Point Matching as a Classification Problem [*] for Fast and Robust Object Pose Estimation ID: IC/2004/06

Vincent Lepetit Julien Pilet Pascal Fua
Computer Vision Laboratory
Swiss Federal Institute of Technology (EPFL)
1015 Lausanne, Switzerland
Email: {Vincent.Lepetit, Julien.Pilet, Pascal.Fua}@epfl.ch

January 13, 2004

Abstract

We propose a novel approach to point matching under large viewpoint and illumination changes that is suitable for accurate object pose estimation at a much lower computational cost than state-of-the-art methods.

Most of these methods rely either on using ad hoc *local descriptors or on estimating local affine deformations. By contrast, we treat wide baseline matching of keypoints as a classification problem, in which each class corresponds to the set of all possible views of such a point. Given one or more images of a target object, we train the system by synthesizing a large number of views of individual keypoints and by using statistical classification tools to produce a compact description of this view set. At run-time, we rely on this description to decide to which class, if any, an observed feature belongs. This formulation allows us to use powerful and fast classification methods to reduce matching error rates.*

In the context of pose estimation, we present experimental results for both planar and non-planar objects in the presence of occlusions, illumination changes, and cluttered backgrounds. We will show that our method is both reliable and suitable for initializing real-time applications.

1 Introduction

While there are many effective approaches to tracking, they all require an initial pose estimate, which remains difficult to provide automatically, fast and reliably. Among methods that can be used for this purpose, those based on feature point matching have become popular since the pioneering work of Schmidt and Mohr [Schmid and Mohr, 1997] because this approach appears to be more robust to scale, viewpoint, illumination changes and partial occlusions than edge or eigen-image based ones. Recently, impressive wide-baseline matching results have been

[*] This work was supported in part by the Swiss National Science Foundation.

obtained [Tuytelaars and VanGool, 2000, Baumberg, 2000, Mikolajczyk and Schmid, 2002, Schaffalitzky and Zisserman, 2002, Lowe, 2003], which make this approach even more attractive.

These wide baseline matching methods, however, are typically designed to match two images but not to take advantage of the fact that, for pose estimation purposes, both a 3D object model and several training images may be available. In this paper, we propose a method that allows us to use this additional information to build compact descriptors that allow recognition of key feature points at a much reduced computational cost at run-time, without loss of matching performance. It also allows to relax the locally planar assumption. We will demonstrate this approach both on piecewise planar objects, such as books or boxes, and non-planar objects such as faces.

The key ingredient of our approach is to treat wide baseline matching of feature points as a classification problem, in which each class corresponds to the set of all possible views of such a point. During training, given at least one image of the target object, we synthesize a large number of views of individual keypoints. If the object can be assumed to be locally planar, this is done by simply warping image patches around the points under affine or homographic deformations. Otherwise, given the 3D model, we use standard Computer Graphics texture-mapping techniques. This second approach is more complex but relaxes the planarity assumptions. At run-time, we can then use powerful and fast classification techniques to decide to which view set, if any, an observed feature belongs, which is as effective and much faster than the usual way of computing local descriptors and comparing their responses. Once potential correspondences have been established between the interest points of the input image and those lying on the object, we apply a standard RANSAC-based method to estimate 3D pose. In Figure 1, we show how it can be used to initialize a 3D tracker we developed in previous work, and to re-initialize if it loses track.

Here, we do not focus on interest point extraction and use the Harris corner detector for our experiments. A more advanced detector such as the one described in [Lowe, 1999] could be used instead. It is therefore noteworthy that we nevertheless obtain excellent results, which can be attributed to the fact that our method tolerates imprecision in point localization. In short, this paper introduces a novel approach to point matching that goes a long way towards reducing the computational burden, thus making it suitable for fast object recognition under viewpoint and illumination changes.

In the remainder of the paper, we first discuss related work. We introduce our approach in Section 3 and present our results in Section 4. We conclude with our plans for future work.

2 Related Work

In the area of automated 3D object detection, we can distinguish between “Global” and “Local” approaches.

Global ones use statistical classification techniques to compare an input image to several training images of an object of interest and decide whether or not it appears in this input image. The methods used range from relatively simple methods such as Principal Component Analysis and Nearest Neighbor search [Nayar *et al.*, 1996] to more sophisticated ones such as AdaBoost and classifiers cascade to achieve real-time detection of human faces at varying scales [Viola and Jones, 2001]. Such approaches, however, are not particularly good at handling occlusions, cluttered backgrounds, or the fact that the pose of the target object may be very different from the ones in the training set. Furthermore, these global methods cannot provide accurate 3D pose estimation.

By contrast, local approaches use simple 2D features such as corners or edges [Jurie, 1999], which makes them resistant to partial occlusions and cluttered backgrounds: Even if some features are missing, the object can still be detected as long as enough are found and matched. Spurious matches can be removed by enforcing geometric constraints, such as epipolar constraints between different views or full 3D constraints if an object model is available [Allezard *et al.*, 2000].

