# Pixel-to-Pixel Matching for Image Recognition using Hungarian Graph Matching

Daniel Keysers, Thomas Deselaers, and Hermann Ney

Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
{keysers, deselaers, ney}@informatik.rwth-aachen.de

**Abstract.** A fundamental problem in image recognition is to evaluate the similarity of two images. This can be done by searching for the best pixel-to-pixel matching taking into account suitable constraints. In this paper, we present an extension of a zero-order matching model called the image distortion model that yields state-of-the-art classification results for different tasks. We include the constraint that in the matching process each pixel of both compared images must be matched at least once. The optimal matching under this constraint can be determined using the Hungarian algorithm. The additional constraint leads to more homogeneous displacement fields in the matching. The method reduces the error rate of a nearest neighbor classifier on the well known USPS handwritten digit recognition task from 2.4% to 2.2%.

## 1 Introduction

In image recognition, a common problem is to match two given images, e.g. when comparing an observed image to given references. In that process, different methods can be used. For this purpose we can define cost functions depending on the distortion introduced in the matching and search for the best matching with respect to a given cost function [6]. One successful and conceptually simple method for determining the image matching is to use a zero-order model that completely disregards dependencies between the pixel mappings. This model has been described in the literature several times independently and is called image distortion model (IDM) here. The IDM yields especially good results if the local image context for each pixel is considered in the matching process by using gradient information and local sub windows [5,6].

In this paper, we introduce an extension of the IDM that affects the pixel mapping not by incorporating explicit restrictions on the displacements (which can also lead to improvements [5,6]), but by adding the global constraint that each pixel in both of the compared images must be matched at least once. To find the best matching under this constraint, we construct an appropriate graph representing the images to be compared and then solve the 'minimum weight edge cover' problem that can be reduced to the 'minimum weight matching' problem. The latter can then be solved using the Hungarian algorithm [7]. The resulting model leads to more homogeneous displacement fields and improves the error rate for the recognition of handwritten digits. We refer to this model as the Hungarian distortion model (HDM).

The HDM is evaluated on the well known US Postal Service database (USPS), which contains segmented handwritten digits from US zip codes. There are many results for different classifiers available on this database and the HDM approach presented here achieves an error rate of 2.2% which is – though not being the best known result – state-of-the-art and an improvement over the 2.4% error rate achieved using the IDM alone.

**Related work.** There is a large amount of literature dealing with the application of graph matching to computer vision and pattern recognition tasks. For example, graph matching procedures can be used for labeling of segmented scenes. Other examples, more related to the discussed method include the following: In [9] the authors represent face images by elastic graphs which have node labels representing the local texture information as computed by a set of Gabor filters and are used in the face localization and recognition process. In [1] a method for image matching using the Hungarian algorithm is described that is based on representations of the local image context called 'shape contexts' which are only extracted at edge points. An assignment between these points is determined using the Hungarian algorithm and the image is matched using thin-plate splines, which is iterated until convergence. Yet, all applications of graph matching to comparable tasks that are known to the authors operate on a level higher than the pixel level. The novelty of the presented approach therefore consists in applying the matching at the pixel level.

## 2 Decision rule and image matching

In this work, we focus on the invariant distance resulting from the image matching process and therefore only use a simple classification approach. We briefly give a formal description of the decision process: To classify a test image $A$ with a given training set of references $B_{1k}, \ldots, B_{N_k k}$ for each class $k \in \{1, \ldots, K\}$ we use the nearest neighbor (NN) decision rule

$$r(A) = \arg\min_k \Big\{ \min_{n=1,\ldots,N_k} D(A, B_{nk}) \Big\},$$

i.e. the test image is assigned to the class of the nearest reference image. For the distance calculation the test image $A = \{a_{ij}\}, i = 1, \ldots, I, j = 1, \ldots, J$ must be explained by a suitable deformation of the reference image $B = \{b_{xy}\}, x = 1, \ldots, X, y = 1, \ldots, Y$. Here, the image pixels take $U$-dimensional values $a_{ij}, b_{xy} \in \mathbb{R}^U$, where the vector components are denoted by a superscript $u$. It has been observed in previous experiments that the performance of deformation models is significantly improved by using local context at the level of the pixels [5,6]. For example, we can use the horizontal and vertical image gradient as computed by a Sobel filter and/or local sub images that represent the image context of a pixel. Furthermore, we can use appropriately weighted position features (e.g. $\frac{i-1}{I-1}, \frac{j-1}{J-1}, \ldots$) that describe the relative pixel position in order to assign higher costs to mappings that deviate much from a linear matching.

