correlation map is a 48×9 matrix, where R is the relative

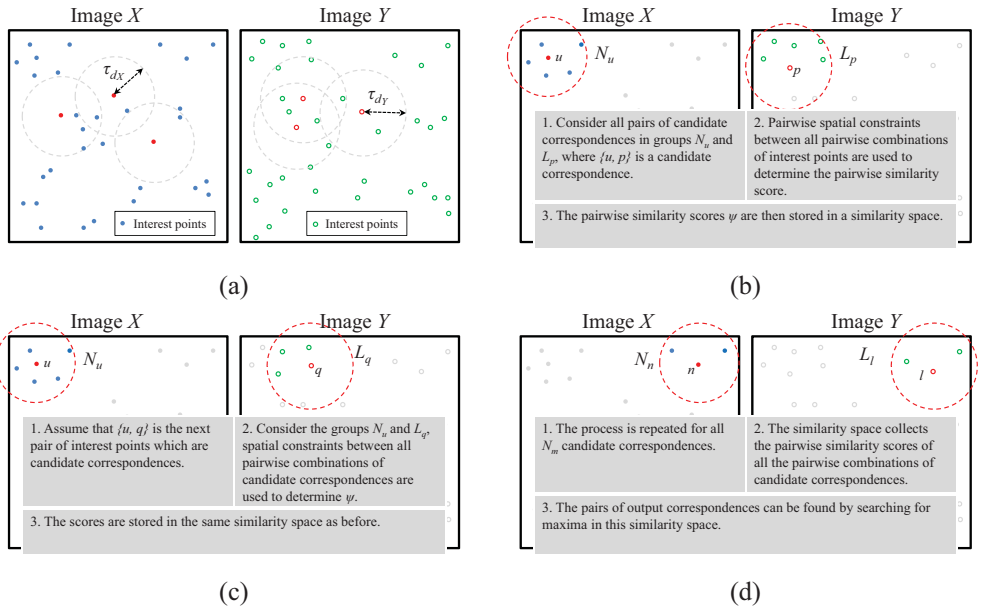


Fig. 5. Summary of the matching algorithm. (a) Two images X and Y, containing interest points with N_m candidate correspondences. (b) Matching a group N_u in X to a group L_p in Y. (c) Collect pairwise similarity scores for all pairs of correspondences in N_u and L_q, where $\{u, q\}$ is also a candidate correspondence. (d) Repeat the process for all N_m candidate correspondences using the groups formed to obtain all the pairs of correspondences, where N_n and L_l represent arbitrary groups in X and Y, respectively, which are candidate correspondences.

orientation and s is the change in scale between two features f_u and f_p.

Consider N_m candidate correspondences produced by the extended polar matching technique, which consist of N interest points in X and L interest points in Y, interest point groups are formed by iterating through the candidate correspondences. For each candidate correspondence $\{u, p\}$, where u and p are interest points in X and Y, respectively, interest point groups can then be formed

$$\begin{aligned} &\text{if} \quad \delta_{n,u} < \tau_{\delta_X}, \qquad n = 1, \ldots, N \rightarrow n \cup N_u \\ &\text{if} \quad \delta_{l,p} < \tau_{\delta_Y}, \qquad l = 1, \ldots, L \rightarrow l \cup L_p \end{aligned} \tag{13}$$

where N_u is the group defined for u containing all the interest points in X with a pairwise distance $\delta_{n,u}$ between n and u below a threshold of τ_{d_X}. Similarly, L_p, l and τ_{d_Y} are defined for the interest point p in Y as well. Note that n and l are interest points in the set of candidate correspondences. Thus, the number of groups formed depends on the number of unique interest points present in the candidate correspondences (with a maximum of $2N_m$ groups).

The proposed matching framework then considers the pairwise spatial constraints between the interest points in N_u and L_p, and the pairwise similarity scores for all possible pairwise combinations (which are also candidate correspondences) are collected as votes. A pairwise similarity space $\mathcal{K}(\omega, \lambda)$, as defined in Section III, is used to accumulate these votes ψ (12). This is then repeated for all N_m candidate correspondences, with the pairwise similarity scores for all pairwise combinations of candidate correspondences in the respective groups being collected in $\mathcal{K}(\omega, \lambda)$. The pairs of apparently valid correspondences are then found by searching for maxima in the $\mathcal{K}$-space, which are larger than τ_p. Note that $\mathcal{K}$ is a 2-D space, which collects the pairwise similarity scores of all the candidate correspondences based on the pairwise spatial constraints defined over a local region. Thus, searching for clusters in $\mathcal{K}$ is equivalent to finding pairs of correspondences, which satisfy similar spatial constraints.

Here, $\mathcal{K}$ is quantized into bins and we can use either a mean-shift mode estimator [38] or a histogram-based method to find the maxima. ψ is calculated from the correlation map $\nu(\omega, \lambda)$ [using (11) and (12)]. In this paper, we use a histogram-based method and search for the maxima of a smoothed histogram containing the votes in $\mathcal{K}$.

E. Implementation Details for Polar Matching

In this section, we elaborate on several implementation details of the extended polar matching technique of Section IV-C, which affect the performance and efficiency of the proposed matching framework. To form the correlation map $\nu(R, s)$, we have assumed that the same set of interest points will be detected across the scales S_o. This assumption can result in errors when matching objects of different scales. Since small objects tend to have fewer interest points, using the same number of interest points from a fine scale may result in more false matches. Quantization errors will also be introduced since the image is downsampled, and features detected at the coarse scale might be less distinct or informative.