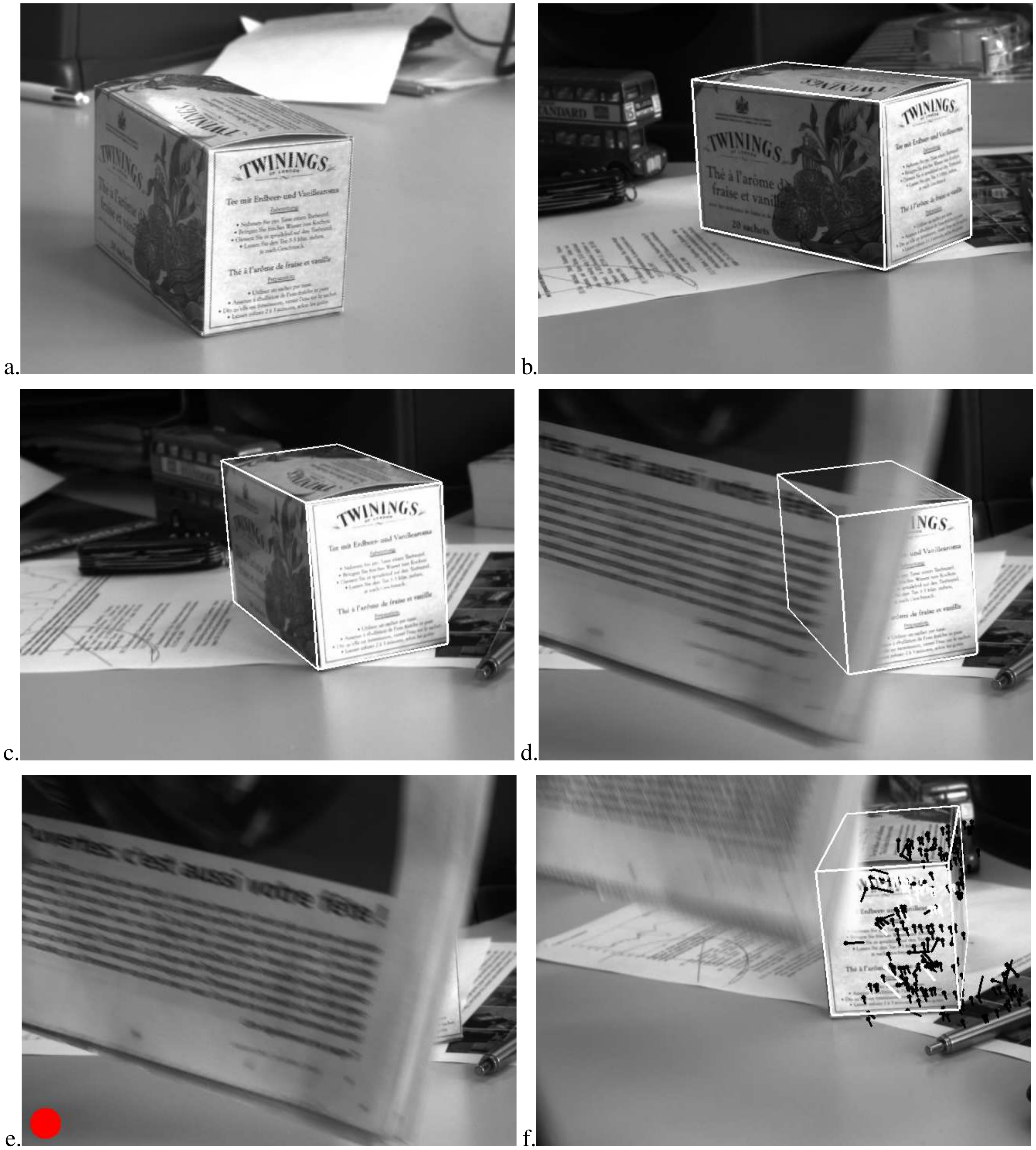

Figure 1: Automated pose estimation of a box to initialize a 3D tracker. (a): One of the training images; (b): An input image in which the box is automatically detected and its pose estimated; (c) and (d): Feature points are tracked to register the camera, even in case of partial occlusions; (e): The tracker fails because of a complete occlusion of the tracked object, the failure is detected and the initialization procedure is reinvoked; (f): This procedure recognizes the object when it reappears.

For local approaches to be effective, feature point extraction and characterization should be insensitive to viewpoint and illumination changes. Scale-invariant feature extraction can be achieved by using the Harris detector [Harris and Stephens, 1988] at several Gaussian derivative scales, or by considering local optima of pyramidal difference-of-Gaussian filters in scale-space [Lowe, 1999]. Mikolajczyck et al. [Mikolajczyk and Schmid, 2002] have also defined an affine invariant point detector to handle larger viewpoint changes, but it relies on an iterative estimation that would be too slow for our purposes.

Given the extracted feature points, various local descriptors have been proposed: Schmidt and Mohr [Schmid and Mohr, 1997] compute rotation invariant descriptors as functions of relatively high order image derivatives to achieve orientation invariance. Baumberg [Baumberg, 2000] uses a variant of the Fourier-Mellin transformation to achieve rotation invariance. He also gives an algorithm to remove stretch and skew and obtain an affine invariant characterization. Allezard et al. [Allezard *et al.*, 2000] represent the key point neighborhood by a hierarchical sampling, and rotation invariance is obtained by starting the circular sampling with respect to the gradient direction. Tuytelaars and al. [Tuytelaars and VanGool, 2000] fit an ellipse to the texture around local intensity extrema and use the Generalized Color Moments [Mindru *et al.*, 1999] to obtain correspondences remarkably robust to viewpoint changes. Lowe [Lowe, 2003] introduces a descriptor called SIFT based on several orientation histograms, that is not fully affine invariant but tolerates significant local deformations. This last descriptor has been shown in [Mikolajczik and Schmidt, 2003] to be one of the most efficient, which is why we compare our results against it in the Section 4.

In short, local approaches have been shown to work well on highly textured objects, to handle partial occlusions, and to tolerate errors in the correspondences. However, even if they can be used for object detection and pose estimation, they rely on relatively expensive point matching between a sample and an input image. By contrast, our approach is geared towards shifting much of the computational burden to a training phase during which we build descriptors from the set of sample images and, as a result, reducing the cost of online matching while increasing its robustness.

3 Feature Point Matching as a Classification Problem

3.1 Approach

Matching interest points found in an input image against feature points on a target object $\mathbf{O}$ can be naturally formulated as a classification problem as follows. During training, we construct a set $\mathbf{F_O}$ of N prominent feature points lying on $\mathbf{O}$. Given an input patch $\mathbf{p} \in \mathbf{P}$, the space of all image patches of a given size, we want to decide whether or not it can be an image of one of the N interest points. In other words, we want to assign to $\mathbf{p}$ a class label in $Y(\mathbf{p}) \in \mathbf{C} = \{-1, 1, 2, \ldots, N\}$, where the –1 label denotes all the points that do not belong to the object — the background. Y cannot be directly observed, but our goal is to construct a classifier

$$\hat{Y} : \mathbf{P} \rightarrow \mathbf{C}$$

such as $P(Y \neq \hat{Y})$ is small.

In other recognitions tasks, such as face or character recognition, large training sets of labeled data are usually available. However, for automated pose estimation, it would be impractical to require very large number of sample images. Instead, to achieve robustness with respect to pose and illumination changes, we use a small number of images and synthesize many new views of the feature points in $\mathbf{F_O}$ using simple rendering techniques to train our classifier.

For each feature point, this constitutes a sampling of its *view set*, that is the set of all its possible appearances under different viewing conditions. We can then use statistical classification techniques to describe them compactly and, finally, use these descriptors to perform the actual classification at run-time. This gives us a set of matches that lets us to estimate the pose.

3.2 Creating View Sets

Constructing the viewset of points is relatively simple, and we focus here on some of the implementation details that ensure invariance to illumination changes and also robustness to point localization error that can occur while extracting the feature points.