We now want to determine an image deformation mapping $(x_{11}^{IJ}, y_{11}^{IJ}) : (i,j) \mapsto (x_{ij}, y_{ij})$ that results in the distorted reference image $B_{(x_{11}^{IJ}, y_{11}^{IJ})} = \{b_{x_{ij}y_{ij}}\}$. The resulting cost given the two images and the deformation mapping is defined as

$$C\big(A, B, (x_{11}^{IJ}, y_{11}^{IJ})\big) = \sum_{i,j} \sum_{u} \; ||a_{ij}^{u} - b_{x_{ij}y_{ij}}^{u}||^2,$$

i.e. by summing up the local pixel-wise distances, which are squared Euclidean distances here. Now, the distance measure between images $A$ and $B$ is determined by minimizing the cost over the possible deformation mappings:

$$D(A, B) = \min_{(x_{11}^{IJ}, y_{11}^{IJ}) \in \mathcal{M}} \Big\{ C\big(A, B, (x_{11}^{IJ}, y_{11}^{IJ})\big) \Big\}$$

The set of possible deformation mappings $\mathcal{M}$ determines the type of model used. For the IDM these restrictions are $x_{ij} \in \{1, \ldots, X\} \cap \{i' - w, \ldots, i' + w\}$, $i' = \left[i \frac{X}{I}\right]$, $y_{ij} \in \{1, \ldots, Y\} \cap \{j' - w, \ldots, j' + w\}$, $j' = \left[j \frac{Y}{J}\right]$, with warp range $w$, e.g. $w = 2$. For different models, the minimization process can be computationally very complex. A preselection of the e.g. 100 nearest neighbors with a different distance measures like the Euclidean distance can then significantly improve the computation time at the expense of a slightly higher error rate.

### 2.1 Image distortion model

The IDM is a conceptually very simple matching procedure. It neglects all dependencies between the pixel displacements and is therefore a zero-order model of distortion. Although higher order models have advantages in some cases, the IDM is chosen here for comparison to the HDM since the Hungarian algorithm does not easily support the inclusion of dependencies between pixel displacements. The formal restrictions of the IDM are given in the previous section, a more informal description is as follows: for each pixel in the test image, determine the best matching pixel within a region of size $w \times w$ at the corresponding position in the reference image and use this match. Due to its simplicity and efficiency this model has been introduced several times in the literature with differing names. When used with the appropriate pixel-level context description it produces very good classification results for object recognition tasks like handwritten digit recognition [6] and radiograph classification [5].

### 2.2 Hungarian matching

The term ‘matching’ is a well-known expression in graph theory, where it refers to a selection of edges in a (bipartite) graph. We can also view the concept of pixel-to-pixel image matchings in this context. To do so, we construct an appropriate graph from the two images to be compared and apply the suitable algorithms known from graph theory. In this section we explore this application and use the so called Hungarian algorithm to solve different pixel-to-pixel assignment problems for images. The Hungarian algorithm has been used before to assign image region descriptors of two images to each other [1].

**Construction of the bipartite graph.** The construction of the bipartite graph in the case discussed here is straight forward: Each pixel position of one of the two images to be compared is mapped to a node in the graph. Two nodes are connected by an edge if and only if they represent pixels from different images. This means that the two components of the bipartite graph represent the two images. The weight of an edge is chosen to be the Euclidean distance between the respective pixel representations, possibly enlarged by penalties for too large absolute distortions.

**Outline of the Hungarian algorithm.** This outline of the Hungarian algorithm is included for the interested reader but it is not essential for the understanding of the proposed method. The outline follows [7, pp. 74–89], which was also the basis for the used implementation. The name 'Hungarian' algorithm is due to a constructive result published by two Hungarian mathematicians in 1931 that is used in the algorithm [7, p. 78].

