Joint Pose Estimator and Feature Learning for Object Detection

Karim Ali[1,2], François Fleuret[1,3], David Hasler[2], Pascal Fua[1]

[1] École Polytechnique Fédérale de Lausanne (EPFL), CVLAB, Switzerland
[2] Swiss Center for Electronic and Microtechnology, Switzerland
[3] Idiap Research Institute, Switzerland
{karim.ali,pascal.fua}@epfl.ch, fleuret@idiap.ch, david.hasler@csem.ch

Abstract

A new learning strategy for object detection is presented. The proposed scheme forgoes the need to train a collection of detectors dedicated to homogeneous families of poses, and instead learns a single classifier that has the inherent ability to deform based on the signal of interest.

Specifically, we train a detector with a standard AdaBoost procedure by using combinations of pose-indexed features and pose estimators instead of the usual image features. This allows the learning process to select and combine various estimates of the pose with features able to implicitly compensate for variations in pose. We demonstrate that a detector built in such a manner provides noticeable gains on two hand video sequences and analyze the performance of our detector as these data sets are synthetically enriched in pose while not increased in size.

1. Preamble

Machine-learning object detection techniques rely on searching for the presence of the target over all scales and locations of a scene. In order to handle complex cases where latent variables modulate changes in appearance, for instance due to rotation or variation in illumination, two strategies have emerged: either building a collection of pose-dedicated classifiers or explicitly visiting the additional latent variables in the same manner as one explores location and scale.

We propose a new approach which consists of designing a family of pose estimators able to compute meaningful values for the additional latent variables directly from the signal. We allow the learning procedure to automatically handle the trade-offs involved in selecting and combining estimates of the hidden parameters obtained from various image areas. This approach sets forth a framework that overcomes both the data fragmentation problem, associated with the training of pose-dedicated classifiers, as well as the labeling and computational overheads of pure pose-indexed methods.

We rely on the AdaBoost algorithm, a simple and efficient learning method that provides reliable detection in real-time [2, 12]. Our key contribution lies in augmenting a set of pose-indexed features with a *family* of pose estimators. Each feature then consists of a pair of functionals: one functional to estimate the pose and the other to compute a feature *indexed* in both location and type with respect to the estimated pose. The AdaBoost learning procedure is allowed complete freedom in deciding how best to combine a pose estimator with a pose-indexed feature. In this manner, training proceeds on the unpartitioned data set while pose estimator learning and feature learning occur jointly in a fully integrated framework. The final detector consists of a variety of features which can deform analytically and independently based on the signal of interest.

This work is motivated by a practical application – the detection of hands to prevent injuries in manufacturing plants – which naturally poses significant challenges. The appearance of the hand, a deformable articulate object, may change considerably and to be of practical interest, detection must proceed in real-time with nearly zero error rates. The pose space can be very rich and so it is the aim of our method to meaningfully combine the various pose estimates induced from training. We demonstrate that our framework provides substantial performance gains in this setting.

We begin this paper with a review and synthesis of related works on multi-view object detection. Next, our proposed framework is introduced and studied. Experimental results are given in the following section and finally, we conclude with a brief discussion.

This work has been supported in part by the Swiss National Science Foundation under project 200021-117997, by the Fond Québécois de recherche sur la nature et les technologies, and by the Swiss National Science Foundation under the National Centre of Competence in Research on Interactive Multimodal Information Management.

2. Related Work

Initial attempts Early efforts aimed at tackling the multi-view object detection problem focused on in-plane rotations. For example, in [6], the authors propose a method for detecting frontal in-plane rotated faces. Two neural network classifiers, one to estimate the pose and the other to detect up-right faces are trained. For each image of interest, the pose of the face is first estimated and used to de-rotate the sample. The image is subsequently classified using the second detector.