3.2.1 Construction Under Local Planarity Assumptions

For a given point in the training image, if the surface can be assumed to be locally planar, a new view of its neighborhood can be synthesized by warping using an affine transformation, that approximates the actual homography:

$$(\mathbf{n} - \mathbf{n}_0) = \mathbf{A}\,(\mathbf{m} - \mathbf{m}_0) + \mathbf{t} \tag{1}$$

where $\mathbf{m}_0$ are the coordinates of the keypoint detected in the training image, $\mathbf{n}_0$ are the coordinates of the patch center, and $\mathbf{n}$ the new coordinates of the warped point $\mathbf{m}$. The matrix $\mathbf{A}$ can be decomposed as: $\mathbf{A} = \mathbf{R}_\theta \mathbf{R}_\phi^{-1} \mathbf{S} \mathbf{R}_\phi$, where $\mathbf{R}_\theta$ and $\mathbf{R}_\phi$ are two rotation matrices respectively parameterized by the angles θ and ϕ, and $\mathbf{S} = \mathrm{diag}\,[\lambda_1, \lambda_2]$ is a scaling matrix; $\mathbf{t} = [t_u, t_v]^t$ is a 2D translation vector [Hartley and Zisserman, 2000].

The view set is created by generating the views corresponding to a regular sampling of the space of the $(\theta, \phi, \lambda_1, \lambda_2, t_u, t_v)$ parameters. As discussed below, we use non null values of $\mathbf{t}$ to handle possible localization error of the keypoints.

3.2.2 Robustness To Localization Error

When a keypoint is detected in two different images, its precise location may shift a bit due to image noise or viewpoint changes. In practice, such a positional shift results in large errors of direct cross-correlation measures. One solution is to iteratively refine the point localization [Mikolajczyk and Schmid, 2002]. The keypoint descriptor in [Lowe, 1999] handles this problem by carefully assuring that a gradient vector contributes to the same local histogram even in case of small positional shifts.

In our case, we simply allow the translation vector $\mathbf{t}$ of the affine transformation of Equation 1 to vary in the range of few pixels when generating the view sets. These small shift corresponds to the noise that arises in corner detection.

3.2.3 Invariance To Illumination Changes

Handling illumination changes can be done by normalizing the views intensities. After experimenting with many normalization techniques, we concluded that scaling the views intensities so that all the views have the same minimum and maximum intensity values is both cheap and effective as shown in Figure 2: This has the advantage of emphazing the view contrast. Because the normalization is performed independently on each keypoint at run-time, it handles correctly complex illumination changes.

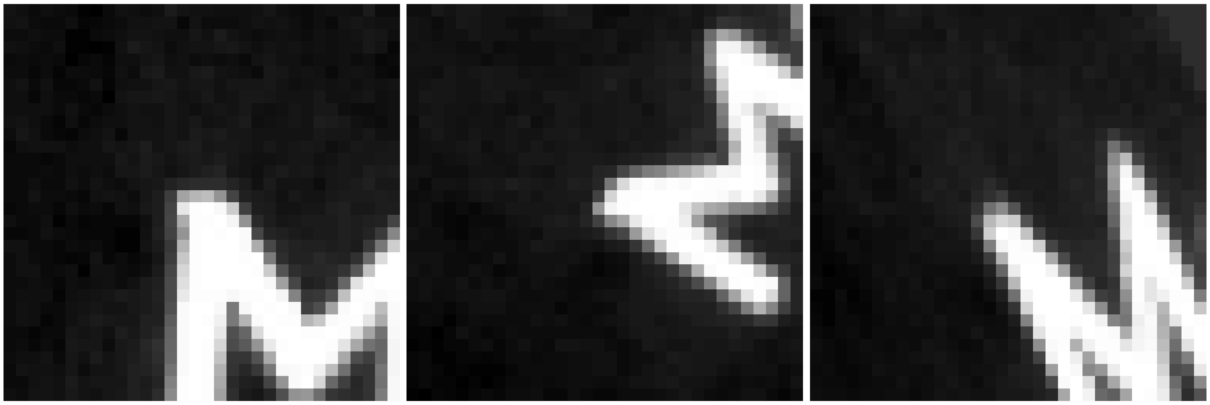

Figure 2: The patch around a keypoint detected in the training image of a book cover, and two patches synthesized using random affine transformations.

3.2.4 Relaxing the Planarity Assumptions

In our target applications such as initialization of model-based 3D tracking, an object model is available. It may be a precise one, as in the case of the box of Figure 1, or a generic one as in the case of faces that we use to illustrate our technique. Such a model is very useful to capture complex appearance changes due to changes in pose of a non convex 3D object, including occlusions and non-affine warping. For example, as shown in Figure 3, we generate several views of a keypoint on the left side of the nose by texture mapping the face in several positions.

This approach also lets us merge the information from several training images in a natural way: The generated viewsets can simply be combined when they correspond to the same 3D point as depicted in Figure 4.

3.2.5 What Do View Sets Look Like ?

Each view can be represented by a 1D column vector in the image space by ordering brightness values in raster scan order. Figure 5 represents the evolution of the view sets for three keypoints as the range of the $(\theta, \phi, \lambda_1, \lambda_2, t_u, t_v)$ parameters is progressively increased.

The viewsets are not convex and the original view is not necessarily close to their centroid. However, they tend to remain compact as long as the range of transformation parameters is not too big. This is why we can formulate wide baseline matching as a classification problem.

3.3 Classification

The classifier must be carefully designed for our problem. Even if we eventually use a robust estimator such as RANSAC to eliminate outliers, the mis-classification rate should be low. A high rate would slow down the robust estimator and the advantage of our fast matching procedure would be lost.

The classifier should also be able to retrieve a sufficient number of points to estimate the object pose even in presence of partial occlusion. On the other hand, not all the keypoints detected during the training stage have to be considered: If some keypoints appear not to be characteristic enough to be matched reliably, it is better to ignore them to reduce the risk of mis-classification.

