

8

New approaches to robust, point-based image registration

DAVID M. MOUNT, NATHAN S. NETANYAHU,
AND SAN RATANASANYA

Abstract

We consider various algorithmic solutions to image registration based on the alignment of a set of feature points. We present a number of enhancements to a branch-and-bound algorithm introduced by Mount, Netanyahu, and Le Moigne (*Pattern Recognition*, Vol. 32, 1999, pp. 17–38), which presented a registration algorithm based on the partial Hausdorff distance. Our enhancements include a new distance measure, the *discrete Gaussian mismatch*, and a number of improvements and extensions to the above search algorithm. Both distance measures are robust to the presence of outliers, that is, data points from either set that do not match any point of the other set. We present experimental studies, which show that the new distance measure considered can provide significant improvements over the partial Hausdorff distance in instances where the number of outliers is not known in advance. These experiments also show that our other algorithmic improvements can offer tangible improvements. We demonstrate the algorithm’s efficacy by considering images involving different sensors and different spectral bands, both in a traditional framework and in a multiresolution framework.

8.1 Introduction

Image registration involves the alignment of two images, called the *reference image* and the *input image*, taken of the same scene. The objective is to determine the transformation from some given geometric group that most nearly aligns the input image with the reference image. Our interest in this problem stems from its application in remote sensing, and in particular in the alignment of satellite images of the Earth taken possibly at different times, by different sensors, and from different spectral bands. The goal is to establish very close alignment, say to within a fraction of a pixel. The problem is complicated by issues such as

obscuration due to the presence of cloud cover, variations caused by time (such as coastline changes due to tidal effects or variations in shadows at different times of the day), and variations to surface features due to the differences in the sensors and spectral bands. Image registration and point pattern matching are closely related computational problems. We refer the reader to surveys by Ton and Jain (1989), Brown (1992), and Zitová and Flusser (2003), and Alt and Guibas (1999) for further information on the many approaches to these problems.

There are two common approaches to image registration. One approach makes direct use of the original image data and the other is based on matching discrete geometric feature points. Both approaches have their relative advantages and disadvantages. This chapter focuses on methods based on matching feature points. Features may be extracted by a number of methods. In our experiments, we have used a feature extraction process based on identifying relatively high-valued coefficients from a wavelet decomposition of the image (Le Moigne *et al.*, 2002). This approach has the advantage that it can produce feature points at multiple levels of resolution. This can be used to drive a progressive multiresolution registration algorithm, which registers images at increasing levels of accuracy (Cole-Rhodes *et al.*, 2003; Netanyahu *et al.*, 2004; Zavorin and Le Moigne, 2005). For the applications we have in mind, the transformations to be considered involve two-dimensional geometric similarities, that is, transformations resulting from the composition of rotation, translation, and uniform scaling.

Accurate image registration (whether point-based or image-based) is a computationally expensive task, especially when large images or point sets are involved and when the transformation space has many degrees of freedom. Hence, it is of interest to develop algorithms that are both accurate and efficient. The formulation that we shall consider in this chapter is based on the following characteristics, founded on the taxonomy proposed by Brown (1992):

Input space: The images are assumed to be presented as a discrete set of two-dimensional feature points. In our experiments, extracted feature points were based on the most significant coefficients of a wavelet decomposition of each of the images (Le Moigne *et al.*, 2002).

Search space: Our software system supports affine registration transformations (allowing for translation, rotation, uniform and nonuniform scaling, and shearing). All of our experiments were conducted on a subspace consisting of similarity transformations (allowing for translation, rotation, and uniform scaling). The user provides intervals limiting the maximum and minimum degree of translation, rotation, and scaling.

Search strategy: Our algorithm is based on a search of the transformation space for the optimal aligning transformation. Specifically, it employs a geometric variant of a branch-and-bound search. See Section 8.3 for a detailed description. This is an extension of the algorithm presented by Mount *et al.* (1999).

Distance metric: We used two different robust measures as the objective function of our search algorithm, the (directed) partial Hausdorff distance (PHD) (Huttenlocher and Rucklidge, 1993; Huttenlocher *et al.*, 1993) and a new smoothed version of the symmetric difference distance called the *discrete Gaussian mismatch* (DGM). Both measures are robust in that they allow for missing, as well as spurious data points.

A large number of papers have been written on the point pattern matching problem in the fields of computer vision, pattern recognition, and computational geometry. It is beyond the scope of this chapter to survey the area in detail, and so we will focus on the most relevant results. Perhaps the simplest similarity among point sets involve the Hausdorff distance and its variants (Huttenlocher *et al.*, 1993; Alt *et al.*, 1994; Chew *et al.*, 1997; Goodrich *et al.*, 1999). The standard notion of Hausdorff distance is not suitable for our application, since it requires that every point (from at least one set) have a nearby matching point in the other set. Computing the optimal alignment of two-point sets even under the relatively simple Hausdorff distance is computationally intensive. In an attempt to circumvent the high complexity of point pattern matching, some researchers have considered *alignment-based* algorithms. These algorithms use alignments between small subsets of points to generate potential aligning transformations, the best of which are then subjected to more detailed analysis. Examples of these approaches include work in the field of image processing (Stockman *et al.*, 1982; Goshtasby and Stockman 1985; Goshtasby *et al.*, 1986) and in the field of computational geometry (Heffernan and Schirra, 1994; Goodrich *et al.*, 1999; Gavrilov *et al.*, 2004; Cho and Mount, 2005; Choi and Goyal, 2006). Alignments can also be part of a more complex algorithm. For example, Kedem and Yarmovski (1996) presented a method for performing stereo matching under translation based on propagation of local matches for computing good global matches.

For our applications it will be important that the distance measure be robust, in the sense that it is insensitive to a significant number of feature points from either set that have no matching point in the other set. Examples of a robust distance measures include the partial Hausdorff distance (PHD) (Huttenlocher and Rucklidge, 1993; Huttenlocher *et al.*, 1993) and symmetric and absolute differences (Alt *et al.*, 1996; Hagedoorn and Veltkamp, 1999). (See Section 8.2 for definitions.)

In this chapter we discuss a number of extensions to the prior work of Mount *et al.* (1999) on the problem of feature-based image registration. We have extended the software system of theirs to include the following new elements:

New distance measure: In addition to PHD, we introduce a new distance measure, called the discrete Gaussian mismatch (DGM), which offers a number of advantages over PHD.

New search algorithms: In addition to the two search algorithms introduced by Mount *et al.* (1999) (pure branch-and-bound and bounded alignment), we introduce a new search algorithm called *bounded least-squares alignment*. We demonstrate that in many cases this new algorithm exhibits significantly better performance. We have also added new variants in which the aforementioned algorithms perform their search, and showed that these variants were considerably more efficient than those of Mount *et al.* (1999).

More extensive experiments: We extend the experimental results of Mount *et al.* (1999) to consider registration of satellite images arising from different platforms and covering different spectral bands.

The rest of this chapter is organized as follows. In Section 8.2 we present the two distance functions that will be used by our algorithm. In Section 8.3, we present our registration algorithm. In Section 8.4 we discuss the results of our experiments on these algorithms.

