Electronic Letters on Computer Vision and Image Analysis 0(0):1-7, 2000

Optimal Geometric Matching for Patch-Based Object Detection

Daniel Keysers* and Thomas Deselaers+ and Thomas M. Breuel*†

Image Understanding and Pattern Recognition Group
* *German Research Center for Artificial Intelligence (DFKI), D-67663 Kaiserslautern, Germany*
† *University of Kaiserslautern, D-67663 Kaiserslautern, Germany*
+ *Human Language Technology and Pattern Recognition Group*
RWTH Aachen University, D-52056 Aachen, Germany

Received 1 January 2000; revised 1 January 2000; accepted 1 January 2000

Abstract

We present an efficient method to determine the optimal matching of two patch-based image object representations under rotation, scaling, and translation (RST). This use of patches is equivalent to a fully-connected part-based model, for which the presented approach offers an efficient procedure to determine the best fit. While other approaches that use fully connected models have a high complexity in the number of parts used, we achieve linear complexity in that variable, because we only allow RST-matchings.

The presented approach is used for object recognition in images: by matching images that contain certain objects to a test image, we can detect whether the test image contains an object of that class or not. We evaluate this approach on the Caltech data and obtain very competitive results.

Key Words: object recognition, registration and matching

1 Introduction

We describe a new method for detecting the presence of an object in an image. This decision problem has applications for instance in the automatic indexing of large image and video databases and forms one of the basic problems of computer vision and pattern recognition. The contribution of this paper is to show that we can use a fully-connected part-based model to efficiently solve this problem. We evaluate the approach on the well known Caltech database [1] and achieve competitive error rates.

Today, many successful approaches that address the problem of general object detection use a representation of the image objects by a collection of local descriptors of the image content. Commonly, SIFT features [2] or just square subimages, called patches, are used to represent the parts. This paradigm has the advantage of being robust with respect to occlusions and background clutter in images. Changes of the relative position of the patches to each other can be handled in different ways and consequently various methods have been proposed in the literature.

A simple but nevertheless effective method is to disregard the relative position of the parts completely [3, 4]. Doing so, however, has the possible disadvantage that no information about the localization of the object is

Correspondence to: <daniel.keysers@dfki.de>

Recommended for acceptance by <name>
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

obtained. Other approaches use models in which the positions of the parts depend on one [5, 6] or up to two [7] root positions. These models allow efficient determination of the maximum likelihood position of the object in the image.

Note that [5] states that detection for a fully-connected part-based model has exponential complexity in the number of parts, while the method presented in this paper finds the optimal match of such a model in time *linear* in the number of parts considered. This is possible, because the search is organized over the transformation parameter space and simultaneously considers all parts. Note that this search organization is only feasible because we implicitly factor the dependencies between the locations of the parts in the image into the four components x-translation, y-translation, rotation, and scale. If we wanted to include all general dependencies, the algorithm would effectively become exponential again, because of the exponential growth of the search space with the number of parameters.

2 Outline of the Method

We first give an overview of the proposed method and discuss the design decisions taken. The two following sections then describe the feature extraction and the geometric matching in more detail. Figure 1 shows an illustration of the method.

We propose to directly match the parts distributed in a reference image that contains the object to those extracted in a test image. The RAST (Recognition by Adaptive Subdivision of Transformation Space) algorithm [8, 9] is able to determine the optimal matching under rotation, scaling, and translation efficiently. In the experiments, the matching between a pair of images was determined in one second on the average (on a standard PC with 1.8GHz clock cycle running Linux). According to [8], using the RAST approach is several orders of magnitudes faster than an equivalent exhaustive search. The RAST method permits globally optimal geometric matching. It demonstrably yields geometric matches that are at least as good as the Hough transform [10] or pose clustering [11], and performs better in practical settings because it permits the incorporation of additional constraints.

Among the various possibilities for representing the image parts, we choose to extract PCA transformed patches, which are extracted using a wavelet-based interest point detector [12], we choose a vector quantization into 2048 clusters obtained by a Linde-Buzo-Gray style clustering [13] using the Euclidean distance on the PCA-transformed patches. This number of clusters was found to be a good compromise between computing time and accuracy in [3] and was not changed during the experiments performed here. Here, we do not focus on feature extraction but focus on the proposed model. Therefore we choose features that were used in previous experiments, in which the patch position was not taken into account [3, 14] and have shown to work reasonably well. In [3], patches are extracted from the images, Gaussian Mixtures are estimated for vector quantization and all information about the patches except their closest cluster identifiers are discarded. Then, a histogram of of the patches is created to represent the images. This method is improved to be more robust wrt. brightness and scale changes in [14] by improving the feature extraction process. Thus, here the patch representation was not optimized for the method proposed here. In the matching, we only consider patches to match if they occur on the same scale and are assigned the same identifier by the vector quantizer.