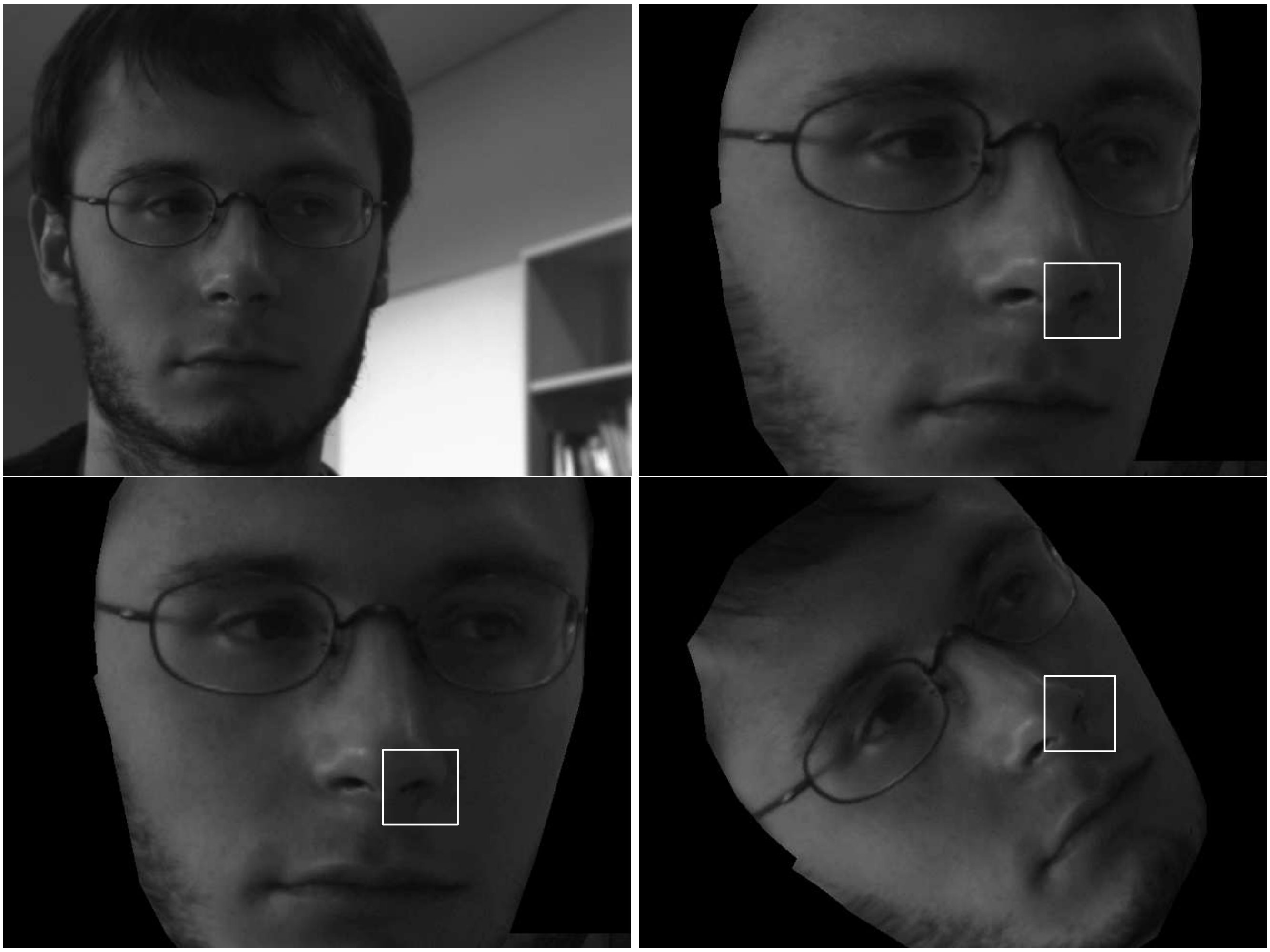

Figure 3: Three synthetic views of a human face, generated from the original image on the top left. The patches extracted from these images to build a viewset of a keypoint on the nose are represented.

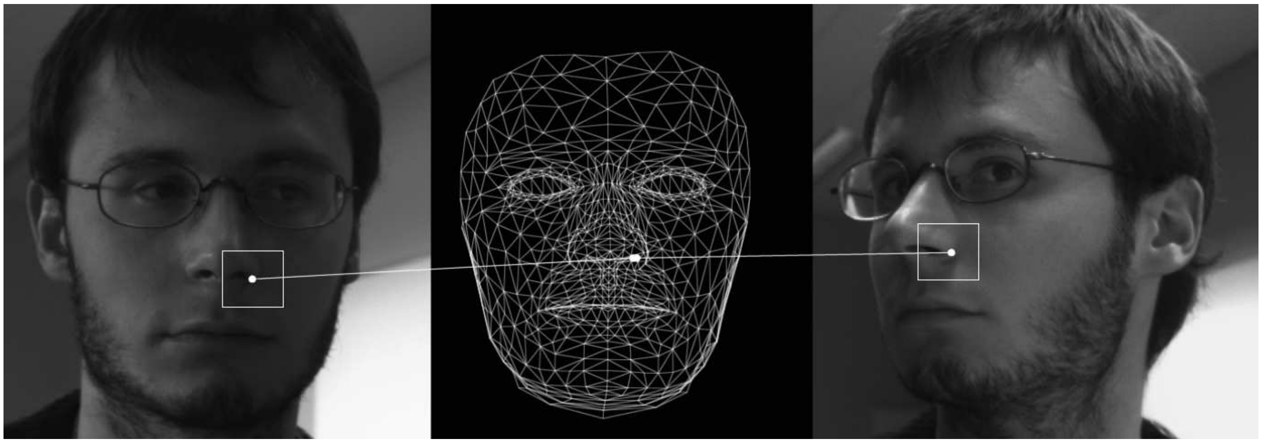

Figure 4: Using two different training images to build the viewset of the same keypoint.