Partitioning the pose space The authors in [11] extend the Viola-Jones detector to address two types of pose variation concerning faces: in-plane rotations and out-of-plane rotations. To deal with in-plane rotations, the pose of the image of interest is estimated using a decision tree constructed to determine the view-point class. Second, one of twelve rotation-specific Viola-Jones detectors is used to classify the image. The treatment of out-of-plane rotations is entirely analogous.

A number of other recent works essentially devise the same strategy in dealing with multi-view object detection [4, 8, 15, 3, 5]. Multiple detectors, each specialized to a specific pose, are built and the pose is estimated as part of a first stage. Though these techniques offer reliable detection performance, they are nevertheless burdened by the need to partition the pose space and to train several pose dedicated classifiers. Dealing with a rich pose space or a finer partition of the pose space is clearly not possible using such a strategy both in terms of training time and population size requirement.

The work of [9, 10] also employs pose dedicated classifiers to detect and estimate three-dimensional hand pose. One notable difference is that pose estimation and detection are organized hierarchically within a tree based system. In this method, each leaf of the tree corresponds to a specific hand pose and the intermediary nodes serve the dual purpose of early rejection as well as gross pose estimation. Each level of the tree gradually refines the pose estimate by the use of more constrained pose dedicated classifiers. We note that this works suffers from the same problem of partitioning the pose space.

Yuan et al. [14] proposed an interesting approach in which detection and pose estimation are learned jointly with a Support Vector Machine. A multiplicative form of two kernel functions is introduced: whereas one kernel measures the within-class similarity of the foreground class (pose estimation), the other one is dedicated to foreground-background classification (detection). Though training proceeds on the entire unpartitioned data set, this method implicitly partitions the training data, albeit in a fuzzy fashion, as features are learned on clusters of similar samples.

Unifying the pose space In order to overcome training data fragmentation the authors in [1] present a framework centered on pose-indexed features. Classifiers are arranged in a hierarchy based on a nested partition of the pose space. At each *level* of the hierarchy, a *single* classifier is induced from the entire data set. All classifiers at a given level are analytically derived from this base predictor with the use of the pose-indexed features.

Though promising results are shown, this technique requires nonetheless the training data to be labelled with the corresponding ground truth. In addition, at each level of the hierarchy a brute force search of the pose space embodied by the latter is required in testing.

In contrast, our method requires no labeling for training and no brute force search during testing. Our framework instead allows for the implicit discovery of pose as explained next.

3. Methodology

Discriminative single-view learning techniques rely on training a classifier for a single scale and location. Detection on the other hand is managed by searching for the presence of the target over all the scales and locations of a given scene. Parsing the scene is such a manner, while deforming the base detector in terms of both scaling and translation, is based upon an *assumption of invariance* to base predictor transformations. In particular, it is assumed that the response of a constituent feature of the base predictor is invariant to scaling and translation given that a target is present. In [1], the authors simply reformulate this well understood classical learning structure as a special-case of so-called *pose-indexing* and extend the idea to richer more complex poses. Specifically, given that a target is present with a certain pose, the distribution of the response of a pose-indexed feature is independent of the given pose, as illustrated in Figure 1.

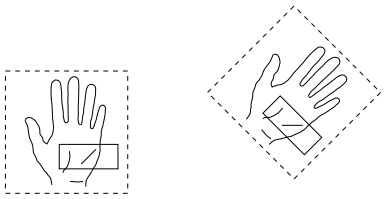

Figure 1. Depiction of the consistency between the object pose and the definition of a pose-indexed feature on two hand instances. The pose space Θ is three-dimensional, encoding the location and the orientation of the target. The dash lines show two poses θ for which the feature is computed, and the full lines show the areas over which edges are summed to compute the actual feature response in these two cases.

The following reformulates mathematically the elements

of the previous paragraph in the context of the AdaBoost learning procedure. Formal presentations of both standard features and pose-indexed features are given followed by a detailed description of our framework. We underline the fact that our developments are not contingent on the use of AdaBoost: one could use our pose-estimator based features with other discriminative machine learning methods such as Support Vector Machines, decision trees or even with generative models.