By using the RAST algorithm, we are able to find the optimal matching for the equivalent of a fully-connected patch-based model. Note that in this work our goal is not to learn a model for each object, which however would be possible. Instead, we match all given training images that contain the object of interest to the test image. This approach is analogous to nearest neighbor classification, using the RAST score as a similarity measure. This procedure has the additional advantage that we determine the best-matching training image, which directly allows the use of the method in an object-based image retrieval scenario.

In the matching, we allow for a displacement of the patch positions by a predetermined number of pixels (four in the experiments). The score we use to describe the quality of a resulting matching is the number of patches that have been correctly matched.

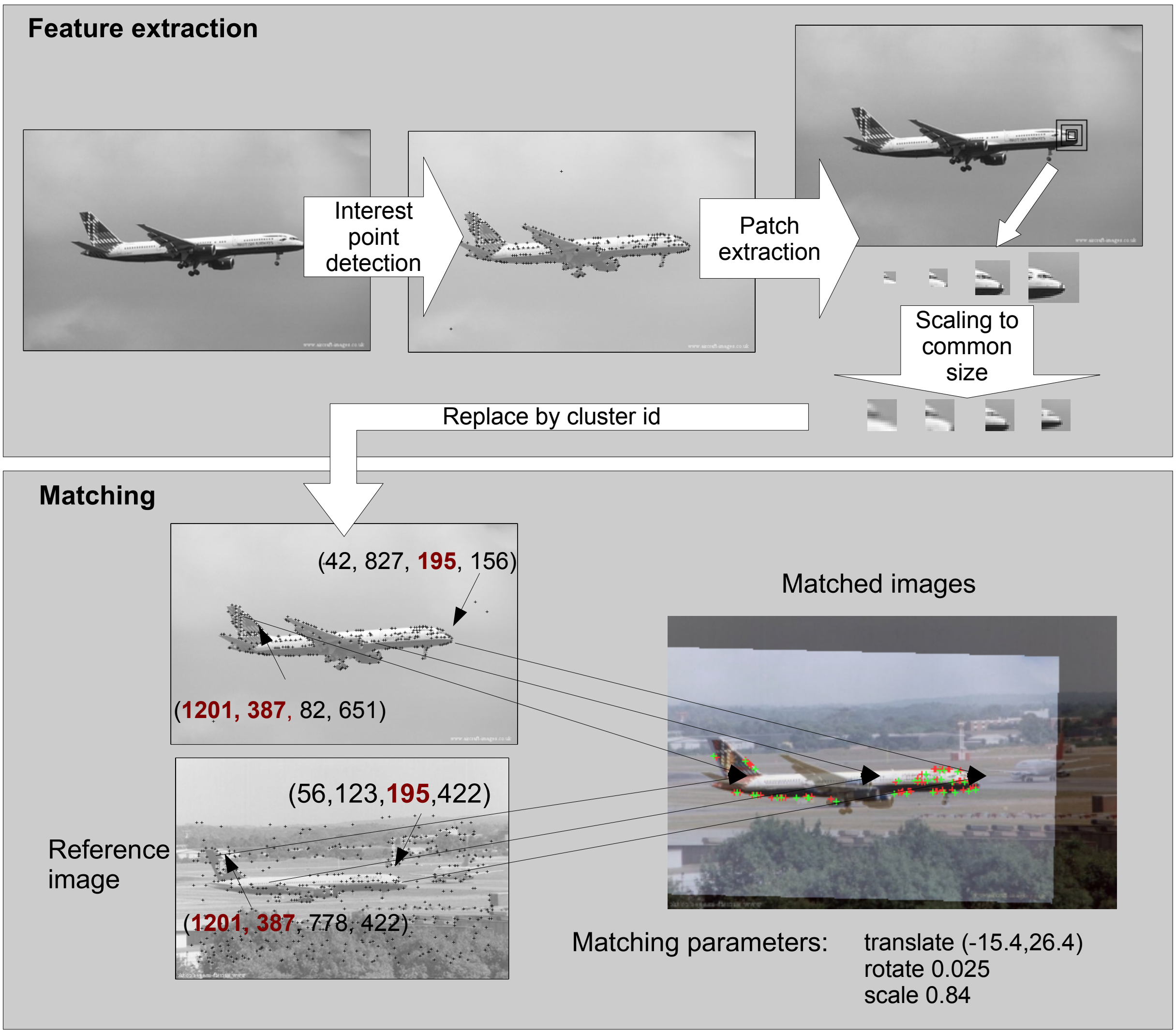

Figure 1: Illustration of the presented approach: top box: detection of interest points; extraction of patches in multiple scales and scaling to a common size. Then, the extracted patches are replaced by the identifiers of their closest clusters . In the bottom box, the interest points, represented by vectors of cluster identifiers, are matched to interest points, represented equally, of a reference image. Corresponding cluster identifiers are printed in red, bold letters. The optimal matching and the according transformation parameters are obtained by applying the RAST algorithm. The final image shows the reference image overlaid on the best matching database image transformed according to the obtained transformation parameters.