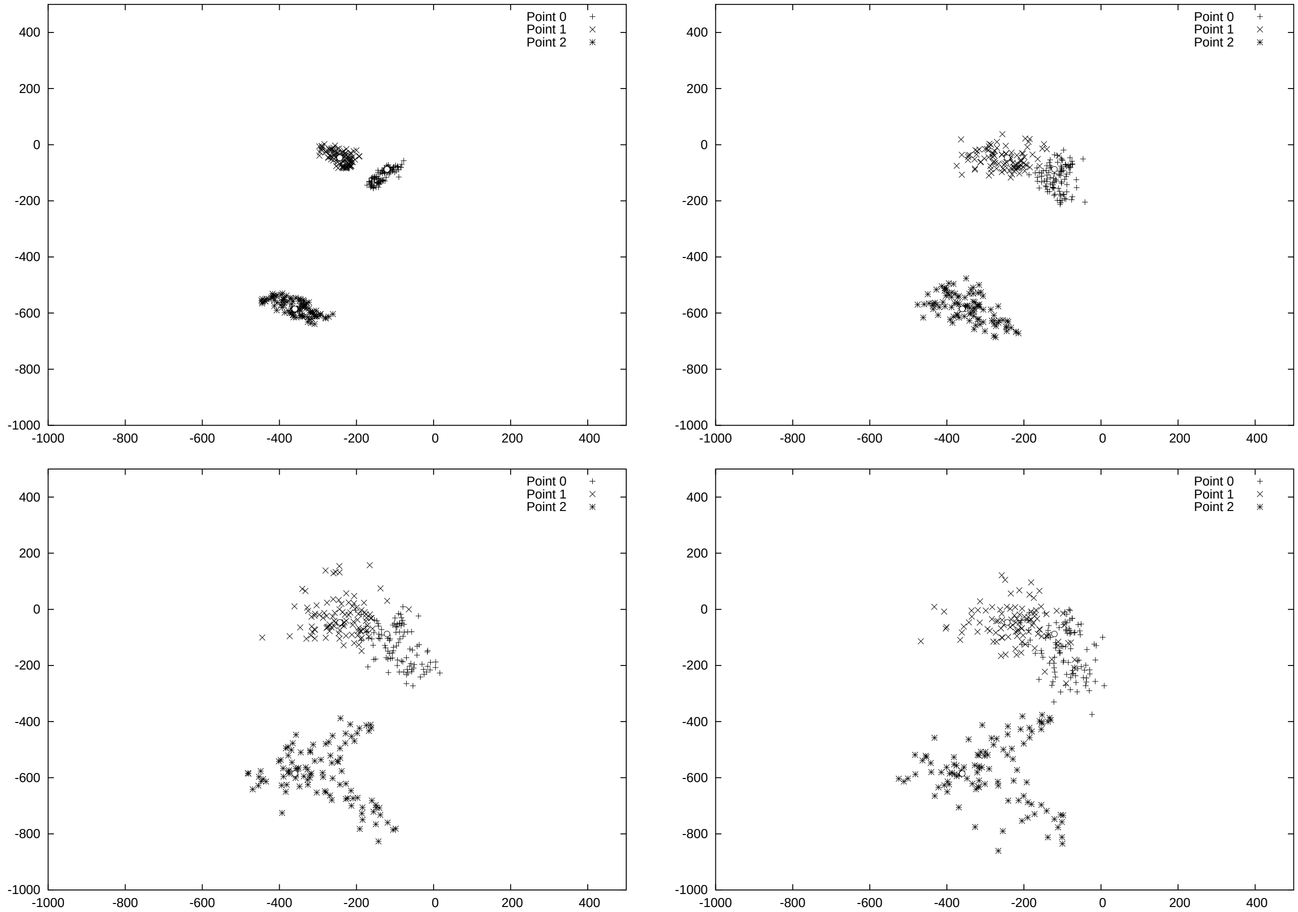

Figure 5: Projections on a 2D plane of the viewsets of three feature points detected on the book cover of Figure 9. The viewsets are composed of one hundred samples each. They have been generated using affine deformations, and we increase the range of the $(\theta, \phi, \lambda_1, \lambda_2, t_u, t_v)$ parameters in Equation 1, from top left to bottom right. The range for θ varies from $(-10; 10)$ to $(-50; 50)$ degrees, for λ_1 and λ_2 from $(0.9; 1.1)$ to $(0.4; 1.6)$, and t_u and t_v from 0 to 2 pixels. The range of ϕ is always $(-180; 180)$ degrees.

Classification should also be performed sufficiently fast for interactive applications. To achieve all these requirements, we perform this task as follows.

First, the viewsets are computed for object points detected in one or more training images. Optionnaly, we can create a background class by taking patches around points detected in images of typical background. This step is not necessary but helps to deal with cluttered background.

To reduce the dimensionality, we perform a Principal Component Analysis on the set of patches. This is followed by K-mean estimation on each viewset independently to handle its potentially complex shape while compacting the viewset representation. In practice, we compute 20 means per viewset. Figure 6 shows some examples of computed means for the keypoint of Figure 2.

Then the classifier can attribute a class to an input patch by a Nearest Neighbor search through the set of means and background points. The keypoints most likely to lead to mis-classification can be found during training, by estimating $P(Y \neq \hat{Y} \mid c)$ from the training set. When it is above a given threshold (say 10%) for a class, this class is removed from the class set $\mathbf{C}$. This way, we keep the more characteristic object points.

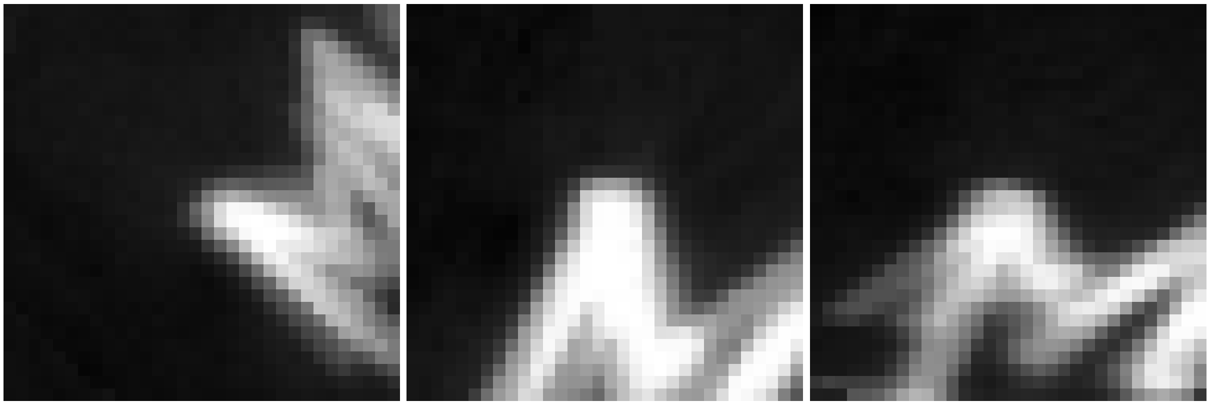

Figure 6: Some of the K-means computed for the viewset of keypoint shown Figure 2.

Finally we build an efficient data structure [Nayar *et al.*, 1996] to perform an efficient run time Nearest Neighbor search in the eigen space computed by the PCA.

3.4 Why Our Approach Is Fast

First, our method does not involve any time consuming pretreatment such as orientation or affine transformation extraction. It is made possible by matching the input points against the compact representation of the viewsets, that contain the possible views under such transformations.

Next the eigen space allows to reduce the dimensionality of each input patch, with negligible loss of information: The eigen images computed by the PCA can be seen as a filter bank like in previous methods, but they are *specialized* for the object to detect, since they are directly computed on the training set.

Finally, the PCA lets us to build an efficient data structure for fast Nearest Neighbor search at run-time, since it sorts dimensions by order of importance.

4 Results

4.1 Experiments on a Planar object

Figure 9 depicts a first set of experiments on a planar object. About 200 key points have been detected in the training image using the Harris detector. For each we constructed a viewset made of 100 samples, synthesized from random affine transformations with $\theta \in (-50; 50)$ degrees, $\phi \in (-180; 180)$ degrees, $\lambda \in (0.4; 1.6)$, $t \in (-2; 2)$ pixels. The original size of the patches is 32×32 pixels, it is then by two to reduce the computation cost. We keep the first 20 dimensions given by the PCA. 1000 feature points have been detected in the input images, and matched against the learned features, which lets us robustly estimate an homography between the two sets. The input images present perspective, intensity, scale and orientation changes, and the object pose is nevertheless correctly recovered. The whole process takes about 200 milliseconds on a 2GHz PC.