3.1. Boosting with Standard Image Features

Let

$$\mathcal{I} = [0,1]^{W \times H}, \tag{1}$$

denote the space of gray scale images of size $W \times H$ and let

$$(X_i, Y_i) \in \mathcal{I} \times \{-1, 1\}, \quad i = 1, \ldots, T, \tag{2}$$

denote a labelled training set.

Given a set $\mathcal{F}$ of image features or mappings of the form

$$h : \mathcal{I} \to \mathbb{R}, \tag{3}$$

a standard AdaBoost procedure constructs a *strong* classifier f as a linear combination of thresholded features or *stumps* of the following form

$$\forall x \in \mathcal{I}, \quad f(x) = \sum_{k=0}^{K} \omega_k \mathbf{1}_{\{h_k(x) \geq \rho_k\}}, \tag{4}$$

where K is the number of stumps, and

$$\forall k, \ \ (\omega_k, h_k, \rho_k) \in \mathbb{R} \times \mathcal{F} \times \mathbb{R}. \tag{5}$$

Here, prior knowledge of the signal is embedded in the choice of the feature set $\mathcal{F}$. For instance, invariance to changes in illumination may be obtained by using edge detectors while invariance to translation may be achieved by using color or gray-scale histograms estimated over large areas.

3.2. Pose-indexed Image Features

Recent works in object detection have introduced the idea of pose-indexed image features. The response of these features depends not only on the image, but also on a candidate pose in order to probe for the presence of the object. Formally, a pose-indexed feature is of the form

$$g : \Theta \times \mathcal{I} \to \mathbb{R}, \tag{6}$$

where Θ is the pose space of the object. In that context, a training set takes the form

$$(X_i, \theta_i, Y_i) \in \mathcal{I} \times \Theta \times \{-1, 1\}, \quad i = 1, \ldots, T, \tag{7}$$

where Y_i is equal to $+1$ if a target is truly visible in X_i with pose θ_i, and to -1 otherwise. Ideally the pairs (X_i, θ_i) visit all the possible combinations of scenes and poses. For instance, a scene X containing a single visible target would appear in $\|\Theta\|$ pairs in such a set, once with label $+1$ and $\|\Theta\| - 1$ times with label -1.

With a set $\mathcal{P}$ of pose-indexed feature, one can construct a pose-indexed classifier of the form

$$\forall \theta \in \Theta, \ x \in \mathcal{I}, \ \ f(\theta, x) = \sum_{k=0}^{K} \omega_k \mathbf{1}_{\{g_k(\theta, x) \geq \rho_k\}}. \tag{8}$$

Classical object-detection at fixed scale, where the scene is parsed at every location, can be formalized in this setting with a two dimensional pose space

$$\Theta = [0, W] \times [0, H]. \tag{9}$$

Given an image x, detection at a particular threshold T consists of computing a list of alarms

$$\mathbf{A}_T(x) = \Big\{ \theta \in \Theta \ \text{ s.t. } f(\theta, x) \geq T \Big\}. \tag{10}$$

This approach extends naturally to arbitrary complex object pose while neutralizing the nuisance hidden variables and maintaining the joint information between different features. However, it requires the training data to be labelled with the corresponding ground truth, and requires the exploration of pose parameters in test. These drawbacks are further exacerbated by adding more dimensions to the pose space.

3.3. Proposed Framework: Pose Estimators

To retain the benefits of the pose-indexed features without their inherent weaknesses, we introduce the idea of a pose estimator, which computes a meaningful pose directly from the signal in order to parametrize the features. Specifically, given a decomposition of Θ into the product of two pose spaces

$$\Theta = \Theta_1 \times \Theta_2, \tag{11}$$

a pose estimator is a mapping of the form

$$\eta : \Theta_1 \times \mathcal{I} \to \Theta_2. \tag{12}$$

In practice, $\Theta_1 = [0, W] \times [0, H]$ is the aforementioned two-dimensional space standing for the location of the target, while $\Theta_2 = [-\pi, \pi[$ consists of an orientation in the image plane. Indeed, though our framework rids us of all other sources of intra-class variation, it still maintains the classical structure of searching through all locations.