We are aware that the design decisions described in the previous paragraphs have alternatives that may also result in a good performance. However, no optimization of patch representation or other parameters has been done for the experiments presented in this work. To avoid over-fitting to the test data, we used the same parameters that were found to work well in [14, 3]. This makes it likely that the matching method could perform even better if more tuning would be applied.

Note that the proposed method does not need any segmentation of the input data in contrast to e.g. [15, 6]. It is likely, though, that the method would benefit from such a segmentation.

3 Patch Extraction

To extract the image patches, first, all images are converted into gray scale and scaled to a common height of 225 pixels. The scaling is applied because in the database we use for evaluation, the Caltech database, the background images are smaller than the training images, which may aid some classifiers [3]. Given an image, we extract patches of multiple sizes (7×7, 11×11, 21×21, 31×31 pixels) around up to 500 interest points obtained using the method proposed by Loupias et al. [12]. The use of patches of different sizes increases robustness to image scaling and allows to use visual clues that occur at different scales simultaneously. This procedure yields up to 2000 patches per image, 1730 on the average. The patches are allowed to extend beyond the image border, in which case the part of the patch falling outside the image is padded with zeroes. After the patches are extracted, they are scaled to a common size of 15×15 pixels to be able to determine a common code book for all extracted patches and to capture patch similarities across scale. A PCA dimensionality reduction is then applied to reduce the dimensionality of the data, keeping 40 coefficients. The first of these coefficients is discarded to achieve brightness normalization as it mainly encodes the overall image brightness [14]. The patches from all training images are then jointly clustered with a Linde-Buzo-Gray algorithm using the Euclidean distance such that 2048 clusters are obtained. Then we discard all information for each patch except its closest corresponding cluster center identifier. For the test data, this identifier is determined by evaluating the Euclidean distance to all cluster centers for each patch. Thus, the clustering assigns a cluster label $l(p) \in \{1, \dots L\}$ to each image patch p and allows to define a similarity of patches based on the cluster identifiers. For the matching, it is allowed to match two patches p and p' only if $l(p) = l(p')$. In principle, it is possible to represent each extracted patch by scores to all cluster centers and thus reducing the amount of information loss by vector quantization, however this would incur much higher costs for finding corresponding points in the final matching algorithm and thus would lead to strongly increased runtimes while not expecting a big gain in accuracy.

4 Determining the Optimal Matching

We now outline the RAST algorithm [16, 9] that we use for the determination of the optimal matching of the patch sets obtained from two images. Assume as input the sets of patches R for the reference and S for the test image. Each patch $p = (x_p, y_p, l_p)$ is a triple of x-position, y-position, and label, where the label here consists of the vector quantizer output and the scale at which the patch was extracted.

We are interested in finding the best transformation of the reference image to explain the patches observed in the test image. Here, we only consider the transformations translation, rotation, and scaling, although it is straightforward to use other sets of transformations. The transformations are characterized by a set of four parameters $\vartheta \in T$, i.e. translation in x- and y-direction, rotation angle, and scale factor. Here, T is the set of all possible initial parameter combinations as detailed below. We find the maximizing set of parameters

$$\hat{\vartheta}(R, S) := \arg\max_{\vartheta \in T} Q(\vartheta, R, S)$$

where the total quality $Q(\vartheta, R, S)$ of a parameter set is defined as the sum of local qualities

$$\begin{aligned} Q(\vartheta, R, S) &:= \sum_{p \in R} q(\vartheta, p, S) \\ q(\vartheta, p, S) &:= \begin{cases} 1 & \text{if } \exists p' \in S : l_p = l_{p'} \wedge d(\vartheta, p, p') \leq d_0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where $q(\vartheta, p, S)$ evaluates the goodness of fit for a given patch p and a set of parameters ϑ to the patches in S by assigning a one in case of a match within a distance d_0 that was set to $d_0 = 4$ pixels in the experiments. The Euclidean distance between the position of patch p transformed using the parameters ϑ and the position of patch p' is denoted by $d(\vartheta, p, p')$ here. Note that other local quality functions that correspond e.g. to Gaussian distributions rather than to bounded error can easily be introduced into the algorithm.