We compared our results with those obtained using the executable that implements the SIFT method [Lowe, 2003] and kindly provided by David Lowe. Our method usually gives a little fewer matches, and has a little higher outlier rate (about 10% of about 5%). Nevertheless, it is largely enough to accurately estimate the object pose after a few tens of RANSAC samples, and it runs faster: About 0.2 seconds for our non-optimized executable against 1 second for the Lowe executable on the same machine.

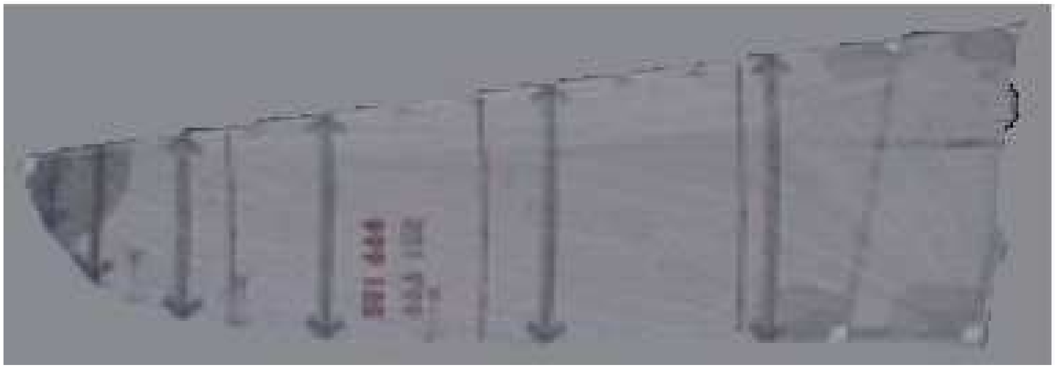

Figure 7: The model used to detect the sail of Figure 8

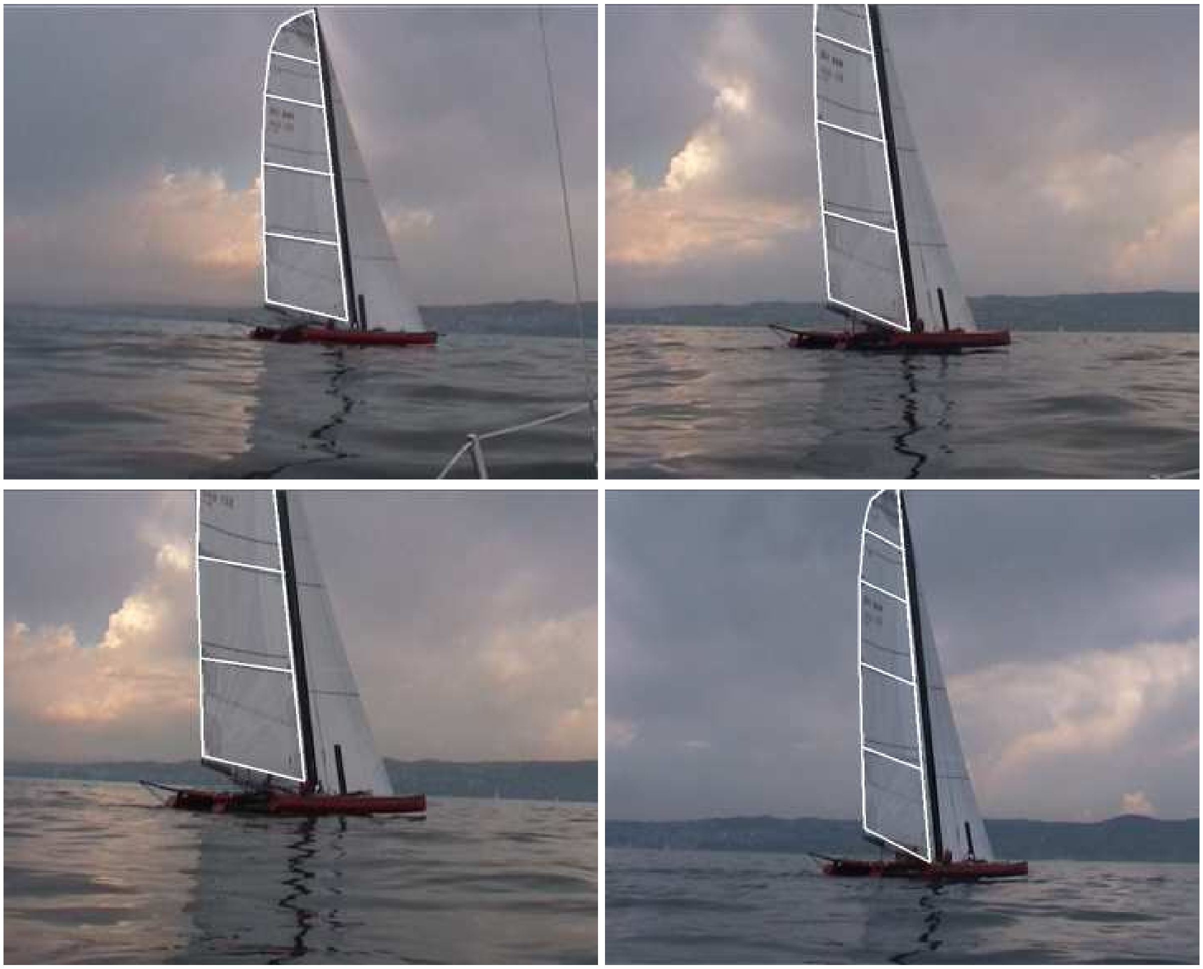

Figure 8: Detecting a sail. We acquired a 4000 thousand frame video using a home camcorder during a regatta. Even though the camera jerks, the zoom changes and the lighting conditions are poor, the sail is detected in all frames, such as those shown above, where a sufficient portion of the sail is seen.