To address this, we select the scales S_o as $2^{(0,0,0,0.5,1)}$, instead of $2^{(-1,-0.5,0,0.5,1)}$, such that the p-matrices used are from the original scale, $S_o = 1$, instead of $S_o = 2^{-1}$ or $2^{-0.5}$, so as to prevent information loss due to subsampling errors. We can then scale the other image with $S_t = 2^{(1,0.5,0,0,0)}$ such that we follow the original design of having 5 scale changes from 0.5 to 2 at intervals of $2^{0.5}$, accounting for multiscale samples for both X and Y.

Unlike SIFT descriptors, which are formed based on an estimated scale obtained by searching for extrema in scale-space, our multiscale design adopts a different approach by considering a range of scales for both X and Y. The p-matrices considered are all formed using the third and fourth levels of the DTCWT of X and Y for S_o and S_t, respectively. Improved tolerance to changes in scale could potentially be achieved by considering p-matrices formed using different levels of the DTCWT. However, we have found this not to be necessary since we are using the spatial constraints in (8) to select the appropriate change in scale between features over the range of 0.5–2. Also, considering p-matrices formed using different levels of the DTCWT decomposition starts to become computationally costly, since we have to consider multiple correlation maps ν for the p-matrices. In practice, we found that the choice of p-matrices formed using third and fourth levels produced good results experimentally.

The computational efficiency of the proposed framework can be improved by choosing the appropriate levels of the DTCWT to form the p-matrices at scales S_o and S_t for the image pair X and Y. More specifically, we only require 2 DTCWT decompositions per image, one at the original scale, the other at a scale factor of $2^{0.5}$. The p-matrices at scales S_o of X can then be obtained by selecting the levels as: $l_X = 3$ for $S_o = 1$ and $l_X = 2$ for $S_o = 2$ from the DTCWT of X, $l_X = 3$ for $S_o = 2^{0.5}$ from the DTCWT of scaled X (by a factor of $\sqrt{2}$), where l_X is the level of the DTCWT, $l_X = 1$ being the finest. Similarly for Y, the p-matrices at scales S_t can be obtained by selecting the levels as: $l_Y = 3$ for $S_t = 1$ and $l_Y = 2$ for $S_t = 2$ from the DTCWT of Y and $l_Y = 3$ for $S_t = 2^{0.5}$ from the DTCWT of scaled Y (by a factor of $\sqrt{2}$), where l_Y is the level of the DTCWT of Y, $l_Y = 1$ being the finest.

We also consider that interest points detected at different scales might differ significantly, and using the same set of interest points across S_o can also lead to poor estimates of candidate correspondences between images. This is especially the case when there is a large difference in the number of interest points. To address this issue, we collect more interest points when one image has significantly fewer interest points than the other. In our tests, when $M_0 < N_0/2$ (M_0 and N_0 are the numbers of interest points in images Y and X), we collect more interest points by upsampling Y by 2. We typically consider X as the "reference" image and Y as the "test" image. Thus, the resampling only applies when the "test" image produces fewer interest points than half the number produced by the "reference" image. Since we are matching single objects with distinctive features across different viewpoints in our tests, we only consider the resampling stage when the scale of the test image is smaller than or equal to that of the reference image.

We find that the above changes to the extended polar matching technique produce correspondences that are more tolerant to changes in scale, thus resulting in better matching performance. By choosing an appropriate initial threshold τ_c for polar matching to select candidate correspondences and an appropriate distance threshold τ_δ to form interest point groups, we can ensure that the proposed framework produces a reasonably large number of correct correspondences, while being computationally comparable to the other algorithms. From the calibration tests in Section V-A, we found that a choice of $\tau_c = 0.65$ and $\tau_\delta = 0.1$ times the maximum dimension of the images produced good results.

F. Discussion

Before presenting the experimental results, we highlight several important points regarding the algorithms in Sections III-B and IV. First, we emphasize that both algorithms have the same pairwise similarity scores ψ, as defined in (3) and (12). In both cases, ψ is the mean of the feature similarity score γ weighted by the orientation consistency χ. Note that ψ is not constrained directly by the scale change between the features, which could be accounted for by defining an additional scale consistency measure. The difference in (3) and (12) is that in (12), ψ varies as a function of scale change and relative orientation between features because of the feature similarity score γ defined in (11). In (3), ψ is defined for scale and rotation-invariant features such as SIFT, thus γ is calculated directly as the Euclidean distance between the features, without the need to consider its variation with scale and orientation. We do not require the scale and orientation of individual features in (12), since we estimate these when we match a pair of features in one image to a pair in another using ω and λ in (2), which depends only on the line vectors between the pairs of features. Note that ψ is the same for the algorithms in Sections III-B and IV, and thus it is fair to compare their performance, as discussed in Section V.

Second, we emphasize that the proposed algorithm does not rely on the feature detector to provide estimates of the scale and orientation of features. This is because unlike SIFT, which forms the descriptor at the estimated scale such that it is nominally scale-invariant, the extended polar matching technique does not require the feature detector to estimate a scale for the descriptor. Instead, we consider the feature over a range of sampled scales. We note that an appropriate feature similarity score can be obtained for the proposed algorithm by using ω and λ to estimate the relative orientation and change in scale between features, calculated according to the line vectors between the feature pairs.

Last, we note that to obtain the set of candidate correspondences using polar matching, the orientation and scale of individual features are also not required. Candidate correspondences are obtained by selecting the pairs of interest points with correlation maps having a peak larger than τ_c. The correlation maps for the candidate correspondences are then used to select an appropriate correlation score in (11) based on ω and λ in the proposed pairwise algorithm.

To conclude our discussion, we highlight how the proposed algorithm differs from other algorithms that consider the pairwise relationships between features, such as [8] and [22]. In [22], the authors defined a pairwise similarity score based on the change in scale, change in distance, and change in heading between features, which is similar to the defined spatial constraints in (4), (7), and (8). References [8] and [22] also both considered the pairwise relationships between

pairs of features, and searched for correspondences by finding strongly connected clusters in the defined affinity matrix.