This maximization will be a complex task for most functional forms of Q. In many applications, such fits of parameters are carried out iteratively and heuristically, which involves the risk that the results found are only locally optimal solutions. Other methods include randomized approaches like e.g. random sample consensus [17].

We employ a branch-and-bound technique [8] to perform the maximization. This algorithm guarantees to find the globally optimal parameter set by recursively subdividing the parameter space and processing the resulting parameter hyper-rectangles in the order given by an upper bound on the total quality. Moreover, with small modifications, the algorithm allows us to efficiently determine the k best matches, not only the best match. Figure 2 shows an illustration of a subdivision of the transformation space and Figure 3 shows the subdivisions occurring during an actual run of the algorithm.

We determine an upper bound on the quality of parameters in a hyper-rectangular region T using

$$\max_{\vartheta \in T} Q(\vartheta, R, S) \leq \sum_{p \in R} \max_{\vartheta \in T} q(\vartheta, p, S)$$

where $\max_{\vartheta \in T} q(\vartheta, p, S)$ is straightforward to compute.

We can now organize the search as follows:

1. Pick an initial region of parameter values T containing all the parameters that we are interested in. (For the experiments we used the following settings: x-translation ±200 pixels, y-translation ±100 pixels, angle ±0.1 radians, scale factor in [0.8,1.2].)

2. Maintain a priority queue of regions T_i, where we use as the priority the upper bound on the possible values of the global quality function Q for parameters $\vartheta \in T_i$.

3. Remove a region T_i from the priority queue; if the upper bound of the quality function associated with the region is too small to be of interest, terminate. (When the upper bound of the quality is smaller than the value we are willing to accept as a match, we can be sure that no match that reaches this minimum quality can be reached and can therefore end the algorithm.)

4. If the region is small enough to satisfy our accuracy requirements, accept it as a solution.

5. Otherwise, split region T_i along the dimension furthest from satisfying our accuracy constraints and insert the subregions into the priority queue; continue at Step 3.

This algorithm will return the maximum quality match. To make the approach practical and avoid duplicate computations, we use a matchlist representation [16]. That is, with each region kept in the priority queue in the algorithm, we maintain a list (the matchlist) of all and only those patches that have the possibility to contribute with a positive local quality to the global quality. We maintain the list for each patch in the reference image. These matchlists will shrink quickly with decreasing size of the regions T_i. It is easy to see that the upper bound of a parameter space region T_i is also an upper bound for all subsets of T_i. When we split a region in

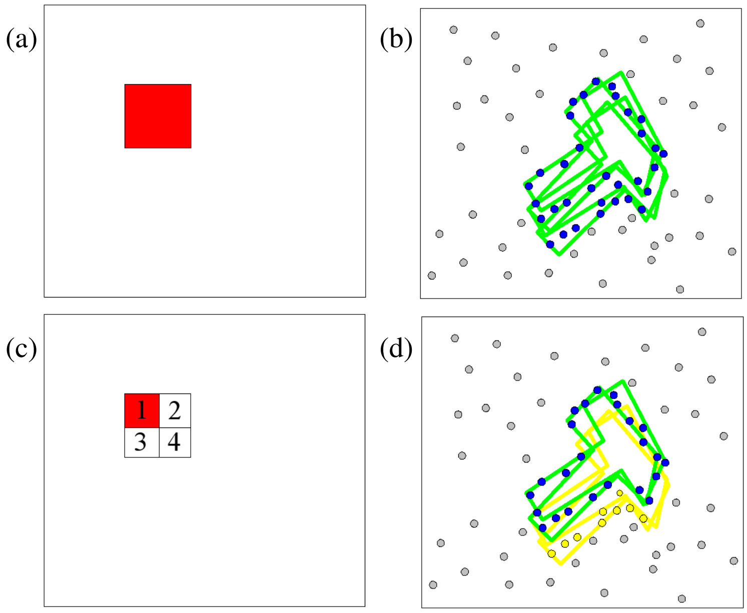

Figure 2: Illustration of the subdivision step within the RAST algorithm. (a),(c) show the region of the search space that is considered and (b),(d) show possible matchings of a model to points in the image for transformations with parameters contained in the region. (Note that these are not computed explicitly in the algorithm, but an upper bound of the quality for all possible matches is determined instead.) After splitting the region (c),(d), fewer transformations are possible and the upper bound for the quality of a match is recomputed accordingly. This process is repeated for each of the subregions.