4.2 Detecting a Sail

We applied our method for the detection of a sail over a 4000 thousand frame video taken with a home camcorder during a regatta. Despite the bad conditions: the sail is not well textured as it is shown Figure 7, it moves in and

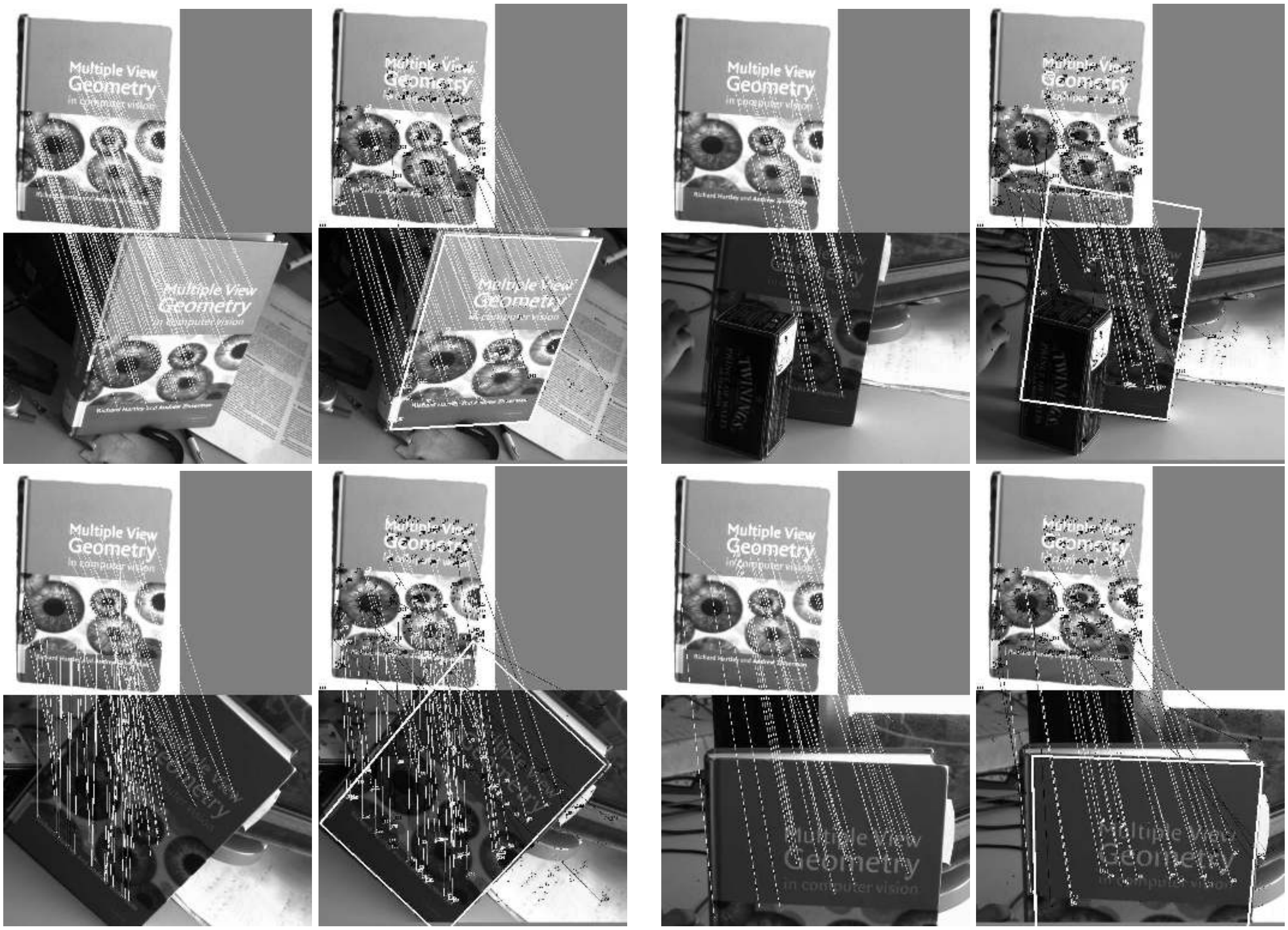

Figure 9: Comparison between SIFT method (left image) and ours (right) for a planar object. Our method usually gives a little less matches, and has a little higher outlier rate. Nevertheless, it is largely enough to accurately estimate the object pose after a few tens of RANSAC samples, with a lower computation cost than SIFT method.

out of the field of view, the camera motions are very jerky and the illumination changes all the time, the sail is detected in all frames, such as those shown Figure 8, where a sufficient portion of the sail is seen.

4.3 3D Object Pose Estimation

In the case of a 3D object, the full pose is recovered from the matches using the POSIT algorithm [DeMenthon and Davis, 1992] in tandem with RANSAC.

A Simple Box The method successfully recovers the pose of a textured box disposed in almost any position, using around six very different views for training. Some examples are presented Figure 1. Because the texture is not symmetric, the pose is never ambiguous and can be found easily. In practice, the pose is recovered in less than a second.

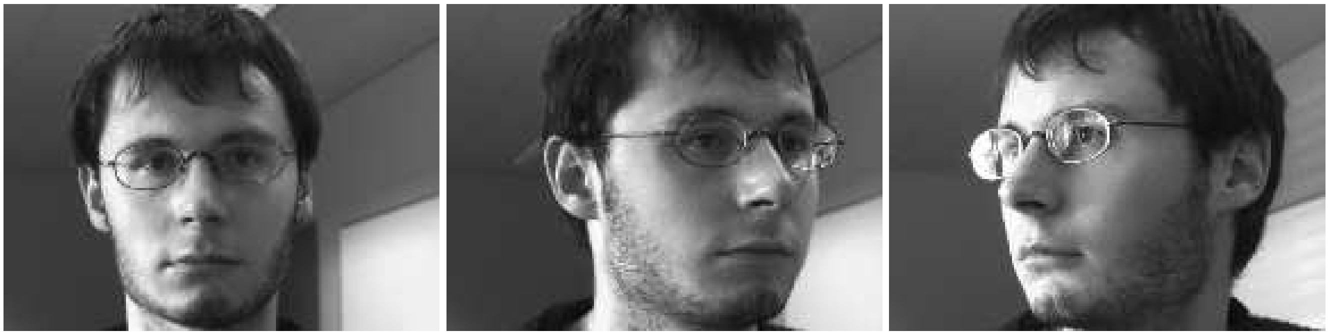

Figure 10: The three training images for face pose estimation in Figure 11. They have been registered manually. The 3D model (not shown) is a generic human face model, and does not include the glasses of the subject.