8.2 Distance measures

The point-based registration can be defined abstractly as follows: We are given two point sets A and B . We refer to A as the *input set* and B as the *reference set*. We are given a space \mathcal{T} of geometric transformations, including some a-priori limits on the range of transformations. (For example, we may limit the range of rotations to some interval of angles.) We are also given some distance function that measures the degree of dissimilarity of the two-point sets. The problem is to find the transformation $\tau \in \mathcal{T}$ that minimizes the distance between $\tau(A)$ and B .

There are two natural sources of error. The feature extraction process is subject to *noise*, that is, small errors in the coordinates due to sensing errors and digitization. The second source of error is the presence of *outliers*, that is, feature points from either image that are not present in the other image. As mentioned in the introduction, outliers can result from many different sources, and may constitute a relatively large (often unknown) fraction of the feature points. Following terminology from statistics (Rousseeuw and Leroy, 1987), we say that a distance measure is *robust* if it is insensitive to the presence of outliers.

In this chapter we consider two robust distance measures. The first is the *partial Hausdorff distance* (PHD) introduced by Huttenlocher and Rucklidge (1993) and Huttenlocher *et al.* (1993). Consider the set of distances resulting from taking each point in one set, and finding the nearest point to it in the other set. Rather than taking the sum or the maximum of these distances, which may be affected by outliers, we consider the median or, in general, the k th smallest distance. More formally, given a set $S \subset \mathbb{R}$, and $1 \leq k \leq |S|$, let $\text{rank}_k S$ denote the k th smallest element of S . Given a point set B and a point a , let $\text{dist}(a, B)$ denote the distance from a to its closest

point of B . Given two-point sets A and B , and an integer parameter $1 \leq k \leq |A|$, the *directed partial Hausdorff distance* of order k from A to B is defined to be

$$\text{PHD}_k(A, B) = \text{rank}_k\{\text{dist}(a, B) \mid \forall a \in A\}.$$

(Note that the standard directed Hausdorff distance arises as a special case when $k = |A|$.) To avoid the dependence on the size of A , we will replace the integer parameter k with a *quantile* q , where $0 < q \leq 1$ represents the fraction of inliers of A . We then set $k = \lceil q \cdot |A| \rceil$. The resulting measure is denoted by $\text{PHD}_q(A, B)$.

One shortcoming of the PHD is the need to estimate the value of k (or q), that is, the expected number of inliers. Since this quantity depends on characteristics of the images (such as the degree of cloud cover) that may be unknown at the time of registration, we introduce an alternative distance measure. This measure is motivated by the symmetric difference of two sets, that is, the number of points that are present in one set but not in the other. To allow for the presence of noise, we attach a weight to the existence of a match by a function that decreases with the distances between each point of A and its nearest neighbor of B . The user provides a positive real parameter σ , which intuitively represents the standard deviation of a Gaussian distribution. Each point $a \in A$ is assigned a weight based on a variant of the Gaussian distribution function applied to the distance to its nearest neighbor of B , such that,

$$w_\sigma(a) = \exp\left(-\frac{\text{dist}(a, B)^2}{2\sigma^2}\right).$$

Note that the weight is 1 if and only if a coincides with a point of B , and (depending on σ) decreases to zero as the distance increases. We define the *discrete Gaussian mismatch distance* (DGM) between A and B to be

$$\text{DGM}_\sigma(A, B) = 1 - \frac{\sum_{a \in A} w_\sigma(a)}{|A|}.$$

Observe that if every point $a \in A$ coincides with some point $b \in B$, then $\text{DGM}_\sigma(A, B) = 0$, and the distance increases to a maximum value of 1 as the degree dissimilarity between the two-point sets increases.

8.3 Framework of the registration algorithm

Our registration algorithms are based on a geometric branch-and-bound framework. This framework has been used by others including Huttenlocher and Rucklidge (1993), Rucklidge (1996), Rucklidge (1997), Hagedoorn and Velkamp (1999), and Mount *et al.* (1999). Recall that we are given two-point sets, an input set A and a reference set B and a space of transformations \mathcal{T} . The problem is to find $\tau \in \mathcal{T}$

that minimizes the distance function, which is either PHD or DGM. The PHD is also parameterized by the *inlier quantile* $0 < q \leq 1$ and the DGM is parameterized by the *standard deviation* σ . Let us assume that A and B will be fixed for the remainder of the discussion, and let us define $\text{PHD}_q(\tau)$ and $\text{DGM}_\sigma(\tau)$ to be the respective distance measures between the transformed input set $\tau(A)$ and the reference set B . Let PHD_{opt} and DGM_{opt} denote the minimum distances under PHD and DGM, respectively, over all transformations $\tau \in \mathcal{T}$.

There are a number of different ways to represent a transformation of \mathcal{T} . Henceforth we assume that \mathcal{T} consists of the space of geometric similarities (allowing for rotation, translation, and uniform scaling). We represent each such transformation by a four-element vector, whose entries are the rotation θ , the translation vector (t_x, t_y) , and the scaling factor s .

There are also a number of ways to define the approximation error for each of our distance measures. We introduce four nonnegative error parameters:

- ε_{rm} : the relative metric error bound,
- ε_{am} : the absolute metric error bound,
- ε_{rq} : the relative quantile error bound,
- ε_{aq} : the absolute quantile error bound.

Only three of these parameters will be relevant to a particular distance measure. Intuitively, the metric error involves errors in the distance between points and quantile error involves errors in the number of points. First, for PHD, define $q^- = (1 - \varepsilon_{rq})q$. Note that since $q^- \leq q$, we have $\text{PHD}_{q^-}(\tau) \leq \text{PHD}_q(\tau)$, for any τ . We say that a transformation τ is *approximately optimal* for PHD relative to these parameters if either

$$\text{PHD}_{q^-}(\tau) \leq (1 + \varepsilon_{rm}) \text{PHD}_{\text{opt}} \quad \text{or} \quad \text{PHD}_{q^-}(\tau) \leq \text{PHD}_{\text{opt}} + \varepsilon_{am}.$$

Thus, the approximate PHD solution is allowed to be less robust by a factor of $(1 - \varepsilon_{rq})$, and it may exceed the optimum distance by a relative error of ε_{rm} or an absolute error of ε_{am} .

For DGM, define $\sigma^+ = (1 + \varepsilon_{rm})\sigma$. Observe that $\text{DGM}_{\sigma^+}(\tau) \leq \text{DGM}_\sigma(\tau)$, for any τ . We say that a transformation τ is *approximately optimal* for DGM relative to these parameters if either

$$\text{DGM}_{\sigma^+}(\tau) \leq (1 + \varepsilon_{rq}) \text{DGM}_{\text{opt}} \quad \text{or} \quad \text{DGM}_{\sigma^+}(\tau) \leq \text{DGM}_{\text{opt}} + \varepsilon_{aq}.$$

Thus, the approximate DGM solution is allowed to match points in a neighborhood that is larger by a factor of $(1 + \varepsilon_{rm})$, and it may exceed the optimum mismatch distance by a relative error of ε_{rq} or an absolute error of ε_{aq} . Note that the parameters ε_{rm} and ε_{rq} are used for both distance functions. Their meanings are slightly different but closely related.