To explain the basic idea, we assume that the weights of the edges are given by the entries of a matrix $W$ and we assume that both components of the graph have $N$ vertices and thus $W \in \mathbb{R}^{N \times N}$ is square. The goal of the algorithm is to find a permutation $\pi : \{1, \ldots, N\} \mapsto \{1, \ldots, N\}$ minimizing $\sum_{n=1}^{N} W_{n\pi(n)}$. Now, we can make the following observations:

(a) Adding a constant to any row or column of the matrix does not change the solution, because exactly one term in the sum is changed by that amount independent of the permutation.

(b) If $W$ is nonnegative and $\sum_n W_{n\pi(n)} = 0$ then $\pi$ is a solution.

Let two zeroes in $W$ be called independent if they appear in different rows and columns. The algorithm now uses the following 'Hungarian' theorem: The maximum number of mutually independent zeroes in $W$ is equal to the minimum number of lines (rows or columns) that are needed to cover all zeroes in $W$. Given an algorithm that finds such a maximum set of mutually independent zeroes and the corresponding minimum set of lines (as summarized below) the complete algorithm can be formulated as follows:

1. from each line (row or column) subtract its minimum element
2. find a maximum set of $N'$ mutually independent zeroes
3. *if* $N' = N$ such zeroes have been found: output their indices and stop
   *otherwise:* cover all zeroes in $W$ with $N'$ lines and find the minimum uncovered value; subtract it from all uncovered elements, and add it to all doubly covered elements; go to 2

To show that the algorithm always terminates and yields the correct result, it is necessary to illustrate how step 2 works. The detailed discussion of the termination is beyond the scope of this overview. We try to give a short idea and otherwise refer to [7]:

*1.* Choose an initial set of independent zeroes (e.g. greedily constructed) and call these 'special'. *2.* Cover rows containing one of the special zeroes and mark all other rows. *3.* While there are marked rows, choose the next marked row: for each zero in the row that is not in a covered column, two cases are possible: a) the column already contains a special zero in another row '$\rho$': cover the column and uncover and mark $\rho$. b) a new special zero is found and processed. When the row is processed completely, unmark it.

Termination of the algorithm is guaranteed, because in step 3 either the number of mutually independent zeroes or the number of covered columns is increased by the newly introduced zero and this can happen at most $N$ times. The total running time of this algorithm is $O(N^3)$, where the average case can be much lower if good initial assignments can be determined. This implies that the application of the HDM to large images is only possible at a high computational cost. Note that there are other algorithms to solve the assignment problem, but most of these algorithms are developed for special cases of the structure of the graph (which is always a complete bipartite graph here).

**Application of the Hungarian algorithm.** The Hungarian algorithm is a tool to solve an assignment problem. For image matching, we can determine the best matching of pixels onto each other, where each pixel is matched exactly once. It is possible to directly use the Hungarian algorithm, but in many cases it is more appropriate to match the pixels onto each other such that each pixel is matched at least once or such that each pixel of the test image is matched exactly once. This last case corresponds to the most frequently used setting. We then require that the reference image *explains* all the pixels in the test image. We thus have three applications of the Hungarian algorithm for image matching:

**Each pixel matched exactly once.** This case is trivial. Construct the weight matrix as discussed above and apply the Hungarian algorithm to obtain a minimum weight matching.

**Each pixel matched at least once.** For this case, we need to solve the 'minimum weight edge cover' problem. A reduction to the exact match case can be done following an idea presented in [3]:
*1.* construct the weight matrix as discussed above *2.* for each node find one of the incident edges with minimum weight *3.* subtract from each edge weight the minimum weight of both connected nodes as determined in the previous step *4.* make the edge weight matrix nonnegative (by subtracting the minimum weight) and apply the Hungarian algorithm *5.* from the resulting matching, remove all edges with a nonzero weight (their nodes are covered better by using the minimum weight incident edges) *6.* for each uncovered node add an edge with minimum weight to the cover

**Each pixel of the test image matched exactly once.** This task is solved by the image distortion model, we only need to choose the best matching pixel for each pixel in the test image.
Another method to obtain such a matching evolves from the previous algorithm if it is followed by the step: *7.* for each pixel of the test image delete all edges in the cover except one with minimum weight.
The resulting matching then does not have the overall minimum weight (as determined by the IDM) but respects larger parts of the reference image due to the construction of the matching. Therefore, the resulting matching is more homogeneous. In informal experiments this last choice showed the best performance and was used for the experiments presented in the following.