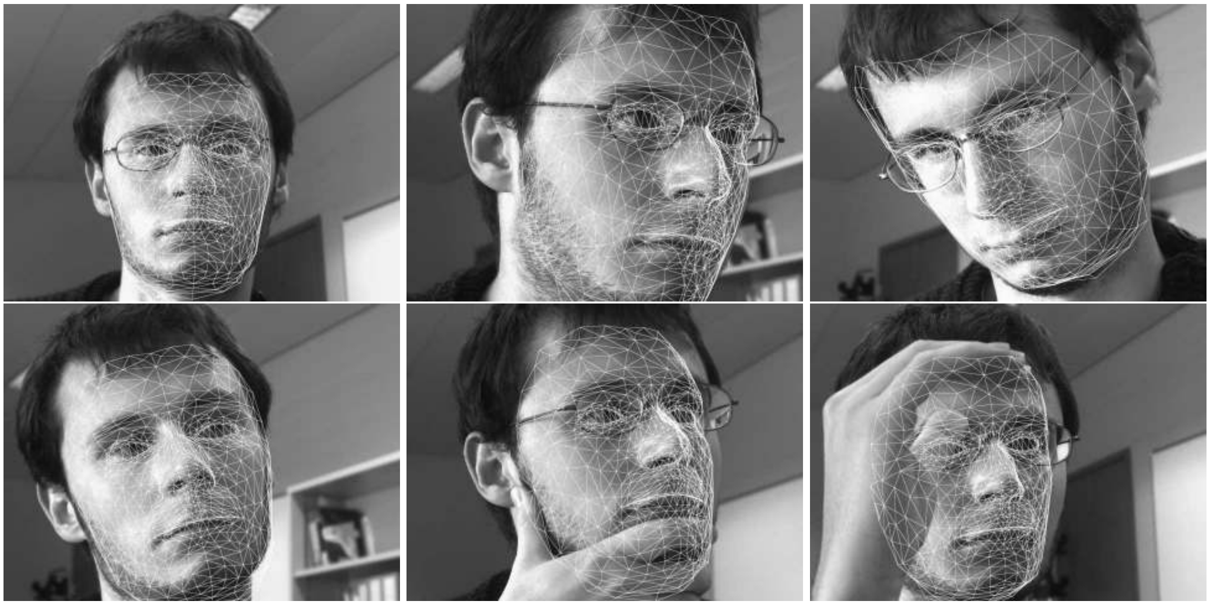

Figure 11: Face pose estimation. First row: recovered pose under normal conditions. Second row: recovered pose under more difficult conditions: the subject removed his glasses in the first image, and the face is partially occluded in the two last images.

A Human Face We applied the same method to a human face, using the 3 training images of Figure 10 and a generic 3D face model. The training images have been registered by hand, by moving the 3D model to the right location. Even if the 3D model is far from perfect — it does not have glasses and its shape does not match exactly — we are able to recover the pose of a specific human face under both illumination changes and partial occlusions as depicted by Figure 11.

Working on faces is much harder than on a textured box because faces provide far fewer feature points and their 3D nature produces complex appearance changes. Nevertheless, only three training images where is enough to recover the poses shown in Figure 11. The process is robust enough to support some occlusion and still work if the subject removes its glasses. It takes around one second. We tried the SIFT method on these face images and others, with little success. After consultation with David Lowe we suspect that this might be because of the face non-planarity and the light changes. These could possibly be addressed by having a larger training set of faces under different orientations and lighting, but our approach requires only a small training set, which is desirable.

5 Conclusion and Perspectives

We proposed an approach to point matching for object pose estimation based on classification. It runs to be faster than previous methods in the planar case, and still works for the non-planar case, contrary to these methods.

Our approach is also very general, and lets us relax the locally planar assumption. In fact, it has the potential to recognize complex shaped textured objects, under large viewpoint and illumination changes even with specular materials, assuming we can generate images of the object under such changes. This is a realistic assumption since there are many Computer Graphics methods designed for this purpose, which opens new avenues of research.

We expect that allowing more complex appearance changes than the ones we have dealing with so far will result in the view-sets becoming more difficult to separate. We therefore intend to replace the simple classification methods we have described here by more sophisticated one that can deal with the facts that each class can exhibit huge variability and thus requires a large number of samples and interest points can belong to several classes or to none at all. We will therefore investigate the use of decision trees [Breiman *et al.*, 1984, Amit and Geman, 1997, Amit *et al.*, 1997], which we believe to be most suitable for our specific problem because they naturally handle multi-class classification, can be enhanced by using powerful statistical methods such as bagging [Breiman, 1996] and boosting [Freund and Schapire, 1996, Mason *et al.*, 2000]. These methods are very fast, and achieve very good recognition rate, usually more than 99%. Thus, we believe that our approach is an important step toward much better object recognition and detection methods.

References

[Allezard *et al.*, 2000] N. Allezard, M. Dhome, and F. Jurie. Recognition of 3d textured objects by mixing view-based and model-based representations. In *International Conference on Pattern Recognition*, pages 960–963, Barcelona, Spain, Sep 2000.

[Amit and Geman, 1997] Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997.

[Amit *et al.*, 1997] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1300–1306, 1997.

[Baumberg, 2000] Adam Baumberg. Reliable feature matching across widely separated views. In *Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000.

[Breiman *et al.*, 1984] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

[Breiman, 1996] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[DeMenthon and Davis, 1992] D. DeMenthon and L. S. Davis. Model-based object pose in 25 lines of code. In *European Conference on Computer Vision*, pages 335–343, 1992.

[Freund and Schapire, 1996] Y. Freund and R.E. Schapire. Experiments with a New Boosting Algorithm. In *International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[Harris and Stephens, 1988] C.G. Harris and M.J. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference, Manchester*, 1988.

[Hartley and Zisserman, 2000] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[Jurie, 1999] F. Jurie. Solution of the Simultaneous Pose and Correspondence Problem Using Gaussian Error Model. *Computer Vision and Image Understanding*, 73(3):357–373, 1999.

[Lowe, 1999] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.

[Lowe, 2003] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2003.

[Mason *et al.*, 2000] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Neural Information Processing Systems*, pages 512–518. MIT Press, 2000.

[Mikolajczik and Schmidt, 2003] K. Mikolajczik and C. Schmidt. A performance evaluation of local descriptors. In *Conference on Computer Vision and Pattern Recognition*, pages 257–263, June 2003.

[Mikolajczyk and Schmid, 2002] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142. Springer, 2002. Copenhagen.

[Mindru *et al.*, 1999] F. Mindru, T. Moons, and L. VanGool. Recognizing color patterns irrespective of viewpoint and illumination. In *Conference on Computer Vision and Pattern Recognition*, pages 368–373, 1999.

[Nayar *et al.*, 1996] Shree K. Nayar, Sameer A. Nene, and Hiroshi Murase. Real-Time 100 Object Recognition System. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1186–1198, 1996.

[Schaffalitzky and Zisserman, 2002] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or ”How do I organize my holiday snaps?”. In *Proceedings of European Conference on Computer Vision*, pages 414–431, 2002.

[Schmid and Mohr, 1997] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, May 1997.

[Tuytelaars and VanGool, 2000] T. Tuytelaars and L. VanGool. Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions. In *British Machine Vision Conference*, pages 412–422, 2000.

[Viola and Jones, 2001] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.