Note that the earlier branch-and-bound algorithms of Huttenlocher and Rucklidge (1993) and Hagedoorn and Velthkamp (1999) implicitly provide ε_{am} but not the other two parameters. The algorithm of Mount *et al.* (1999) does allow for the same error settings as used for PHD above, but it does not include the DGM distance.

We begin by describing the general structure of the algorithm. We will not prove the algorithm's correctness formally, but it is a straightforward modification of the proof of Mount *et al.* (1999). As mentioned earlier, the algorithm is based on a geometric branch-and-bound search of transformation space. The algorithm implicitly generates a search tree, where each node of the tree is identified with an axis-parallel hyperrectangle in four-dimensional transformation space (for rotation, x -translation, y -translation, and scale). Each such rectangle, or *cell*, represents a subspace of possible transformations. The user provides initial limits on the subset of transformations to be considered, and the root of the search tree is associated with the associated cell.

The search processes each cell in a recursive manner, starting with the root cell. At any time there are a collection of *active cells* and a candidate transformation that is the best seen so far by the search. Let τ^* denote this best transformation, and let dist^* denote the associated distance. When it can be determined that a cell does not contain a transformation whose distance is smaller than dist^* , the algorithm *kills* the cell, that is, it eliminates it from further consideration. If a cell cannot be killed then it is processed as discussed below. The processing will generally involve hierarchically partitioning the cell into smaller cells, which will then be added to the list of active cells. The algorithm terminates when all cells have been killed, or when a user-supplied upper bound on the maximum number of cells to be processed has been exceeded. Upon termination, the best transformation encountered, τ^* , is returned.

Let us now present the algorithm in greater detail. For each cell T that we process, we are interested in the transformation of this cell, for which the distance measure is smallest. We compute an upper bound $\text{dist}^+(T)$ and a lower bound $\text{dist}^-(T)$ on this smallest distance (explained below). For each upper bound, there will be a specific transformation that serves as a witness to this upper bound.

Upper bound. To compute the upper bound, we may sample any transformation from within the cell. There are a few ways in which to do this. Our software implements three different approaches. The first two were introduced by Mount *et al.* (1999), and the third is new to this chapter.

Pure: (PURE) The midpoint of the cell is selected as the candidate transformation to be used for the upper bound.

Bounded Alignment: (BA) When the cell satisfies a given set of conditions (details given by Mount *et al.* (1999)) a small number of point pairs is sampled repeatedly

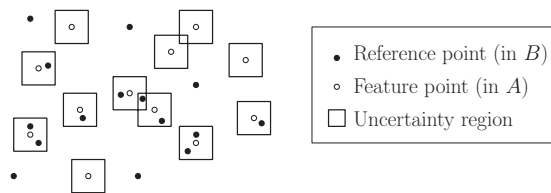


Figure 8.1. Uncertainty regions.

and at random from the subset of points of A that have a unique nearby point of B (after applying the cell's midpoint transformation). Each such pair $a_1, a_2 \in A$ is associated with its respective closest points $b_1, b_2 \in B$. There is a unique transformation τ mapping the pair (a_1, a_2) to (b_1, b_2) . The distance (i.e., the similarity measure) of this transformation is computed. The transformation that produces the smallest distance is chosen to provide the upper bound for the cell. In our experiments, 10 samples were used.

Bounded Least-Squares Alignment: (BLSA) Each point of A is associated with its closest point of B (after applying the cell's midpoint transformation). Given the resulting correspondences, the similarity transformation that minimizes the sum of squared distances between corresponding points is computed. This is done by an approach that first computes the transformation that aligns the centroids of the point sets, then computes the scale factor that aligns their spatial variances, and finally computes the rotation that minimizes the sum of squared distances. This is similar to an approach by Goshtasby *et al.* (1986) and Agarwal and Phillips (2006).

Nearest neighbors are computed by storing the points of B in a kd-tree data structure, and applying known efficient search techniques (Friedman *et al.*, 1977; Arya *et al.*, 1998; Arya and Mount, 2001). Let $\text{dist}^+(T)$ be the distance of the resulting sampled transformation.

Lower bound. To compute the lower bound, we use a technique presented by Mount *et al.* (1999), which is similar to that described by Huttenlocher and Rucklidge (1993) and Hagedoorn and Velkamp (1999). Given any cell $T \subset \mathcal{T}$, and given any point $a \in A$, consider the image of a under every $\tau \in T$.

We compute a bounding rectangle enclosing this region, which we call the *uncertainty region* of a relative to T . In this way, each cell is associated with a collection of uncertainty regions, one for each point of A . (See Fig. 8.1.)

Define the distance between an uncertainty region and a point $b \in B$ to be the minimum distance between b and any part of the uncertainty region. (If b lies inside the uncertainty region, then the distance is zero.) To derive our lower bound for T , for each point $a \in A$, we compute the distance from the corresponding uncertainty region to its nearest neighbor of B . Observe that this distance is a lower bound on the distance from $\tau(a)$ to its nearest neighbor of B , for any $\tau \in T$. We then apply

the distance computation to these lower bounds. For example, for the PHD we compute the q th smallest among these nearest neighbor distances, and for DGM we compute the match weights based on these distances. Let $\text{dist}^-(T)$ denote the result. Because it can never overestimate the distance from any point of A to its closest point of B , it is a lower bound on the actual PHD or DGM distance for any $\tau \in T$, and hence is indeed a lower bound for the cell. The nearest neighbor to an uncertainty region is computed by a straightforward generalization of the kd-tree-based nearest neighbor method described above.

Cell processing. As mentioned above, the algorithm operates by selecting an active cell T and processing it. Processing consists of the following steps. (A more detailed description is given by Mount *et al.*, 1999.) First, we compute the uncertainty regions for each point a and apply the aforementioned procedures to compute the upper and lower bound distances, $\text{dist}^+(T)$ and $\text{dist}^-(T)$. In the case of the PHD distance we kill the cell if either of the following two conditions hold:

$$\text{dist}^-(T) > \frac{\text{dist}^*}{1 + \varepsilon_{rm}} \quad \text{or} \quad \text{dist}^-(T) > \text{dist}^* - \varepsilon_{am}.$$

The rule for DGM is the same but replacing metric error parameters with their quantile counterparts. If $\text{dist}^+(T) < \text{dist}^*$, we set $\text{dist}^* \leftarrow \text{dist}^+(T)$ and save the associated transformation in τ^* . We then split the cell into two smaller subcells, T_1 and T_2 , which replace T in the set of active cells.

There are three strategies that we implemented for selecting the next cell to be processed:

Maximum Uncertainty (MAXUN): The next cell is the active cell with the largest average diameter of its uncertainty regions.

Minimum Upper Bound (MINUB): The next cell is the one with the smallest upper bound.

Minimum Lower Bound (MINLB): The next cell is the one with the smallest lower bound.

We refer to the above choices as the *search priority*. The MAXUN method was used by Mount *et al.* (1999) and the other two are new. Based on the fact that it demonstrated the best performance in our preliminary analysis, we used MINLB in all of our experiments.