Despite the similarities, there are several significant differences between our work and [8] and [22]. First, the pairwise similarity score ψ in Section III is defined differently. Our work considers the soft consistency measure χ in (4) defined using the difference in feature orientation and rotation of the line vectors between pairs of correspondences, which is used as weights to γ in (3). This is different from [22], which formulated the pairwise similarity score as a Gaussian function of the defined semi-local spatial relationships. Second, the proposed algorithm in section IV formulates a pairwise similarity score ψ that is *independent* of the scale and orientation of individual features. Instead, it is calculated based on the rotation and length-ratio of the line vectors between pairs of features, using (7) and (8). This is different from [22], which considered the orientation and scale of features provided by feature detectors and descriptors in the pairwise similarity score. Third, the proposed algorithm searches for correspondences in a similarity space with ω and λ in (2) as its dimensions. This is different from both [8], [22], which used graph-based approaches to find strongly connected clusters in the affinity matrix. For these reasons, we only compared the proposed algorithm with [8] by using the defined pairwise similarity score in (3) to form the affinity matrix, since [8], [22] have similar approaches to finding correspondences in the affinity matrix. By using the same pairwise similarity score ψ in (3), we try to ensure a uniform comparison between the algorithms. We also compare the proposed algorithm with the Hough transform algorithm in [20] and [4].

V. Experimental Results

We compared the performance of our proposed algorithm with four other algorithms.

1) The basic unconstrained SIFT matching algorithm from [4] (*uc-sift*), which uses the nearest-neighbor distance ratio threshold τ_r as a baseline algorithm. In our experiments, $\tau_r = 0.8$ such that a large number of candidate correspondences were considered.
2) The spectral technique in [8] (*sp-sift*) using SIFT features with bistochastic normalization [17]. The candidate correspondences were selected with $\tau_r = 0.8$ and pairwise affinities defined as (3).
3) An algorithm based on the proposed algorithm in [4] and [20] (*hough-sift*). This algorithm refines the matches produced by *uc-sift* using a Hough transform to find the parameters of an affine transform between the candidate correspondences. Here, $\tau_r = 0.8$.
4) Our earlier pairwise algorithm from [23] (*pw-sift*), which uses SIFT features for all matching and the distance-based technique in Section IV-D to form interest point groups. In our experiments, $\tau_r = 0.8$, and $\tau_\delta = 0.1$. The scale factor σ in the feature similarity score (5) was set to 0.75. We accepted votes in $\mathcal{K}(\omega, \lambda)$ with pairwise similarity score (3) larger than a threshold $\tau_0 = 0.8$.

We define the proposed extended polar matching technique as *pmat*, and the pairwise framework based on this as *pw-pmat*.

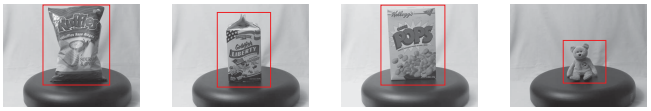


Fig. 6. Examples of test objects and the regions selected for interest points.

When we include the resampling process, which is described in the second-last paragraph of Section IV-E and is used to overcome problems of small scale in the test image, we denote the framework as *pw-pmatsc*. For the SIFT-based algorithms, interest points were detected using the standard DoG detector from [4]. To ensure a fair comparison, the same sets of interest points produced by this DoG detector were also used for the proposed framework, without using their scale and orientation. To compare the algorithms' performance, we calculate the correspondence ratio r_c (*inlier ratio*), defined as

$$r_c = \frac{N_c}{N_t} \tag{14}$$

where N_c is the number of correct or true correspondences (*inliers*) and N_t the total number of output correspondences. r_c is an appropriate performance measure since we are considering the improvements that these algorithms can bring to a set of candidate correspondences, and it has been used to measure the quality of matching algorithms, most recently in [39]–[41]. We also consider N_c, since large r_c and N_c imply that the algorithm is capable of producing a reasonably large number of robust correspondences while removing false correspondences (*outliers*) effectively.

We selected 30 objects from the database in [25], which can be found at http://www-sigproc.eng.cam.ac.uk/∼esn21 The database contains images of different objects that were taken as they were rotated on a turntable at intervals of 5° and each image is 1024 × 768 pixels. We have selected objects in the database that have distinctive features since we are using interest points and descriptors to find correspondences. In particular, we left out objects with near-spherical surfaces and specular reflections. Since the same objects have been used for all the tests, we believe this to be a fair comparison of the algorithms. Interest points were detected from a rectangular region selected by hand around each object, such that only features from the objects are being considered for matching. Some of the objects are shown in Fig. 6.

We adopted an evaluation framework similar to that proposed in [25] using epipolar constraints of the calibrated stereo rig, and we calculated the correspondence ratio when test views are matched to a selected reference view of each object. Note that the test views of each object considered have viewpoint changes of −45° to 45° at intervals of 5°, relative to the reference view of the object.