Given a pose-indexed feature

$$g : (\Theta_1 \times \Theta_2) \times \mathcal{I} \to \mathbb{R}, \tag{13}$$

and a pose-estimator η, we can now define a pose-indexed image feature γ for poses in the pose space Θ_1 with

$$\forall \theta \in \Theta_1,\, x \in \mathcal{I},\;\; \gamma(\theta, x) = g\big((\theta, \eta(\theta, x)), x\big), \qquad (14)$$

where it is recalled that $\eta(\theta, x) \in \Theta_2$.

In practice, to evaluate such a functional γ on a scene x for a location $\theta \in \Theta_1$, we first computes an angle $\theta' = \eta(\theta, x) \in \Theta_2$, and then evaluate g on x for the combined pose $(\theta, \theta') \in \Theta$. In words, these features estimate the pose in the image plane and are still parameterized by the classical pose space of location, as is the case for the standard features.

Finally, from a set $\mathcal{P}$ of pose-indexed features based on poses in $\Theta = \Theta_1 \times \Theta_2$ and a family of pose-estimators $\mathcal{E}$ that predict a pose in Θ_2 from a location in Θ_1, we can construct a set $\mathcal{P}_1$ of pose-indexed features for poses in Θ_1 by considering all the possible combinations as follows:

$$\mathcal{P}_1 = \Big\{(\theta, x) \mapsto g\big((\theta, \eta(\theta, x)), x\big),\, g \in \mathcal{P},\, \eta \in \mathcal{E}\Big\}. \quad (15)$$

This augmented family $\mathcal{P}_1$ can then be used with AdaBoost in a straightforward manner. At every iteration, the most successful pose estimator and pose indexed pair is chosen with the next pair chosen so as to rectify the errors of the previous one resulting in a boosted ensemble. Pose estimator learning and feature learning occurs jointly in a fully integrated fashion: various pose parameter estimates are combined to reduce classification error. The final detector is flexible and able to simultaneously examine the signal in $|\mathcal{E}|$ different ways to determine pose parameters and deform its features accordingly.

4. Implementation Details

The specifics of our implementation are given in this section. We follow the same notation as that of §3.

4.1. Standard Feature Set

We describe here a family of standard image features, not yet indexed by a pose. A scene x is preprocessed[1] by computing and thresholding the derivatives of the image intensity to obtain an edge image. The orientation of these edges are further quantized into eight bins, resulting in eight edge maps, see Fig. 2.

Let $\Psi = \{0, \pi/4, 2\pi/4, \ldots, 7\pi/4\}$ denote the possible orientations of a quantized edge, and $\forall e \in \Psi$, $x \in \mathcal{I}$, $l \in \{1, \ldots, W\} \times \{1, \ldots, H\}$, let

$$\xi_e(x, l) \in \{0,\, 1\}, \qquad (16)$$

[1]As noted in §5, one of our data set is directly obtained from a hardware-specialzed camera [7] and requires no such processing.

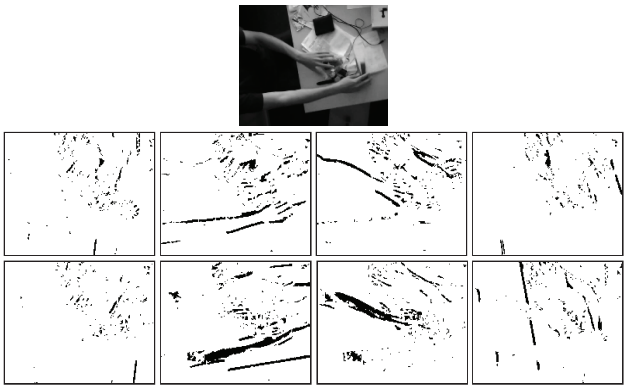

Figure 2. From the original gray-scale image (top), we compute eight edge maps (two lower rows), corresponding to eight different orientations of a simple edge detector. Integral images of these edge maps are used to efficiently compute proportions of edges in rectangular windows.

denote the presence of an edge of orientation e at pixel l in image x. We assume $\xi_e(x, l)$ is equal to 0 if the location l is not in the image plane.