8.3.1 Multiresolution registration

A common approach for improving the efficiency of image registration is to apply a multiresolution framework. This framework has been considered extensively in the context of image registration for remotely sensed images. See, for example,

Cole-Rhodes *et al.* (2003), Netanyahu *et al.* (2004), and Zavorin and Le Moigne (2005). The approach involves representing the two images at a series of increasingly finer spatial resolutions. Feature points are extracted from each of these images. This is followed by progressive registration of the resulting point sets by applying the registration process from the coarsest level to the finest. As we proceed from one level to the next, the spatial resolution increases by a factor of two. Thus, the coarsest level involves the greatest degree of spatial uncertainty, but also involves the smallest number of feature points. The process begins by registering the images at the coarse level, using a significantly wider range of transformations. The transformation generated by our program at each stage is used as a center point for the transformation cell at the next stage. Thus, as we proceed level by level, the accuracy of the aligning transformation is expected to improve, while the running time increases, due to the increased number of feature points expected with higher-resolution images.

As observed by Zavorin and Le Moigne (2005), there are a number of advantages of using a multiresolution approach compared to working solely with the original images. It can reduce computation time by performing much of the work at coarse resolutions, leaving minor adjustments to later stages. Since this type of image decomposition usually involves low-frequency smoothing, this regularizes the registration problem thus yielding better convergence properties and improved accuracy of the search algorithm. Finally, if image scales differ significantly, decomposition could be used to bring the two images into similar scales, which may be advantageous for some registration algorithms.

8.4 Experimental studies

In order to assess the performance of our registration algorithms, we have implemented a number of variants and tested their performance on a combination of remotely-sensed satellite imagery. The algorithms have been implemented in C++ (g++ version 3.2.3), and all experiments were run on a PC with a 2.4-GHz processor running Linux 2.4. Nearest neighbor and range queries were performed using kd-trees as generated by the ANN library for approximate nearest neighbor searching (Arya *et al.*, 1998). In particular, we were interested in studying the relative performance of:

Distance function: DGM versus PHD.

Search algorithm: PURE versus BA versus BLSA.

Our experiments involved satellite images that were taken from three distinct locations: *Konza* (Konza Prairie in the state of Kansas, July to August 2001), *Virginia* (Virginia's Hog Island Coast Reserve Area, October 2001), *Cascades*

(Cascades Mountains, September 2000). In each case the images were taken from two satellite platforms (sensors), IKONOS (4 meters per pixel) and ETM+ (30 meters per pixel), and involved the red and near infrared (NIR) spectral bands. Thus, by considering all possible combinations, we have four images, denoted IKONOS-red (IR), IKONOS-NIR (IN), ETM+-red (ER), and ETM+-NIR (EN), for each location, resulting in six possible ways of pairing them for registration. Some examples are shown in Fig. 8.2.

We tested the performance of our algorithms for both the single-pair and multiresolution frameworks. The results of the multiresolution experiments will be discussed in Subsection 8.4.3. In all of our experiments we considered matches under similarity transformations. Unless otherwise stated, the transformation width allows 4° of rotation, 4 pixels of x - and y -translation, and 20% of scaling. The initial cell of the search was centered at a random point whose maximum distance from the ground truth transformation was 25% of the transformation width. For the PHD we used an inlier quantile of $q = 0.5$ and for DGM we used a standard deviation of $\sigma = 1.0$. In all cases the program was allowed to execute for at most 10 000 cells, but it usually terminated well before then. We also used the following settings for the various error parameters:

Relative metric error: $\varepsilon_{rm} = 0.1$ (for both PHD and DGM),

Relative quantile error: $\varepsilon_{rq} = 0.2$ (for both PHD and DGM),

Absolute metric error: $\varepsilon_{am} = 0.4$ (for PHD),

Absolute quantile error: $\varepsilon_{aq} = 0.05$ (for DGM).

Most of our experiments involve computing two measures of performance. The first is running time, measured in CPU seconds. The second is a measure of accuracy, called the *transformation distance*. This is designed to measure how close the computed transformation is to our best estimate of “ground truth.” We estimated the ground truth by visual inspection of the datasets and consensus of other image registration programs combined with prior analysis of these image (Le Moigne *et al.*, 2003). Examples of alignments produced for three of the data sets under the ground truth transformation are shown in Fig. 8.2. The transformation distance of a transformation τ is defined to be the average Euclidean distance of each point $p \in A$ of the input dataset from its image under this transformation, $\tau(p)$, and its image under the ground truth transformation.

8.4.1 Experiment 1: Comparison of distance functions

Our first experiment involves a comparison of the effectiveness of each of the distance functions. We used the simplest of the algorithms, namely pure branch-and-bound (PURE) together with the minimum lower bound (MINLB)