A. Calibration Tests

First, we performed calibration tests to select parameters for *pw-pmat*. We tested the selected 30 objects with reference views at the viewpoint of 0°. Test views were taken at ±45°, ±30°, and ±15°, at the same scale as the reference view (i.e., no change in scale). To select an appropriate threshold

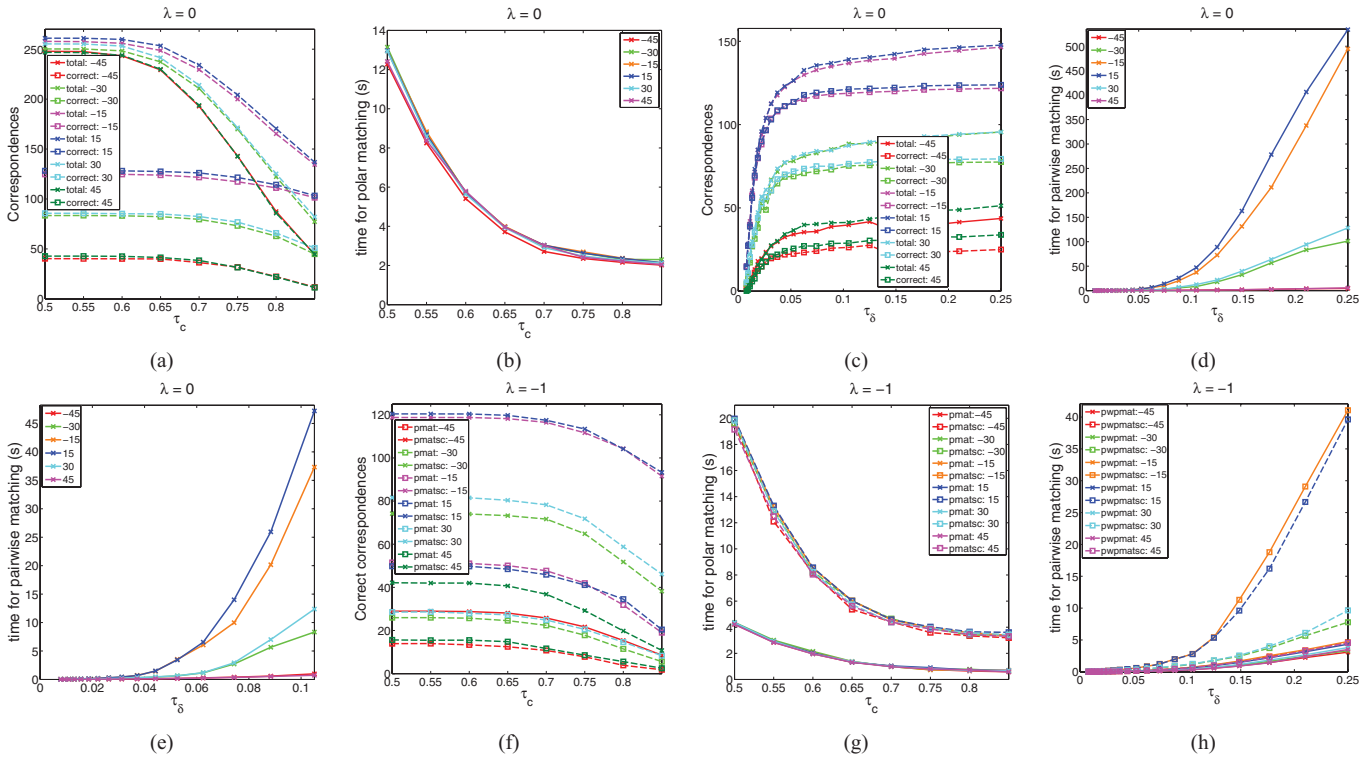


Fig. 7. Calibration tests using 30 images (objects) at each viewpoint. τ_c, the threshold for accepting correspondences produced by polar matching, is varied from 0.5 to 0.85 at intervals of 0.05. τ_δ, the distance threshold for interest point groups for pairwise matching, is varied from 2^{-7} to 2^{-2} at $\log_2$ intervals of 0.25. (a) Average number of correspondences produced by polar matching for $\lambda = 0$ per image. (b) Average computation time of polar matching for $\lambda = 0$ per image. (c) Average number of correspondences produced by pairwise matching for $\lambda = 0$ per image. (d) Average computation time of pairwise matching for $\lambda = 0$ per image. (e) Average computation time of pairwise matching for $\lambda = 0$ for small τ_δ per image [zoomed in version of (d)]. (f) Average number of correspondences produced by polar matching per image, with and without resampling, for $\lambda = -1$. (g) Average computation time of polar matching per image, with and without resampling, for $\lambda = -1$. (h) Average computation time of pairwise matching per image, using candidate correspondences produced by both versions of polar matching, for $\lambda = -1$.

τ_c for the maximum correlation score of *pmat*, we varied τ_c from 0.5 to 0.85 at intervals of 0.05, and observed the N_c produced. As shown in Fig. 7(a), we observe that N_c remained approximately constant as τ_c is increased initially, while above $\tau_c \approx 0.65$, the number of candidate correspondences tends to decrease. We also observe in Fig. 7(b) that the average computation time for *pmat* becomes approximately constant when $\tau_c \geq 0.65$. Thus, a suitable choice of τ_c will be 0.65 for selecting candidate correspondences *pmat*.

Next, we calibrated *pw-pmat* by varying τ_δ in the range of $2^{-7,-6.75,\ldots,-2}$, using candidate correspondences selected with $\tau_c = 0.65$. The scale factor σ in (10), was set empirically to 0.85, and we accepted votes in $K(\omega, \lambda)$ with ψ larger than a threshold $\tau_p = 0.7$. As shown in Fig. 7(c), N_c varies with τ_δ, and τ_δ was selected such that approximately 50% of correct correspondences in N_m were retained during calibration. The other algorithms are calibrated to produce the same number of correct correspondences approximately as *pw-pmat*. For *pw-pmat*, we observe that $\tau_\delta \approx 0.1$ is an appropriate selection, since the computation time increases significantly above this.