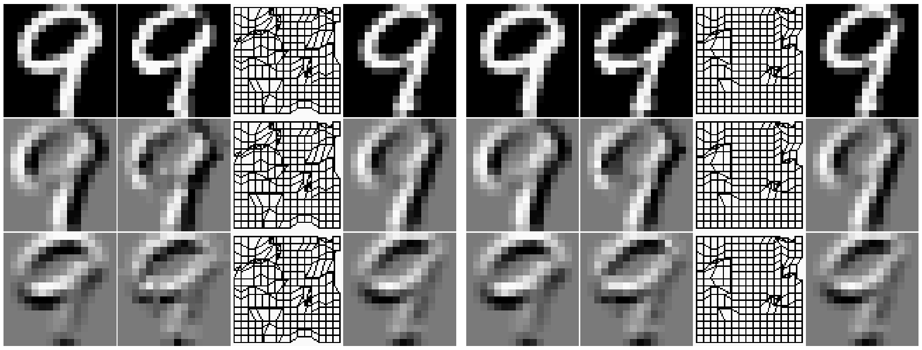

**Fig. 1.** Examples of pixel displacements; left: image distortion model; right: Hungarian distortion model. Top to bottom: grey values, horizontal, and vertical gradient; left to right: test image, distorted reference image, displacement field, and original reference image. The matching is based on the gradient values alone, using 3×3 local sub images and an absolute warp range of 2 pixels.

## 3 Experiments and Results

The software used in the experiments is available for download at http://www-i6.informatik.rwth-aachen.de/∼gollan/w2d.html. We performed experiments on the well known US Postal Service handwritten digit recognition task (USPS). It contains normalized greyscale images of size 16×16 pixels of handwritten digits from US zip codes. The corpus is divided into 7,291 training and 2,007 test images. A human error rate estimated to be 1.5-2.5% shows that it is a hard recognition task. A large variety of classification algorithms have been tried on this database and some of the best results are summarized in Table 1.

Figure 1 shows two typical examples of pixel displacements resulting from IDM and HDM in the comparison of two images showing the digit ‘9’. It can be observed that the HDM leads to a significantly more homogeneous displacement field due to the additional restriction imposed in the calculation of the mapping.

**Table 1.** Best reported recognition results for the USPS corpus (top: general results for comparison; bottom: results related to the discussed method)

| method | | ER[%] |
|---|---|---|
| invariant support vector machine | [8] | 3.0 |
| extended tangent distance | [6] | 2.4 |
| extended support vector machine | [2] | 2.2 |
| local features + tangent distance | [4] | 2.0 |
| ext. pseudo-2D HMM, local image context, 3-NN | [6] | 1.9 |
| no matching, 1-NN | | 5.6 |
| IDM, local image context, 1-NN | [6] | 2.4 |
| HDM, local image context, 1-NN | this work | **2.2** |

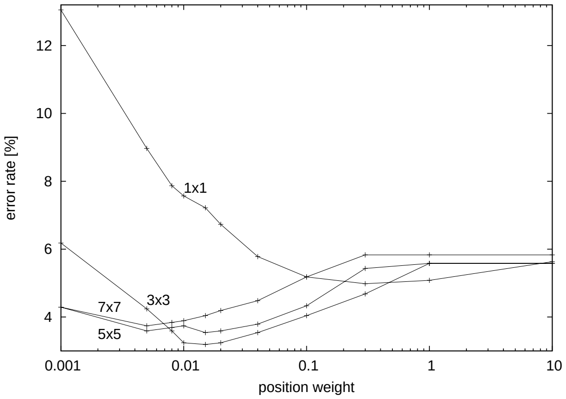

**Fig. 2.** Error rates on USPS vs. position weight and sub image size using HDM with greyvalues (preselection: 100 nearest neighbors, Euclidean distance).