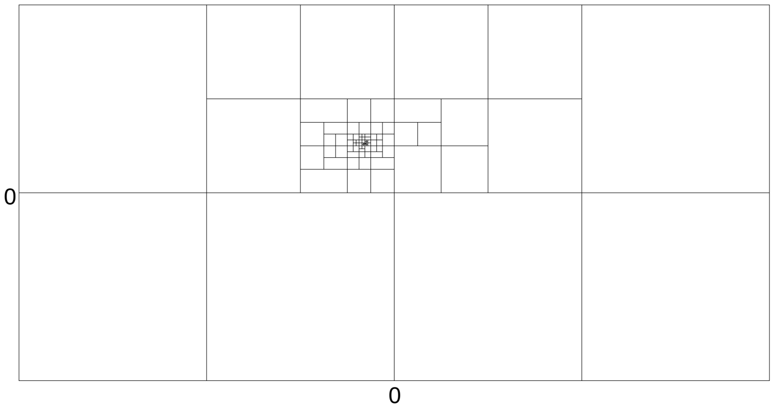

Figure 3: Illustration of the explored space during an actual run of the RAST algorithm. The two matched images are the ones shown in Figure 1. For the visualization we only searched for the translation component while keeping scale and angle fixed. We can observe how the subdivisions that occurred during the exploration of search space center around the final solution (-15.4, 26.4) and how large parts of the search space need not be explored in detail at all.

Step 5, we therefore never have to reconsider patches in the children that have already failed to contribute to the quality computation in the parent and thus the matchlists can be reused in the children.

The running time of the algorithm is largely determined by two factors:

- The time necessary to determine $\max_{\vartheta \in T} Q(\vartheta, R, S)$. This time is bounded by the product of the sizes of the sets R and S and therefore linear in the number of patches in the model as mentioned above. Note that, due to the use of matchlists as discussed above, the average number of comparisons is much smaller in each step. All other computations that are necessary in each subdivision step are much simpler and dominated by the determination of the upper bound.

- The number of times the initial region is split before a solution is reported. The interactions between the following variables influence this number:

  - The dimensionality of the search space: the number of splits tends to grow approximately exponentially with the dimensionality. However, in the application presented here, this dimensionality is always fixed at four.
  - The distribution of the patches in the images: the number of splits tends to decrease strongly if good matches are present.
  - The number of matching labels between R and S: fewer matches allow to reduce the matchlists and to find the solution with fewer splits.
  - The accuracy constraints imposed: if a more precise solution is needed, the number of splits increases.

5 Experiments and Results

The proposed method was evaluated on the Caltech data as introduced by Fergus et al. [1]. The task is to determine whether an object is present in a given image or not. For this purpose, several sets of images containing certain objects (airplanes, faces, and motorbikes) and a set of background images not containing any of these objects are available [*]. The images are of various sizes and for the experiments they were converted to gray scale. The airplanes and the motorbikes task consist of 800 training and 800 test images each, the faces task consists of 436 training and 434 test images. For each of these tasks, exactly half of the images contain the object of interest. Here, we only used the training images that contain an object.

In the experiments, the decision if a test image belongs to the object or background class was based on the following decision rule: decide for class ‘object’ if the average total quality for the best-fitting half of the training images is larger than a given threshold, otherwise decide for class ‘background’. The threshold is the parameter that is used to evaluate the results along the ROC curve. The motivation for this approach is to counteract the effect that one well-matching reference image has on the decision, because one such match often exists for the background class as well, but in much fewer cases there exist multiple good matches.

Table 1 shows the results obtained on the three Caltech data sets in comparison to those published by other groups. We give the equal error rate for each task for our approach. We observe that the error rates obtained are competitive, especially for the motorbikes set, even though the detection method was not tuned to the data set. The higher error rates for the airplanes tasks in comparison to the two other tasks may be partly caused by disregarding parts of the homogeneous background (sky) found in many images of the object class here due to the use of the interest point extractor. Another reason for the decreased accuracy on the airplanes task might be that airplanes landing or taking off show a higher degree of rotation than can be observed in the faces and motorbikes tasks and the lack of rotational invariance in the feature extraction. As mentioned above, the features used for the experiments were not optimized wrt. this particular method and we assume that a

[*] http://www.robots.ox.ac.uk/~vgg/data

Table 1: Comparison of experimental results on the Caltech data (error rates [%]).

method		airp.	faces	mot.
constellation model	[18]	32.0	6.0	16.0
automatic segmentation	[15]	2.2	0.1	10.4
texture feature combination	[19]	0.8	1.6	8.5
constellation model	[20]	9.8	3.6	7.5
PCA SIFT features	[21]	2.1	0.3	5.0
discrim. salient patches, SVM	[22]	7.0	2.8	3.8
spatial part-based model	[7]	6.7	1.8	3.0
constellation model	[5]	6.3	9.7	2.7
patch histograms	[3]	3.8	7.1	2.5
feat. inspired by visual cortex	[23]	3.3	1.8	2.0
patch histograms+	[14]	1.4	3.7	1.1
this work		4.8	2.8	1.3

performance increase could be obtained using better features, e.g. the improvements obtained in [14] over [3] are only due to improved feature extraction. Another improvement might be obtained by enriching the matching features by additional information about other cluster centers than the best matching one (cp. Section 3).