We also observe in Fig. 7(d) that the computation time for the pairwise matching stage increases with τ_δ. In particular, the computation time increases rapidly when $\tau_\delta \geq 0.1$. From Fig. 7(b) and (e), we observe that the computation time for the pairwise matching stage is comparable with

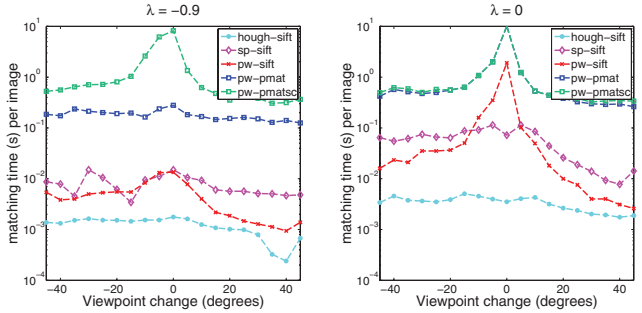


Fig. 8. Example of computation times for matching algorithms in $\log_{10}$ scale. The computation times include both the initial single feature matching stage and the subsequent matching stage using spatial information.

the polar matching stage for $\tau_c = 0.65$ and $\tau_\delta = 0.1$ when the viewpoint change is large. To investigate the differences between using *pmatsc* and *pmat* for selecting candidate correspondences, we consider $\lambda = -1$, since the resampling stage in *pmatsc* was designed specifically to account for large changes in scale between the features to be matched. In Fig. 7(f), we compare N_c produced by both polar matching techniques as τ_c is varied. We observe that *pmatsc* generally increases N_c. However, the increased N_c has a tradeoff of increased computation time, as shown in Fig. 7(g). This is the case for *pw-pmatsc* as well, as shown in Fig. 7(h), where it has longer computation times compared to *pw-pmat*.

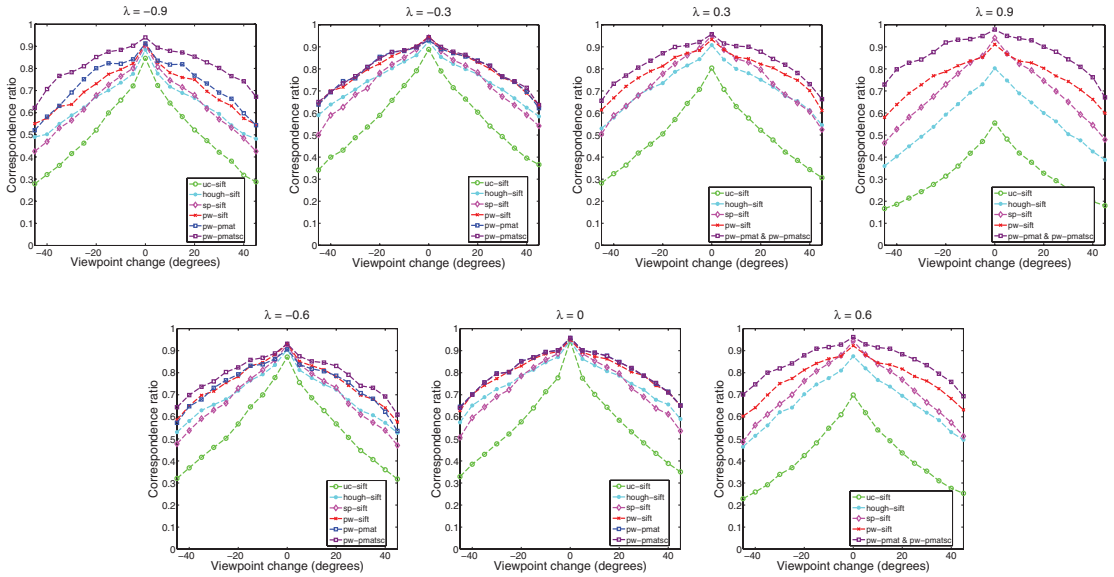


Fig. 9. Correspondence ratio of algorithms for viewpoint change of −45° to 45°. In general, *pw-pmat* performs better across all scales, producing higher correspondence ratios than the other algorithms.

B. Viewpoint Change

Next, we tested the effects of viewpoint change on the correspondences produced. The test views of each object considered have viewpoint changes of −45° to 45° at intervals of 5°, relative to the reference view of the object. Each test view was scaled in $\log_2$ intervals of 0.3 from −0.9 to 0.9, and the test was repeated 3 times for each scaled image, with the reference view at viewpoints of −30°, 0°, and 30°. The average correspondence ratios r_c of the algorithms at each scale interval across all viewpoints are compared in Fig. 9. From the results, we observe that the pairwise methods (*pw-sift*, *pw-pmat*, *pw-pmatsc*) generally perform better across the tested scales and viewpoints, which imply that the use of pairwise spatial constraints with our approach can produce more robust correspondences by removing correspondences that do not satisfy the defined constraints. When the change in scale λ is small, *pw-pmat*, *pw-sift*, and *hough-sift* have similar r_c, but when λ is large, the performance of *hough-sift* decreases, while *sp-sift* improves. *pw-pmatsc* performs better than *pw-pmat* when λ is small and it has a more consistent r_c across the test scales compared to *pw-pmat*, which indicates that the proposed improvements in Section IV-E contribute to a more consistent performance across changes in scale.

More importantly, we observe that *pw-pmat* and *pw-pmatsc* produced large r_c across all test scales, and the r_c decreased more gradually than the other algorithms when the viewpoint angle is increased. This indicates that the defined spatial constraints result in a matching framework that is tolerant of distortions caused by changes in scale and viewpoint of an object. We also compared the number of correct correspondences N_c produced by the algorithms in Fig. 10. Generally, *pw-pmat* and *pw-pmatsc* produce more N_c compared to the other algorithms, which show that the r_c observed has not been skewed by small values of N_c and N_t. This suggests that the proposed pairwise matching technique is capable of producing a good number of inliers, while removing the outliers using the defined spatial constraints more effectively than the other matching algorithms. *pw-pmatsc* produced more N_c than *pw-pmat* due to the resampling stage, and we also observe that N_c remains approximately consistent across scales for *pw-pmatsc*.