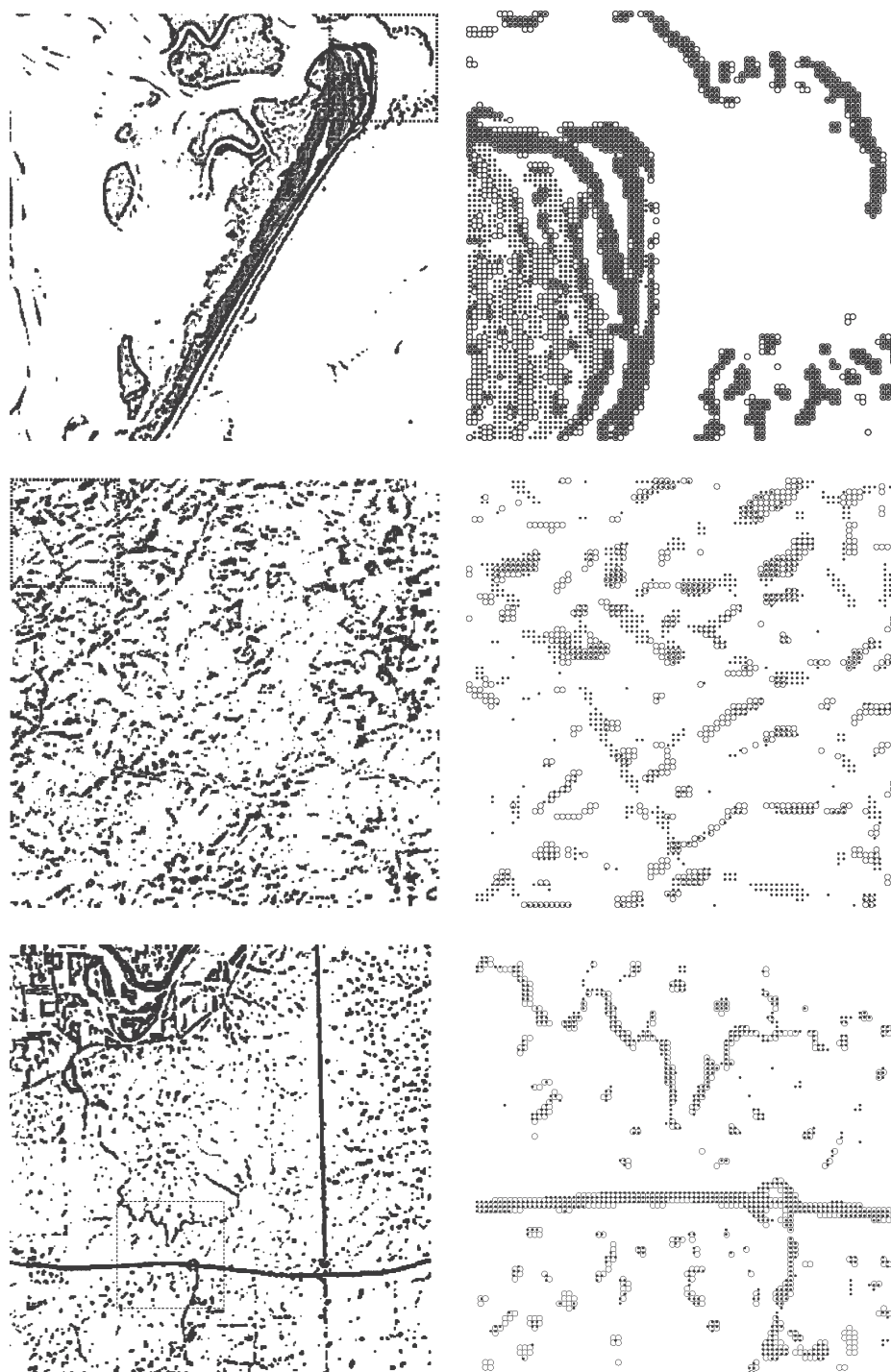


Figure 8.2. Feature-point sets for Experiment 3. Pixels of input and reference images are shown, respectively, as hollow and black points. Both sets are shown on the left, under our estimate of best aligning transformation. Detailed figures of highlighted subimages are shown on the right.

8 Point-based image registration

191

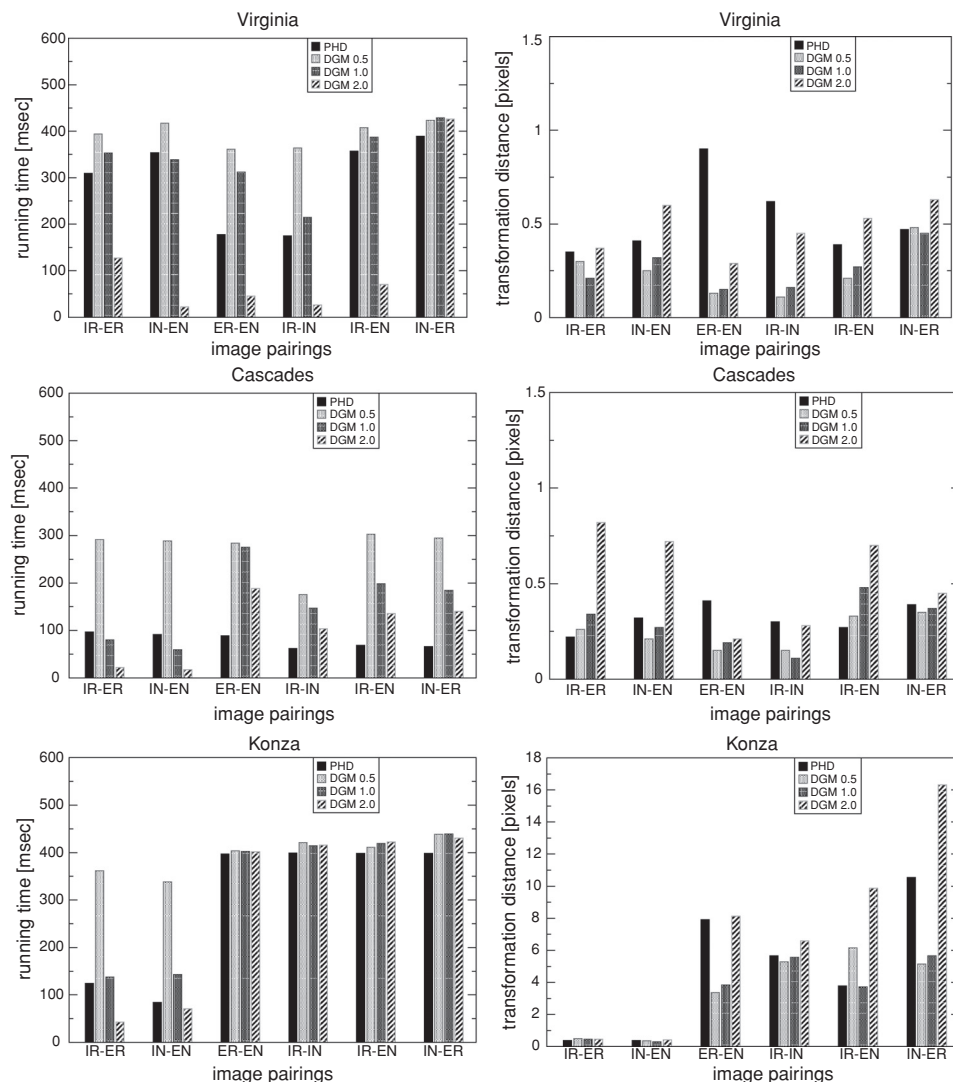


Figure 8.3. Results of Experiment 1 comparing the various distance functions. Relative performance, in terms of speed and accuracy, is shown as a function of distance measures.

search priority. We tested all six combinations of sensor-band pairs for each of the three images. For the discrete Gaussian mismatch, we tested values of $\sigma \in \{0.5, 1.0, 2.0\}$. For each experiment, we computed both execution time and transformation distance, and reported the average over five trials in each case. The results are presented in Fig. 8.3.

There are a number of conclusions that can be drawn from the experiments. First, increasing the standard deviation parameter σ in DGM tends to result in faster execution times, but also results in poorer performance with respect to transformation distance. (This is most clearly seen in the cases of Virginia and Cascades.) This is because increasing σ has the effect of making each feature point “fuzzier,” which in turn makes it easier to localize matches but makes the algorithm less sensitive to minor errors in placement. If we compare PHD with DGM 0.5, we see that DGM takes comparable time to compute and achieves as good or better transformation distances than PHD. In one case (Virginia IR-IN) the difference is quite dramatic. The problem with this dataset is evident from Fig. 8.2. This dataset has a high number of distinctive features that match very well, and it has a much lower number of outliers that do not match at all. The DGM measure seeks to match as many points as possible, while PHD is satisfied once it has matched the given quantile of points (which was $q = 0.5$ in this case). We feel that its greater degree of sensitivity to the actual number of outliers is the principal strength of DGM. Note that many of the Konza registrations were not successful. With the exception of IR-ER and IN-EN (both of which involve the same spectral band) the sets of feature points between the two images are very different.