Figure 4 shows example results of the matching algorithm. (Recall that the matching uses gray value information only.) We show some good matches for the object and background class for all three tasks. Note that in some cases the matching recognizes the background instead of the object, as in example (b). This may seem to not be the intended behavior, but because the system does not know the position of the objects in the training image and no object model is explicitly learned, the method correctly retrieves the best match among the training images showing the same airport from a slightly shifted point of view. For the face images, in almost all the 'object' cases an image of the same person is chosen as the best-matching reference, in spite of changes in scale and lighting. This interesting behavior is however simplified by the fact that all images of one person seem to have been taken on the same day. Example (i) shows a special case, in which the background image also occurs as background in the reference image. Note how in examples (d,e,j,n) a part of the background test image is explained by a similar structure in the chosen reference image.

6 Conclusion

We presented a method to efficiently (i.e. in time linear in the number of patches) determine the optimal matching between two image objects based on the equivalent of a fully-connected patch-based model. The approach was evaluated on the Caltech data set using an appropriate decision rule based on the obtained matchings to the reference object data. The obtained quantitative results suggest that the method is well-suited for the task of matching image objects.

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03). We would like to acknowledge the use of software based on E. Loupias' salient point detector [12].

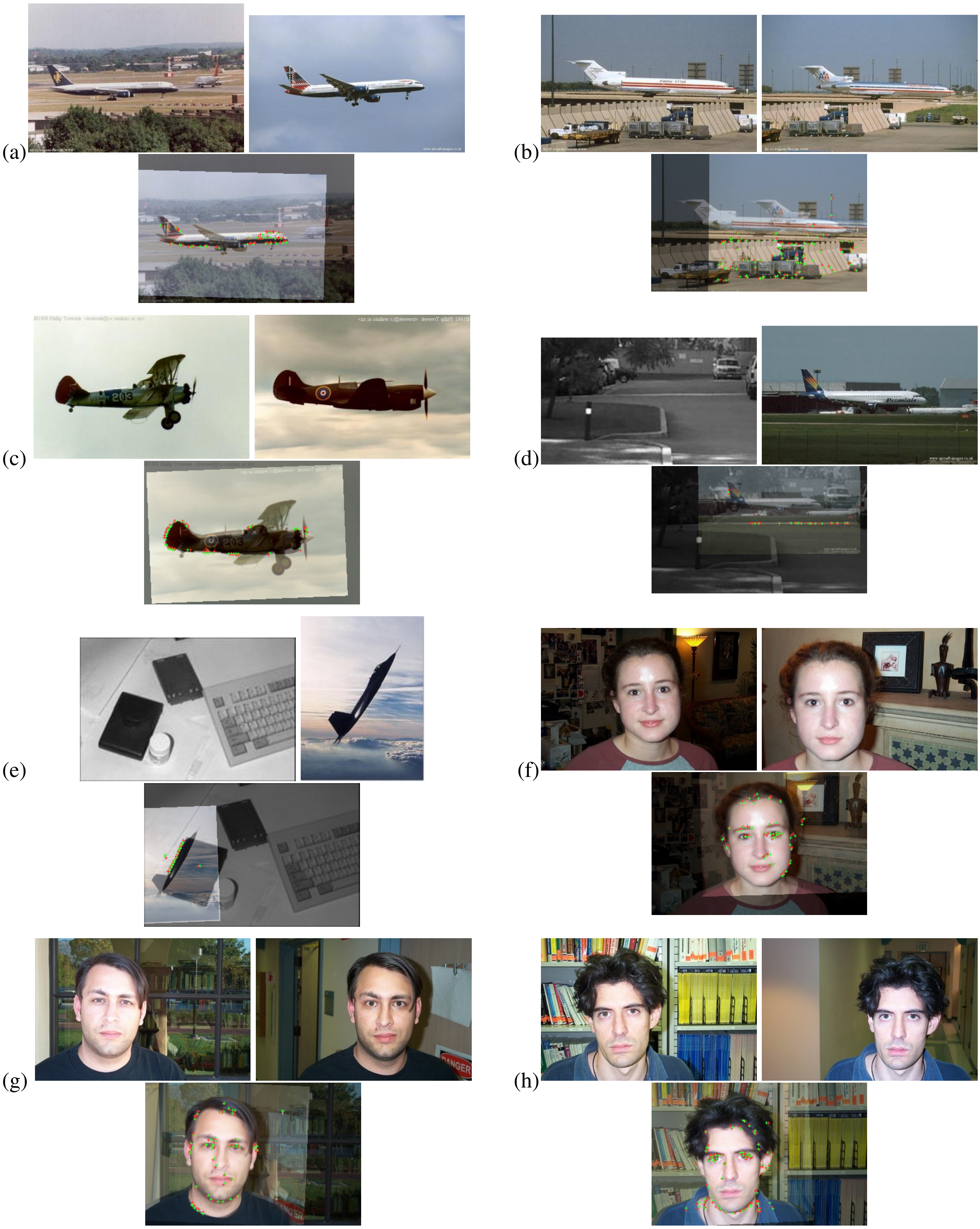

Figure 4: Examples of matching results. Each triple of images shows (top row) the test image, the matched reference image, and the reference overlaid on the test image after application of the determined transformation (bottom row). The crosses show the position of the matched patches. Note that we only match to reference images showing an object of the category and decide about presence or not using the average matching scores.

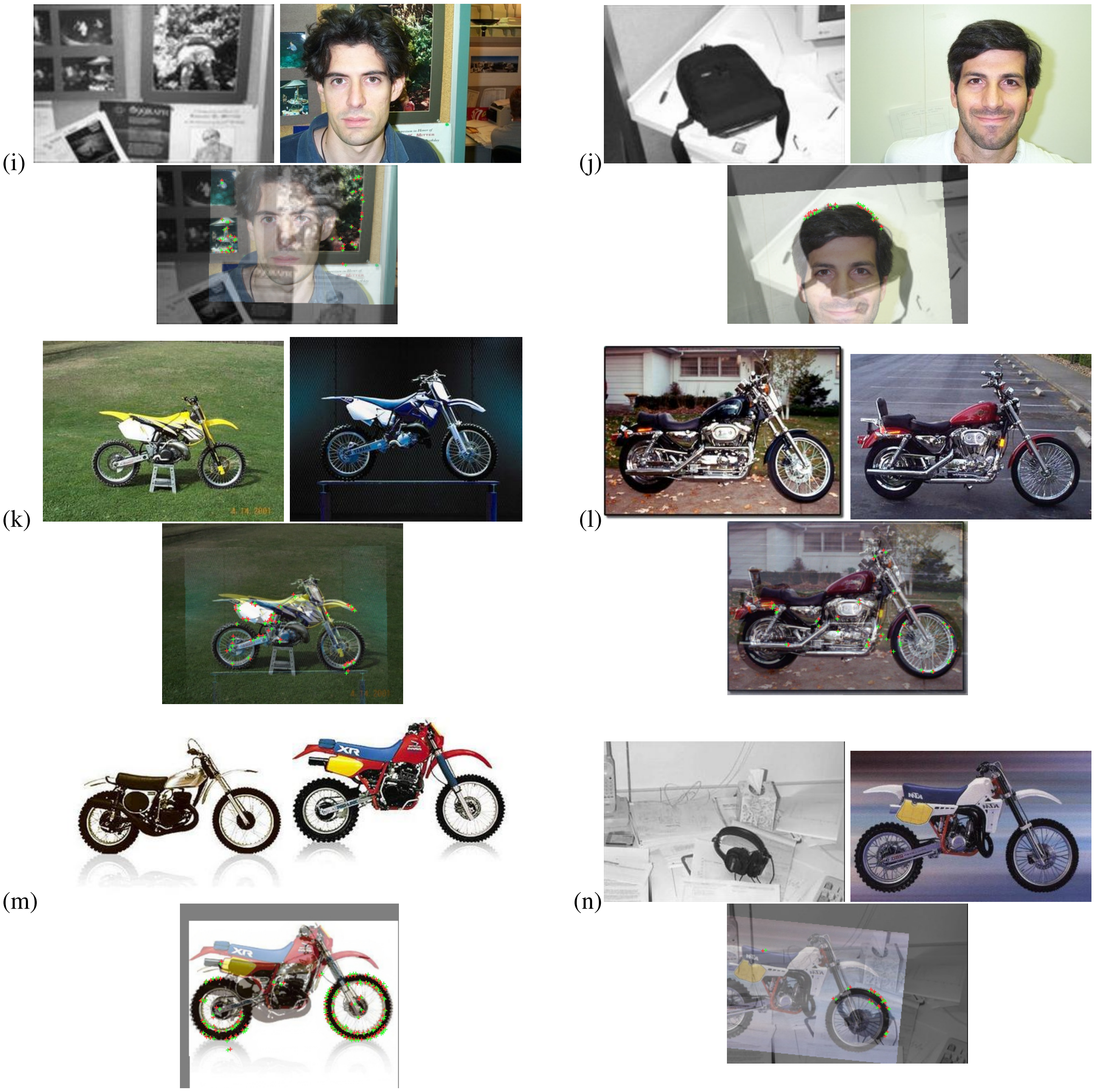

Figure 4 continued

References

[1] R. Fergus, P. Perona, and A. Zissermann. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Int. Conf. Computer Vision and Pattern Recognition*, Blacksburg, VG, pages 264–271, June 2003.