In general, we observe that the curves for the tested algorithms are approximately symmetric about 0° viewpoint change, which is expected due to the approximate symmetry of the objects. *pw-pmatsc* and *pw-pmat* also produce similar correspondence ratios for uniform changes in the $\log_2$ scale, with $\lambda = -0.9, -0.6, -0.3$ having approximately the same correspondence ratios as $\lambda = 0.9, 0.6, 0.3$, respectively.

Based on our experiments, we have shown that the proposed matching framework can produce more robust correspondences than algorithms that rely on the orientation and scale estimated by interest point detectors. Note that the improved performance of *pw-pmat* and *pw-pmatsc* comes with a tradeoff of increased computation time as shown in Fig. 8. The computation time considered here is the total time taken for both the initial feature matching stage and the subsequent matching stage using spatial information for all the matching algorithms, as shown in Fig. 1. Typically, these stages form the early stages of various image processing and computer vision applications, such as object detection algorithms. We also observe in Fig. 8 that *pw-pmatsc* has longer computation times than *pw-pmat* due to the resampling stage. Thus, *pw-pmat* may be more appropriate for most image processing and computer vision applications, since *pw-pmat* and *pw-pmatsc* have similar r_c for most test scales.

Note that at small viewpoint changes (±5°), the pairwise algorithms, *pw-pmat*, *pw-pmatsc*, and *pw-sift*, have longer computation times than other algorithms since the algorithms are considering the same image and they need to consider a large number of pairwise combinations of interest points. However, this is not a situation which will typically arise in complicated image processing and computer vision

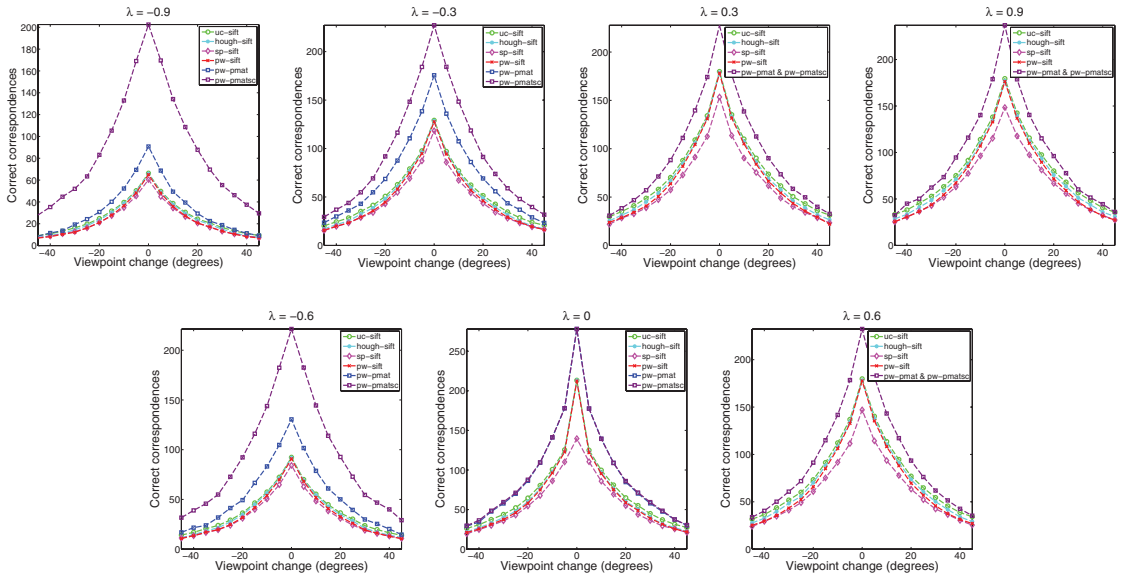


Fig. 10. Average number of correct correspondences per test view for viewpoint change of $-45°$ to $45°$. In general, *pw-pmat* produces more correct correspondences as compared to the other algorithms. *pw-pmatsc* produces the largest number of correct correspondences due to the additional resampling stage.

applications, since we expect distortions to be present between the test and reference images in practice. Thus, it is more important to compare the algorithms' computation times at larger viewpoint changes. In general, the choices of $\tau_c = 0.65$ and $\tau_\delta = 0.1$ for *pw-pmat* produced good results experimentally, along with a reasonable increase in computation times. The increase in computation time is likely to be justified, considering the improved matching performance of the proposed algorithm.

C. k-nn Tests

Last, we conducted tests for viewpoint change using alternative ways of selecting candidate correspondences in the first stage of Fig. 1. Here, we considered the case of selecting up to five nearest neighbors (5 nn) as candidate correspondences, with feature similarity scores, which satisfy a certain threshold, for each feature in the reference view.

More specifically, if there are k candidate correspondences, which satisfy the threshold τ_c for each feature in the reference view, we select the five best correspondences if $k > 5$. Alternatively, we select k best correspondences if $k \leq 5$. This follows the approach adopted in [2] where various ways of selecting correspondences were considered for the evaluation of feature descriptors, including the use of nearest-neighbor distance ratio, the nearest-neighbor, and distance threshold. Our approach of selecting 5 nn, which satisfy a certain threshold is similar to the use of a distance threshold, and it achieves a good balance between increasing the number of correct correspondences considered, while maintaining reasonable computation times. Since we are using the same method of selecting candidate correspondences for the algorithms, we can still ensure a fair comparison of the algorithms without considering all the candidate correspondences selected with a given threshold. Generally, considering all candidate correspondences, which satisfy the threshold is computationally expensive for the matching algorithms.