Our features, similar to those of [13], compute the ratio of edges of a particular orientation within a sub-window of the detector's $r \times r$ square of interest, with respect to the total number of edges within the same sub-window. Let R denote such a sub-window of random size and location contained in $\{1, \ldots, r\} \times \{1, \ldots, r\}$ plane. Our features are entirely parameterized by the sub-window R and the edge type e and are defined as:

$$h_{R,e}(x) = \frac{\sum_{m \in R} \xi_e(x, m)}{\sum_{d \in \Phi,\, m \in R} \xi_d(x, m)}. \qquad (17)$$

These features give the classifier the ability to check for the presence of outlines and textures and can be computed in constant time using eight integral images, one for each edge map.

4.2. Pose-Indexed Image Features

From the image features described above, we define a set of features indexed by a location in the image plane and an orientation. We define $\Theta_1 = \{1, \ldots, W\} \times \{1, \ldots, H\}$ and $\Theta_2 = [-\pi, \pi[$.

Given a rectangular sub-window R, and poses $\theta_1 = (u, v) \in \Theta_1$, and $\theta_2 \in \Theta_2$, we define

$$R_{\theta_1, \theta_2} \qquad (18)$$

as the rectangular window in the image plane obtained by applying a rotation of angle θ_2 and a translation (u, v). The

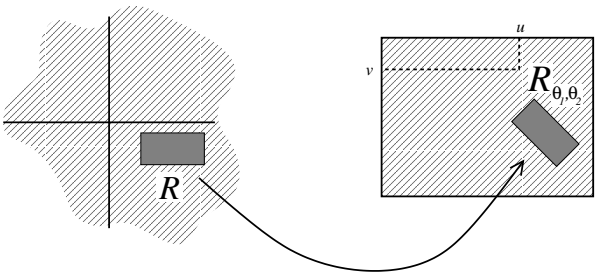

Figure 3. From a rectangular window R and a pose (u, v, θ_2), we define a indexed window R_{θ_1,θ_2}. Here $\theta_2 = \pi/4$.

orientation of the resulting window is quantified with a resolution of $\pi/4$ for computational reasons, see Fig. 3.

Similarly, given a edge orientation $e \in \Psi$ and an angle $\theta_2 \in \Theta_2$, we define

$$e \oplus \theta_2 \tag{19}$$

as the orientation obtained after a rotation of θ_2 is applied to the edge, that is, the edge orientation in Ψ closest to $e + \theta_2$.

With the above notation, we can define a set of pose-indexed features from $h_{R,e}$ introduced above, with

$$g_{R,e}((\theta_1, \theta_2), x) = h_{R_{\theta_1,\theta_2}, e\oplus\theta_2}, \tag{20}$$

that is, the proportion of edges, with a rotated edge orientation in the translated and rotated rectangular window.

4.3. Pose Estimators

We define a family of pose estimators which estimate a meaningful orientation $\theta_2 \in \Theta_2$ from a location $\theta_1 = (u, v)$. Our pose estimators compute the dominant edge orientation in a particular window Λ contained in the neighborhood of θ_1. More precisely, we define

$$\eta_\Lambda(\theta_1) = \arg\max_{e\in\Psi} h_{\Lambda_{\theta_1}, e} \ , \tag{21}$$

which computes the dominant edge orientation θ_2 in the window Λ translated according to θ_1.

Given the $\{1, \ldots, r\}\times\{1, \ldots, r\}$ plane, we define 14 regions for the pose-estimators corresponding to the complete square, the four regular sub-squares, and the nine regular sub-squares, which leads to 14 different pose-estimators, as shown on Fig. 4. Note, the estimated pose with the same number of bins so as to allow for the reuse of the integral images.