A deeper understanding of the nature of distance functions is illustrated in Fig. 8.4. We first computed the value of the distance function with respect to ground truth. We then applied an additional horizontal shift of the input set A to both the left and right and reevaluated the cost function. One would expect the distance function to achieve a minimum at an offset of 0 and then to increase on either side. The ideal shape of an objective function is one that gradually descends towards a single, well-concentrated global minimum (at 0). The figure illustrates the challenges of doing this with the existing images. First, observe that both distance functions for Virginia exhibit not one, but many local minima. The PHD cost function is worse (note the different vertical scales), since it exhibits multiple local minima with identical distance values for $q = 0.25$ and $q = 0.5$. Thus, it is not surprising that the algorithm does not distinguish among these minima and produces an erroneous transformation. In the case of DGM, as the value of σ increases, the local minima are smoothed out (which explains the faster execution times) but the accuracy decreases as well. In the case of Konza, the objective function shows a single global minimum, but the objective function is not well concentrated for all parameter settings. Setting the σ value too small is problematic (as seen in the case of Konza for $\sigma = 0.1$) since there may be no transformation under which a significant number of feature points match within the σ bound.

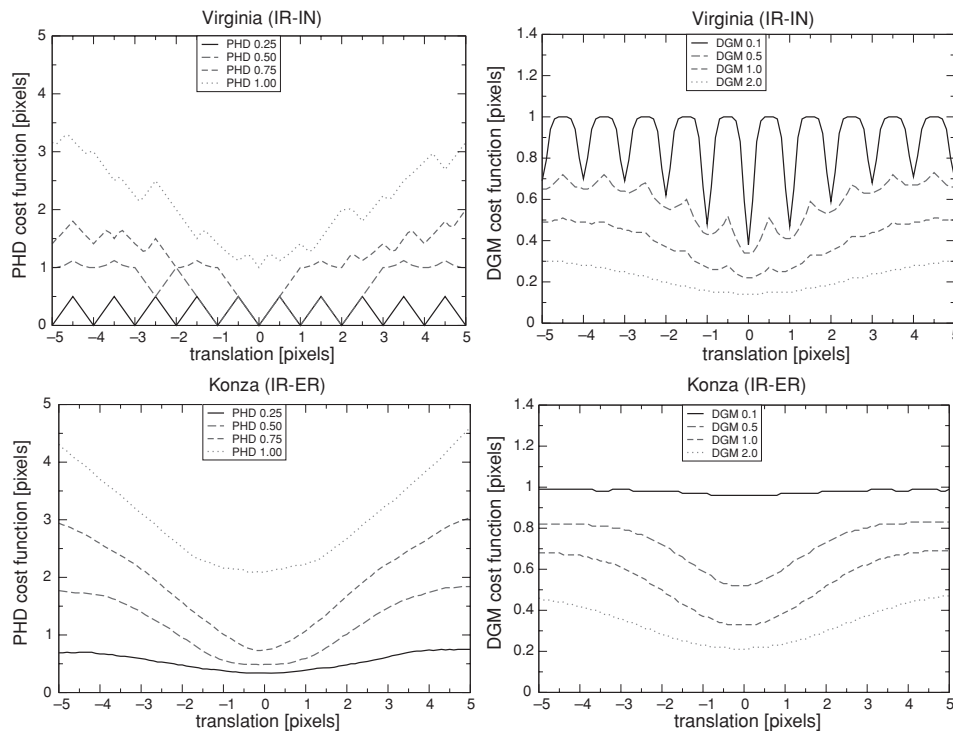


Figure 8.4. Comparison of objective functions subject to a horizontal shift relative to the ground truth transformation (at 0).

8.4.2 Experiment 2: Different sensors and bands

Given the relatively large number of possible experimental combinations (six pairs for registration from each of the three principal locations, for a total of 18), in our next experiment we compared the performance of the algorithms in all instances, in order to identify a relatively small set of representative cases. The hypothesis on which this experiment is based, is that the greater degree of commonality among the input images, the easier the registration should be in terms of running time and accuracy. Because some features are apparent only at certain spectral bands, we have noted that in the images tested, differences in the spectral band seem to be more significant the differences in the sensor. To test these effects, we grouped registration pairings into the following three groups:

- Case 1:** Images of the same spectral band but from different sensors (IR-ER, IN-EN).
- Case 2:** Images from the same sensor but from different spectral bands (ER-EN, IR-IN).
- Case 3:** Images from different spectral bands and different sensors (IR-EN, IN-ER).

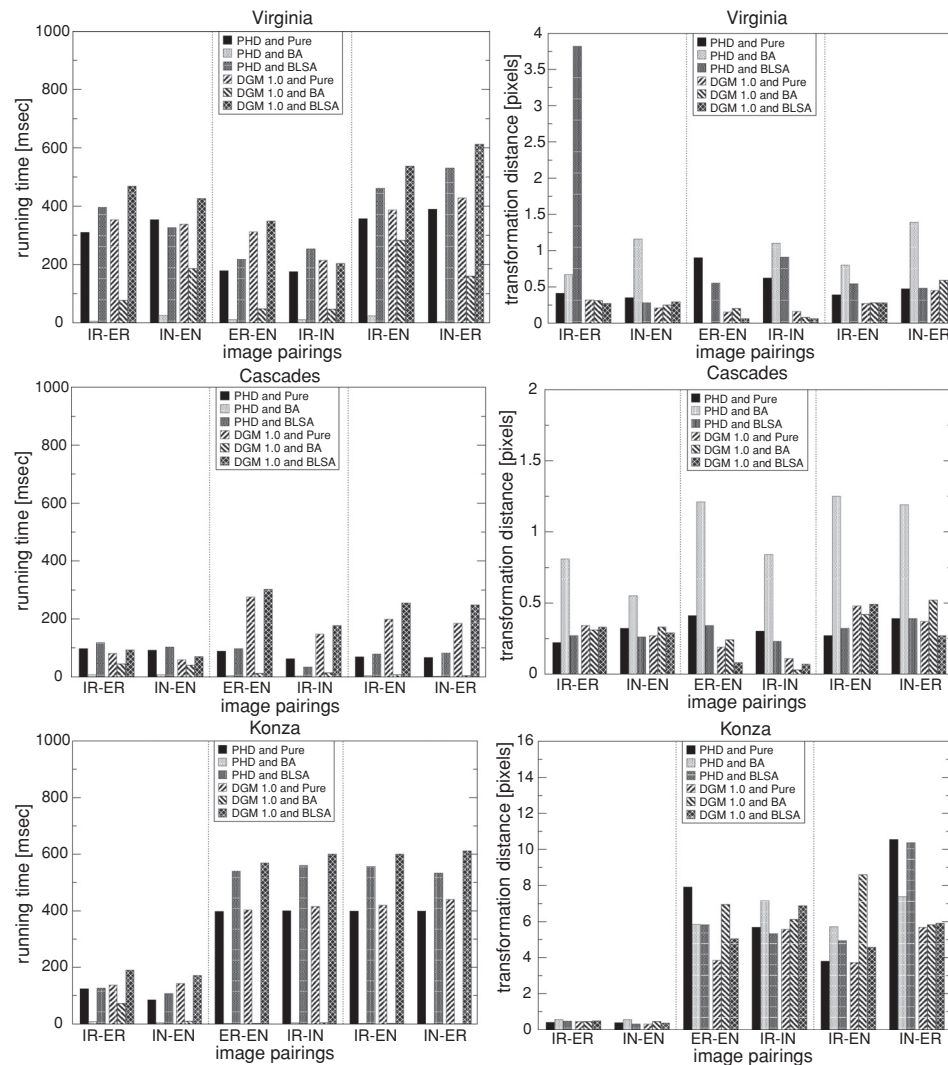


Figure 8.5. Results of Experiment 2 comparing the performance of the algorithms for various choices of sensor and spectral bands. See Plate 5 in color plates section.