For the SIFT-based algorithms, a calibration process similar to the one in Section V-A is performed, and we set the distance threshold such that the Euclidean distance between candidate correspondences is always less than 0.5. For *pw-pmatsc* and *pw-pmat*, we set $\tau_c = 0.65$ to select the candidate correspondences. The remaining parameters for the algorithms remain the same as defined earlier in Section V-B, along with the same experimental setup.

This test increases the number of false correspondences and correct correspondences present in N_m, and good matching algorithms are expected to remove the increased number of false correspondences, while maintaining a high r_c and N_c. By comparing the algorithms' performance with larger N_m, we can reinforce our observations and show which algorithms are capable of producing robust correspondences. Due to space limitations, we only show results for a limited range of λ.

We observe in Fig. 11 that *pw-pmat* and *pw-pmatsc* have higher r_c compared to the other algorithms across the test scales, which indicate that the defined spatial constraints can remove false correspondences effectively, while retaining the correct correspondences. The SIFT-based algorithms tend to perform poorly, which suggests that the use of spatial constraints to select the change in scale and relative orientation may be a better approach for matching. *pw-sift* performs better than *hough-sift* and *sp-sift* for small λ, however, as λ is increased, *sp-sift* performs better. *hough-sift* performs poorly at large λ, which suggests that the algorithm may not be effective at removing outliers when more outliers are present. In contrast, *sp-sift* performs better with higher r_c, which suggests that the algorithm is more effective at removing outliers. We also observe that *pw-pmat* and *pw-pmatsc* produced similar results to Section V-B for λ, which indicates that they can perform consistently even with more candidate correspondences.

Furthermore, we observe from Fig. 12 that *pw-pmat* and *pw-pmatsc* generally produced more N_c compared to the other

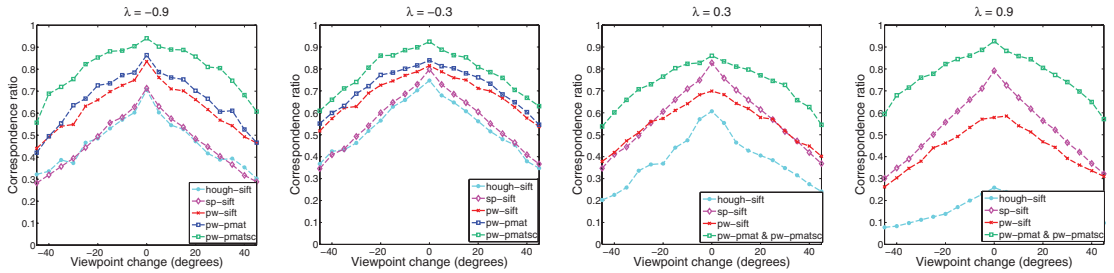


Fig. 11. Correspondence ratio of algorithms for viewpoint change of $-45°$ to $45°$ using 5 nn. Generally, *pw-pmat* and *pw-pmatsc* perform better across all scales at large viewpoint changes, producing higher correspondence ratios than the other algorithms.

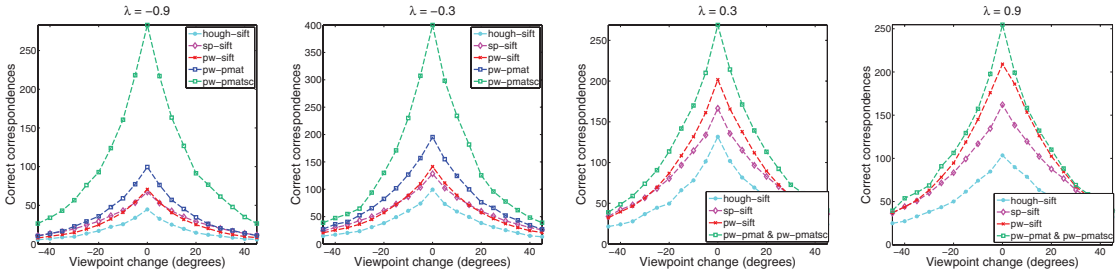


Fig. 12. Average number of correct correspondences per test view for viewpoint change of $-45°$ to $45°$ using 5 nn. In general, *pw-pmat* produces more correct correspondences as compared to the other algorithms. This suggests that the proposed framework is able to remove outliers effectively, while retaining a large number of correct correspondences without having to rely on the estimated orientation and scale of interest points.

algorithms. *pw-sift* also produced more N_c than *hough-sift* and *sp-sift*, with *hough-sift* having the smallest N_c. Thus, even though *pw-sift* produced lower r_c for large λ, it produced more N_c than *sp-sift*. Generally, our results show that when more candidate correspondences are being considered, the proposed pairwise matching framework has better matching performance compared to the other algorithms, which have significant differences in performance compared to Fig. 9.

VI. Conclusion

Matching features based on local appearance alone is often insufficient to produce robust and accurate correspondences under different conditions, such as geometric distortions or viewpoint changes. The use of additional information, such as the orientation and scale of features, can result in better correspondences. In this paper, we developed a pairwise matching framework that defines spatial constraints on the relative orientation and change in scale between pairs of correspondences, such that robust correspondences can be found by searching for clusters in a 2-D pairwise similarity space. The proposed framework does not depend on orientation and scale of individual features estimated by the interest point detector, thus avoiding any undesirable fluctuations in matching performance due to poor orientation and scale estimation. This additional benefit of the framework results from the defined pairwise spatial constraints. Features based on DTCWT coefficients (p-matrices) and polar matching were used such that the feature similarity score between candidate correspondences was calculated efficiently as a function of both relative orientation and change in scale. Thus, the pairwise similarity score can be determined based on the defined spatial constraints on the relative orientation and change in scale between pairs of actual correspondences. Our tests have shown that the proposed framework performs better than a number of other matching algorithms under viewpoint changes. This improvement can be attributed to both the spatial constraints used and the search for clusters in the similarity space. The proposed framework also provides an alternative to relying on the estimated orientation and scale of a feature during feature detection. With better correspondences, the performance of subsequent computer vision and image processing tasks will improve as well. As an extension to the proposed algorithm, future work could involve testing the matching algorithm in cluttered scenes and to include the pairwise similarity score in a classification system.