5. Empirical Results

To evaluate the performance of our proposed learning strategy, two sets of experiments were performed. In the first experiment, we compare the performance of our method with that of the aforementioned standard feature set.

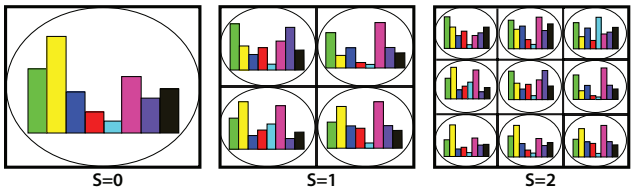

Figure 4. Our family of pose estimators. Given the square of interest of size $r \times r$ centered on θ_1, there are 14 pose estimators in total operating: each one computes the dominant edge orientation θ_2 in one of the sub-squares at three different scales S.

In the second experiment, we synthetically enrich our data with random rotations while maintaining its size. We analyse the performance of our method with respect to a typical pose normalization scheme. In what follows, the specifics of our experimental setup are given and the results of our experiments provided.

5.1. Learning

The standard AdaBoost learning procedure is used. Two boolean flags are added to the definition of our augmented pose-indexed features. The first indicates if the feature is to take the pose estimate into account. If so, the second flag specifies if the feature's window is to be registered according to the rotation described in § 3.2. Given a pose,$(\theta_1, \theta_2) \in \big(\{1, \ldots, W\} \times \{1, \ldots, H\}\big) \times [-\pi, \pi[$, three types of features are hence obtained:

- The first ignores the pose estimate and thus reduces to the standard feature as it simply translates its window with θ_1.
- The second considers the pose estimate θ_2 insofar as its edge orientation type is concerned while still translating its window with θ_1.
- The third translates its window with θ_1 , applies a rotation to the latter and changes its edge orientation type according to θ_2.

The selection of the stump at every iteration of AdaBoost results from examining 1000 of these features. The boolean flags are naturally selected randomly, with probability 0.5. The pose estimator is also chosen randomly: the scale at which it examines the signal is first chosen uniformly and the same is true for the sub-square over-which orientation is computed (among 1, 4 or 9 possible), see Fig. 4. Finally, the window R and the edge orientation e are also chosen uniformly at random. In all our experiments, the classifier combines 100 of these features. Since we observed the absence of over-fitting, we did not optimize that number through cross-validation. The threshold ρ_i of the selected stumps is optimized through an exhaustive search. When comparing with the standard feature set, we also considered

a total of 100 stumps and a search over 1000 features at every iteration.

5.2. Data set

We carried out our tests on two data sets. Each data set contains two hand sequences: while one sequence is used for training, the other is used for testing. Our first data set was obtained from a hardware-specialzed camera [7] which directly computes edges and is to an extent illumination invariant. These sequences have a resolution of 128×160, a frame rate of approximately 7 *fps* and an approximate duration of 4 minutes. The scene consists of a piece of heavy machinery with a few moving parts and clutter and the detector's window of interest is of size 44×44. Our second data set was obtained from a standard webcam. These sequences have a resolution of 144×192, a frame rate of approximately 10 *fps* and an approximate duration of 5 minutes. The scene consists of a typical disorderly office desk while the detector's window is of size 34×34

5.3. Error rate

Error rates were computed in a conservative fashion. A detection is a true alarm if its location is within a certain distance from the target and a false alarms otherwise. The considered distance is half the diagonal length of the detector's window of interest: $\frac{\sqrt{2*44^2}}{2}$ for our first data set obtained with the hardware-specialized camera and $\frac{\sqrt{2*34^2}}{2}$ for our second data set obtained from the webcam. In several frames and in both data sets, hands often completely occlude each other. In this scenario, if only one alarm is raised, a miss is counted.