For each of the three locations, we ran all six combinations of image pairings. The interpretation of the labels is given below. Based on the results of Experiment 1, we set $\sigma = 0.5$ for our experiments involving DGM, since this almost always produced the most accurate results. Throughout we considered matching under similarity transformations and used MINLB as the search priority. We tested all three search algorithms. As before, we measured execution time and transformation distance averaged over five trials. The results are presented in Fig. 8.5. Each plot is

split into three groups; the leftmost group corresponds to Case 1, the middle group to Case 2, and the right group to Case 3.

The experiments show that there are similarities among the various cases. This is most dramatically true for Konza and Cascades, where the patterns of execution times and transformation distances are notably similar within each group and dissimilar between groups. Among the different search algorithms, PURE and BLSA demonstrated generally steady and predictable performance. In contrast, BA was almost always the fastest of the algorithms, but it demonstrated the highest degree of variation in the quality of the results. In some instances (e.g., Virginia ER-EN), the BA algorithm convincingly outperformed the optimal transformation. However, in other instances (e.g., Virginia IR-EN and IN-ER), its performance was significantly worse. Inspection of the individual trials showed that in two out of five trials it found the optimal transformation, and in three cases it was off by a full pixel.

8.4.3 Experiment 3: Multiresolution framework

For this experiment we considered the performance of the algorithm in a multiresolution framework as described in Subsection 8.3.1. The algorithm was applied to four different resolution levels, and in each case the output from one level was used as the starting transformation for the next level. At the coarsest level of resolution we used a relatively high range of transformations, allowing for 16° of rotation, 32 pixels of x - and y -translation, and 30% of scaling. At all the other levels the transformation width allows 6° of rotation, 6 pixels of x - and y -translation, and 30% of scaling. Subsequent levels used the more restrictive transformation ranges described at the start of Section 8.4. Otherwise, we used the same parameter settings as in Experiment 2. As always, the results were averaged over five trials, each with a different random starting transformation. The other parameter settings were the same as in Subsection 8.4.2. We tested three cases, Virginia IR-IN (Case 2), Cascades IN-ER (Case 2), and Konza IR-ER (Case 1).

To determine the relative performance of our algorithms in the multiresolution framework, we measured execution time and transformation distance. The results are shown in Fig. 8.6. (Note that plots are on a logarithmic scale, and values less than 0.01 have been rounded up to 0.01.) A number of trends are apparent from the plots. First, as expected, the running times of the algorithm increase roughly exponentially with each subsequent level, since the image sizes and, hence, number of feature points increase similarly. Also, as expected, the transformation distances tend to decrease monotonically, since the accuracy of the feature points is increasing. There are two notable exceptions. In the case of PHD for Virginia IR-IN, the accuracy either exhibits very little change or actually gets worse (in the case of BLSA). Virginia is known to be a hard case for PHD, and this anomalous behavior reflects

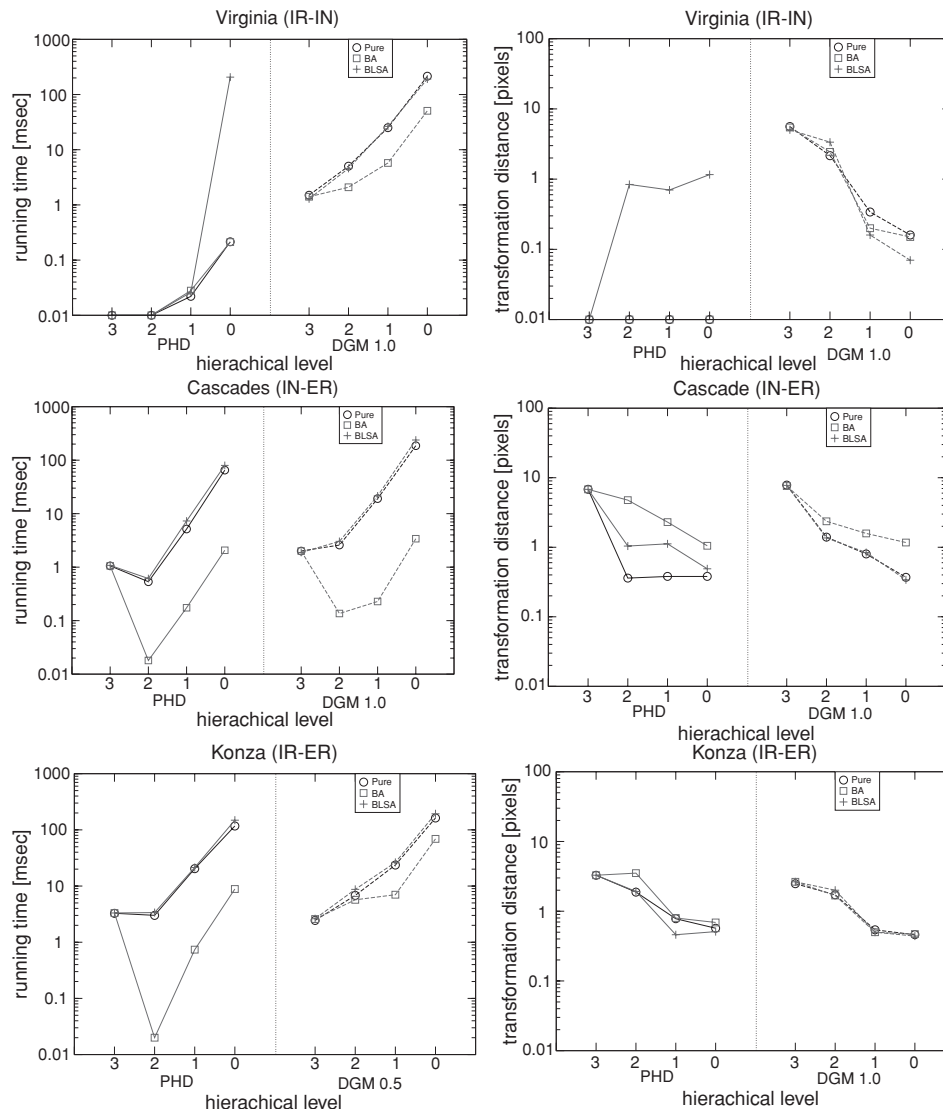


Figure 8.6. Results of Experiment 3 on multiresolution registration. The units for transformation distance are pixels at the highest level of resolution.

this fact. In contrast, DGM does quite well in this case. Other than this anomaly, both distance functions and all algorithms tended to perform quite similarly. As in the previous experiment, BA is the fastest of the methods. In terms of accuracy, it tends to be slightly worse in some instances than others.