References

[1] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, nos. 1–2, pp. 43–72, Nov. 2005.

[2] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[3] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 151–172, Jun. 2000.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[5] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 377–388, Apr. 1996.

[6] J. Maciel and J. Costeira, "A global solution to sparse correspondence problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 187–199, Feb. 2002.

[7] H. Bunke, "Error correcting graph matching: On the influence of the underlying cost function," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 917–922, Sep. 1999.

[8] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 1482–1489.

[9] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, no. 3, pp. 265–298, 2004.
[10] S. Umeyama, "An eigendecomposition approach to weighted graph matching problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 5, pp. 695–703, Sep. 1988.
[11] S. L. Shapiro and M. J. Brady, "Feature-based correspondence: An eigenvector approach," *Image Vis. Comput.*, vol. 10, no. 5, pp. 283–288, Jun. 1992.
[12] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Understand.*, vol. 89, nos. 2–3, pp. 114–141, Feb. 2003.
[13] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. 2005, pp. 26–33.
[14] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Apr. 2002.
[15] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimisation," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 596–609.
[16] M. Leordeanu, M. Hebert, and R. Sukthankar, "Beyond local appearance: Category recognition from pairwise interactions of simple features," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[17] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1–8.
[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[19] S. Gold, A. Rangarajan, C. P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, Aug. 1998.
[20] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 1999, pp. 1150–1157.
[21] S. Lazebnik, C. Schmid, and J. Ponce, "Semi-local affine parts for object recognition," in *Proc. Brit. Mach. Vis. Conf.*, vol. 2. 2004, pp. 959–968.
[22] G. Carneiro and A. D. Jepson, "Flexible spatial configuration of local image features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2089–2104, Dec. 2007.
[23] E. S. Ng and N. G. Kingsbury, "Matching of interest point groups with pairwise spatial constraints," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 2693–2696.
[24] N. Kingsbury, "Rotation-invariant local feature matching with complex wavelets," in *Proc. Eur. Conf. Signal Process.*, Sep. 2006, pp. 1–5.
[25] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," *Int. J. Comput. Vis.*, vol. 73, no. 3, pp. 263–284, Jul. 2007.
[26] T. Lindeberg, "Scale-space for discrete signals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 234–254, Mar. 1990.
[27] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *Int. J. Comput. Vis.*, vol. 30, no. 2, pp. 79–116, 1998.
[28] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. Comput. Vis.*, vol. 30, no. 2, pp. 79–116, 1998.
[29] K. Mikolajczyk, B. Leibe, and B. Schiele, "Local features for object class recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2. Oct. 2005, pp. 1792–1799.
[30] L. Wiskott, J. Fellous, N. Kruger, and C. Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 775–779, Jul. 1997.
[31] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, Apr. 2002.
[32] A. Elenyan, H. Ozkaramanli, and H. Demirel, "Complex wavelet transform-based face recognition," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 185281, pp. 1–5, Jan. 2008.
[33] R. Alferez and Y. F. Wang, "Geometric and illumination invariants for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 6, pp. 505–536, Jun. 1999.
[34] M. I. Khalil and M. M. Bayoumi, "Invariant 2D object recognition using the wavelet modulus maxima," *Pattern Recognit. Lett.*, vol. 21, no. 9, pp. 863–872, Aug. 2000.
[35] V. Kyrki, J. Kamarainen, and H. Kalviainen, "Simple gabor features space for invariant object recognition," *Pattern Recognit. Lett.*, vol. 25, no. 3, pp. 311–318, Feb. 2003.
[36] N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *J. Appl. Comput. Harmon. Anal.*, vol. 10, no. 3, pp. 234–253, May 2001.
[37] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 123–151, Nov. 2005.
[38] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[39] T. Sattler, B. Leibe, and L. Kobbelt, "SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2090–2097.
[40] C. Cui and K. N. Ngan, "A novel geometric filter for affine invariant features," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 865–868.
[41] A. Schmidt, M. Kraft, and A. Kasinski, "An evaluation of image feature detectors and descriptors for robot navigation," in *Proc. Comput. Vis. Graph.*, vol. 2010, no. 6375, 2010, pp. 251–259.

Ee Sin Ng received the B.Eng. degree in electrical and electronic engineering and the M.Sc. degree in communications and digital signal processing from Imperial College, London, U.K., in 2003 and 2004, respectively. He is currently pursuing the Ph.D. degree in computer vision and image processing with Cambridge University, Cambridge, U.K.

His current research interests include computer vision, image processing, and machine learning.

Nick G. Kingsbury received the honours degree in 1970 and the Ph.D. degree in 1974 from the University of Cambridge, Cambridge, U.K.

He was with Marconi Space and Defence Systems, Portsmouth, U.K., from 1973 to 1983. Since 1983, he has been a Lecturer of communications systems and image processing with the University of Cambridge and a fellow of Trinity College, Cambridge. He was appointed as a Professor of signal processing in 2007, and is currently the Head of the Signal Processing and Communications Research Group. He has developed the dual-tree complex wavelet transform, and is especially interested in the application of complex wavelets and related multiresolution methods to the analysis of images and 3-D datasets. His current research interests include image analysis and enhancement techniques, object recognition, motion analysis, and registration methods.