5.4. Results

In the first experiment, we compared the performance of our augmented feature set with that of the standard features. As shown in Figures 5 and 6, incorporating pose estimator learning with feature learning provides with a significant gain in improvements at *all* detection rates and for both data sets. Indeed our method is able to capture the strong changes in appearance of the hand where the standard features fail. Most notably, at 95% true positive rate our first hardware-specialzed data set, our method raises 0.11 false alarms per frame versus 0.18 for the standard features. Some example frames, chosen uniformly at random, are shown in Figure 7 for our webcam data set.

We are interested in how well our method performs when the data set is enriched in pose but not increased in size. To this end, we devised a simple setup where every training frame is synthetically rotated at an arbitrary angle between $[0, \alpha]$. The same is done to every testing frame. For fairness, we compared the performance of our learning method with a typical pose normalization scheme. Specifically, a

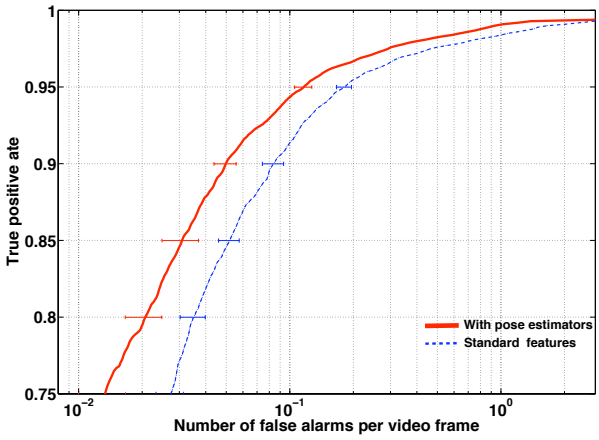

Figure 5. True-positive rate vs. the number of false alarms per frame on a logscale. The thin blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators. The training and testing hand sequences are from the hardware-specialized camera.

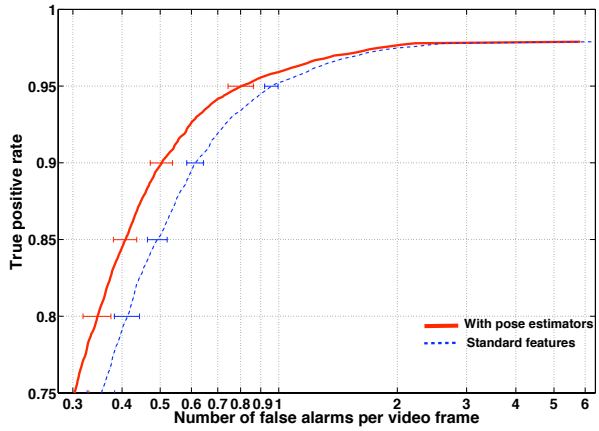

Figure 6. True-positive rate vs. the number of false alarms per frame on a logscale. The thin blue curve corresponds to the performance of the standard feature set while the thick red curve shows the performance of the detector using the combination of pose-indexed features and pose-estimators. The training and testing sequences are from the webcam.

scheme which during both training and testing would estimate the global pose and correct for it. This setup can easily be simulated, with a high degree of fidelity, simply by using our augmented feature set constrained to examine the pose at the largest scale and constrained to rotate the windows of the features.

Figure 8 shows an interesting trend. Our method shows significant gains compared to the pose normalization