Conclusions

In this chapter we have presented a number of enhancements to a feature-based registration algorithm introduced by Mount *et al.* (1999). In particular, we have

considered a new distance measure, the discrete Gaussian mismatch (DGM), new search algorithms based on a new method (BLSA) for computing the upper bound associated with a cell, and variants for selecting the next active cell to be processed, based on the cell's lower or upper bounds (MINLB and MINUB, respectively). Our experimental studies show that DGM is almost always as good as PHD, providing significant improvement in a few difficult cases. The other innovations offer tangible improvements, but these improvements are less pervasive and of lesser significance. Finally, we have demonstrated the algorithm's efficacy in significantly more general instances than reported by Mount *et al.* (1999), by considering images from different sensors and covering various spectral bands. We have further demonstrated the algorithm's efficacy in both a traditional single-pair framework and in a multiresolution framework.

Acknowledgements

The work of David Mount was partially supported by the National Science Foundation under grant CCR-0635099 and the Office of Naval Research under grant N00014-08-1-1015. The work of San Ratanasanya was supported in part by the National Science Foundation under grant CCR-0635099 while the author was visiting in the Department of Computer Science at the University of Maryland, College Park.

References

- Agarwal, P. K. and Phillips, J. M. (2006). On bipartite matching under the RMS distance. In *Proceedings of the Eighteenth Canadian Conference on Computational Geometry*, Kingston, Canada, pp. 143–146.
- Alt, H. and Guibas, L. J. (1999). Discrete geometric shapes: Matching, interpolation, and approximation. In J.-R. Sack and J. Urrutia, eds., *Handbook of Computational Geometry*. Amsterdam: Elsevier Science B.V., North-Holland, pp. 121–153.
- Alt, H., Aichholzer, O., and Rote, G. (1994). Matching shapes with a reference point. In *Proceedings of the Tenth Annual ACM Symposium on Computational Geometry*, Stony Brook, NY, pp. 85–91.
- Alt, H., Fuchs, U., Rote, G., and Weber, G. (1996). Matching convex shapes with respect to the symmetric difference. In *Proceedings of the Fourth Annual European Symposium on Algorithms*. London: Springer-Verlag, pp. 320–333.
- Arya, S. and Mount, D. M. (2001). Approximate range searching. *Computational Geometry: Theory and Applications*, **17**, 135–163.
- Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., and Wu, A. (1998). An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, **45**, 891–923.
- Brown, L. G. (1992). A survey of image registration techniques. *ACM Computing Surveys*, **24**, 325–376.
- Chew, L. P., Goodrich, M. T., Huttenlocher, D. P., Kedem, K., Kleinberg, J. M., and Kravets, D. (1997). Geometric pattern matching under Euclidean motion. *Computational Geometry Theory and Applications*, **7**, 113–124.

- Cho, M. and Mount, D. M. (2005). Improved approximation bounds for planar point pattern matching. In *Proceedings of the Ninth Workshop on Algorithms and Data Structures*, Waterloo, Canada; *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 3608, pp. 432–443.
- Choi, V. and Goyal, N. (2006). An efficient approximation algorithm for point pattern matching under noise. In *Proceedings of the Seventh Latin American Theoretical Informatics Symposium*, Valdivia, Chile; *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 3887, pp. 298–310.
- Cole-Rhodes, A. A., Johnson, K. L., Le Moigne, J., and Zavorin, I. (2003). Multiresolution registration of remote sensing imagery by optimization of mutual information using a stochastic gradient. *IEEE Transactions on Image Processing*, **12**, 1495–1511.
- Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, **3**, 209–226.
- Gavrilov, M., Indyk, P., Motwani, R., and Venkatasubramanian, S. (2004). Combinatorial and experimental methods for approximate point pattern matching. *Algorithmica*, **38**, 59–90.
- Goodrich, M. T., Mitchell, J. S. B., and Orletsky, M. W. (1999). Approximate geometric pattern matching under rigid motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**, 371–379.
- Goshtasby, A. and Stockman, G. C. (1985). Point pattern matching using convex hull edges. *IEEE Transactions on Systems, Man, and Cybernetics*, **15**, 631–637.
- Goshtasby, A., Stockman, G. C., and Page, C. V. (1986). A region-based approach to digital image registration with subpixel accuracy. *IEEE Transactions on Geoscience and Remote Sensing*, **24**, 390–399.
- Hagedoorn, M. and Velkamp, R. C. (1999). Reliable and efficient pattern matching using an affine invariant metric. *International Journal of Computer Vision*, **31**, 103–115.
- Heffernan, P. J. and Schirra, S. (1994). Approximate decision algorithms for point set congruence. *Computational Geometry Theory and Applications*, **4**, 137–156.
- Huttenlocher, D. P. and Rucklidge, W. J. (1993). A multi-resolution technique for comparing images using the Hausdorff distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, pp. 705–706.
- Huttenlocher, D. P., Kedem, K., and Sharir, M. (1993). The upper envelope of Voronoi surfaces and its applications. *Discrete and Computational Geometry*, **9**, 267–291.
- Huttenlocher, D. P., Klanderman, G. A., and Rucklidge, W. J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 850–863.
- Kedem, K. and Yarnovski, Y. (1996). Curve based stereo matching using the minimum Hausdorff distance. In *Proceedings of the Twelfth Annual ACM Symposium on Computational Geometry*, Philadelphia, Pennsylvania, pp. C15–C18.
- Le Moigne, J., Campbell, W. J., and Crompt, R. F. (2002). An automated parallel image registration technique of multiple source remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, **40**, 1849–1864.
- Le Moigne, J., Morissette, J., Cole-Rhodes, A., Netanyahu, N. S., Eastman, R., and Stone, H. (2003). Earth science imagery registration. In *Proceedings of the IEEE Geoscience and Remote Sensing Symposium*, Toulouse, France, pp. 161–163.
- Mount, D. M., Netanyahu, N. S., and Le Moigne, J. (1999). Efficient algorithms for robust point pattern matching. *Pattern Recognition*, **32**, 17–38.

- Netanyahu, N. S., Le Moigne, J., and Masek, J. G. (2004). Georegistration of Landsat data via robust matching of multiresolution features. *IEEE Transactions on Geoscience and Remote Sensing*, **42**, 1586–1600.
- Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. New York: Wiley.
- Rucklidge, W. J. (1996). Efficient visual recognition using the Hausdorff distance. *Lecture Notes in Computer Science*, Vol. 1173. Berlin: Springer-Verlag.
- Rucklidge, W. J. (1997). Efficiently locating objects using the Hausdorff distance. *International Journal of Computer Vision*, **24**, 251–270.
- Stockman, G. C., Kopstein, S., and Benett, S. (1982). Matching images to models for registration and object detection via clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **4**, 229–241.
- Ton, J. and Jain, A. K. (1989). Registering Landsat images by point matching. *IEEE Transactions on Geoscience and Remote Sensing*, **27**, 642–651.
- Zavorin, I. and Le Moigne, J. (2005). Use of multiresolution wavelet feature pyramids for automatic registration of multisensor imagery. *IEEE Transactions on Image Processing*, **14**, 770–782.
- Zitová, B. and Flusser, J. (2003). Image registration methods: A survey. *Image and Vision Computing*, **21**, 977–1000.