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

[3] T. Deselaers, D. Keysers, and H. Ney. Discriminative Training for Object Recognition using Image Patches. In *CVPR 2005, Int. Conf. on Computer Vision and Pattern Recognition*, volume II, San Diego, CA, pages 157–162, June 2005.

[4] R. Paredes, J. Perez-Cortes, A. Juan, and E. Vidal. Local Representations and a Direct Voting Scheme for Face Recognition. In *Workshop on Pattern Recognition in Information Systems*, Setúbal, Portugal, pages 71–79, July 2001.

[5] R. Fergus, P. Perona, and A. Zisserman. A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition. In C. Schmid, S. Soatto, and C. Tomasi, editors, *Conf. Computer Vision and Pattern Recognition*, volume 2, San Diego, CA, USA, pages 380–389, June 2005. IEEE.

[6] B. Leibe and B. Schiele. Scale Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search. In *DAGM*, number 3175 in LNCS, pages 145–153, August 2004.

[7] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial Priors for Part-Based Recognition using Statistical Models. In *Conf. Computer Vision and Pattern Recognition*, volume 1, San Diego, CA, pages 10–17, June 2005.

[8] T. M. Breuel. Implementation Techniques for Geometric Branch-and-Bound Matching Methods. *Computer Vision and Image Understanding*, 90(3):258–294, June 2003.

[9] T. M. Breuel. On the Use of Interval Arithmetic in Geometric Branch-and-Bound Algorithms. *Pattern Recognition Letters*, 24(9-10):1375–1384, June 2003.

[10] J. Illingworth and J. Kittler. A Survey of the Hough Transform. *Computer Vision, Graphics and Image Processing*, 44:87–116, 1988.

[11] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, vol.40, no.3:361–87, 1987.

[12] E. Loupias, N. Sebe, S. Bres, and J. Jolion. Wavelet-based Salient Points for Image Retrieval. In *Intl. Conf. Image Processing*, volume 2, Vancouver, Canada, pages 518–521, September 2000.

[13] Y. Linde, A. Buzo, and R. Gray. An Algorithm for Vector Quantizer Design. *IEEE Trans. Communications*, 28(1):84–95, 1980.

[14] T. Deselaers, D. Keysers, and H. Ney. Improving a Discriminative Approach to Object Recognition using Image Patches. In *DAGM 2005, Pattern Recognition, 27th DAGM Symposium*, volume LNCS 3663, Vienna, Austria, pages 326–333, August 2005.

[15] M. Fussenegger, A. Opelt, A. Pinz, and P. Auer. Object Recognition Using Segmentation for Feature Detection. In *ICPR*, volume 3, Cambridge, UK, pages 41–48, aug 2004.

[16] T. M. Breuel. Fast Recognition using Adaptive Subdivisions of Transformation Space. In *Int. Conf. Computer Vision and Pattern Recognition*, Champaign, IL, pages 445–451, June 1992.

[17] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24:381–395, 1981.

[18] M. Weber, M. Welling, and P. Perona. Unsupervised Learning of Models for Recognition. In *European Conf. Computer Vision*, volume 1, Dublin, Ireland, pages 18–32, June 2000.

[19] T. Deselaers, D. Keysers, and H. Ney. Classification Error Rate for Quantitative Evaluation of Content-based Image Retrieval Systems. In *ICPR 2004, 17th Int. Conf. on Pattern Recognition*, volume II, Cambridge, UK, pages 505–508, August 2004.

[20] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, Madison, WI, pages 264–271, June 2003.

[21] W. Zhang, B. Yu, G. J. Zelinsky, and D. Samaras. Object Class Recognition Using Multiple Layer Boosting with Heterogeneous Features. In *Conf. Computer Vision and Pattern Recognition*, volume 2, San Diego, CA, USA, pages 323–330, June 2005. IEEE.

[22] D. Gao and N. Vasconcelos. Discriminant Saliency for Visual Recognition from Cluttered Scenes. In *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, pages 481–488, 2005.

[23] T. Serre, L. Wolf, and T. Poggio. Object Recognition with Features Inspired by Visual Cortex. In *Conf. Computer Vision and Pattern Recognition*, volume 2, San Diego, CA, USA, pages 994–1000, June 2005.