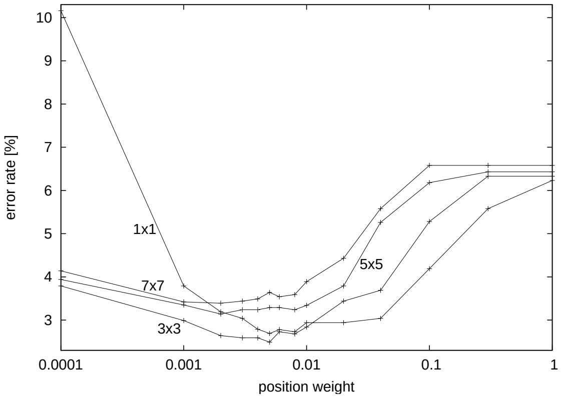

**Fig. 3.** Error rates on USPS vs. position weight and sub image size using HDM with gradients (preselection: 100 nearest neighbors, Euclidean distance).

Figure 2 shows the error rate of the HDM with respect to the weight of the position feature in the matching process. The pixel features used are the grayvalue contexts of sizes 1×1, 3×3, 5×5, and 7×7, respectively. Interestingly, already using only pixel greyvalues (1×1), the error rate can be somewhat improved from 5.6% to 5.0% with the appropriate position weight. Best results are obtained using sub images of size 3×3 leading to 3.2% error rate.

Figure 3 shows the error rate of the HDM with respect to the weight of the position feature using the vertical and horizontal gradient as the image features with different local contexts. Interestingly, the 1×1 error rate is very competitive when using the image gradient as features and reaches an error rate of 2.7%. Again, best results are obtained using sub images of size 3×3 and position weights around 0.005 relative to the other features, with an error rate of 2.4%.

All previously described experiments used a preselection of the 100 nearest neighbors with the Euclidean distance to speed up the classification process.

(One image comparison takes about 0.1s on a 1.8GHz processor for 3×3 gradient contexts.) Using the full reference set in the classifier finally reduces the error rate from 2.4% to 2.2% for this setting. Note that this improvement is not statistically significant on a test corpus of size 2,007 but is still remarkable in combination with the resulting more homogeneous displacement fields.

## 4 Conclusion

In this paper, we extended the image distortion model which leads to state-of-the-art results in different classification tasks when using an appropriate representation of the local image context. The extension uses the Hungarian algorithm to find the best pixel-to-pixel mapping with the additional constraint that each pixel in both compared images must be matched at least once. This constraint leads to more homogeneous displacement fields in the matching process. The error rate on the USPS handwritten digit recognition task could be reduced from 2.4% to 2.2% using a nearest neighbor classifier and the IDM and HDM as distance measures, respectively.

## Acknowledgments

This work was partially funded by the DFG (Deutsche Forschungsgemeinschaft) under contract NE-572/6. The authors would like to thank Christian Gollan for implementing the software framework the experiments benefited from.

## References

1. S. Belongie, J. Malik, J. Puzicha: Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
2. J.X. Dong, A. Krzyzak, C.Y. Suen: A Practical SMO Algorithm. In *Proc. Int. Conf. on Pattern Recognition*, Quebec City, Canada, August 2002.
3. J. Keijsper, R. Pendavingh: An Efficient Algorithm for Minimum-Weight Bibranching. Technical Report 96-12, Amsterdam Univ., Amsterdam, The Netherlands, 1996.
4. D. Keysers, R. Paredes, H. Ney, E. Vidal: Combination of Tangent Vectors and Local Representations for Handwritten Digit Recognition. In *SPR 2002, Statistical Pattern Recognition*, Windsor, Ontario, Canada, pp. 538–547, August 2002.
5. D. Keysers, C. Gollan, H. Ney: Classification of Medical Images using Non-linear Distortion Models. In *Proc. BVM 2004, Bildverarbeitung für die Medizin*, Berlin, Germany, pp. 366–370, March 2004.
6. D. Keysers, C. Gollan, H. Ney: Local Context in Non-linear Deformation Models for Handwritten Character Recognition. In *ICPR 2004, 17th Int. Conf. on Pattern Recognition*, Cambridge, UK, August 2004. In press.
7. D.E. Knuth: *The Stanford GraphBase: A Platform for Combinatorial Computing.* Addison-Wesley, Reading, MA, 1994.
8. B. Schölkopf, P. Simard, A. Smola, V. Vapnik: Prior Knowledge in Support Vector Kernels. In *Advances in Neural Information Processing Systems 10.* MIT Press, pp. 640–646, 1998.
9. L. Wiskott, J. Fellous, N. Krüger, C. v.d. Malsburg: Face Recognition by Elastic Bunch Graph Matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):775–779, July 1997.