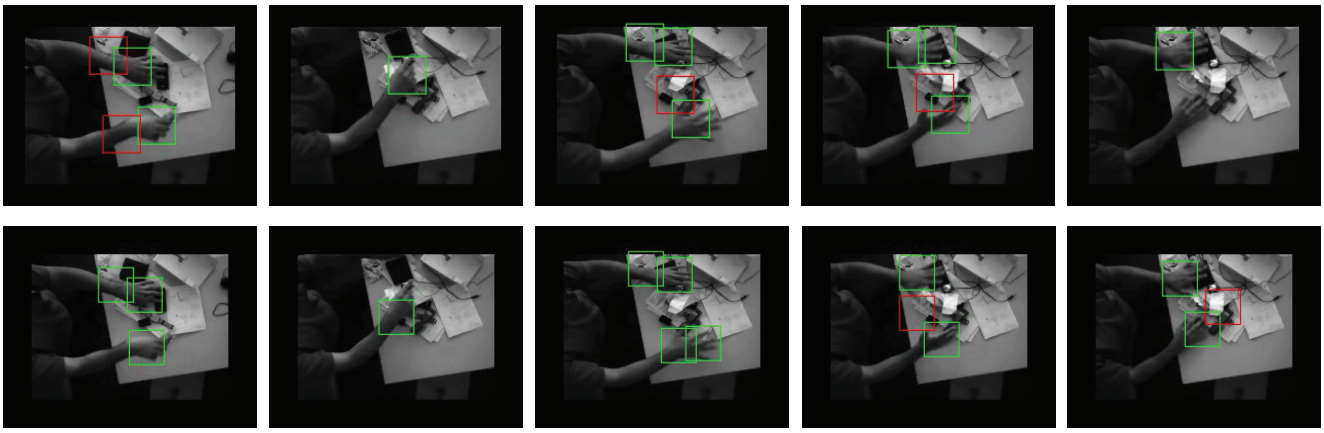

Figure 7. Examples of detections with the standard feature set (top row) and our classifier using pose-estimators (bottom row). This frames have been picked at random uniformly over the full sequence. The correct alarms are shown in green and the false alarms in red.

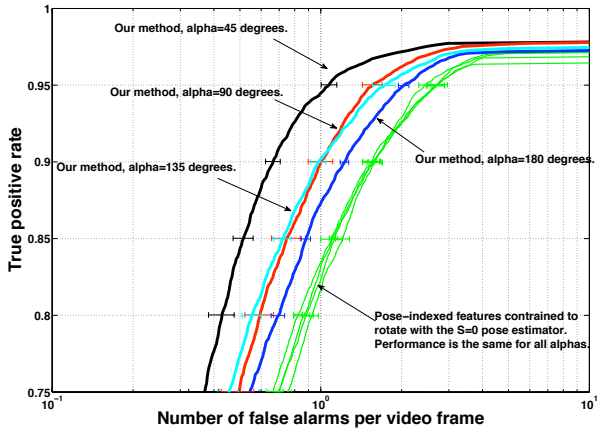

Figure 8. True-positive rate vs. the number of false alarms per frame on a logscale. A rotation taken at random in $[0, \alpha]$ was artificially applied to every training and test frame. The cluster of green curves corresponds to the performance of a scheme which corrects for in-plane rotations during training and testing for all α's. Our method demonstrates significant gains at $\alpha = 45$, $\alpha = 90$ and $\alpha = 135$ and more modest gains for $\alpha = 180$.

scheme for every value of α. As would be expected, the pose normalization scheme's performance is identical for all α's while our method shows high gains for $\alpha = 45$ with decreasing gains for higher values of rotations. Table 1 details these results for true positive rates of 95% and 90%. Note that at 90% true positive rate, our method achieves gains of 139%.

On a separate note, interesting empirical observations were made with respect to the selected features. Indeed, for our first experiment, we noted that out of the total features selected by Adaboost, approximately 55% of them were of

Table 1. Number of false-alarms at two fixed true-positive rates vs. an the amplitude of the rotation variation of the images. A rotation taken at random in $[0, \alpha]$ was artificially applied to every training and test frame.

TP = 95%		
α	Standard features	Pose-estimators
0	1.92	0.79
45	2.44	1.07
90	2.73	1.56
135	2.69	1.73
180	2.68	2.04

TP = 90%		
α	Standard features	Pose-estimators
0	1.11	0.53
45	1.58	0.66
90	1.55	1.00
135	1.59	0.98
180	1.56	1.24

the augmented variety, with the remaining being standard features. In the case of the second experiment involving the rotations, this number increases substantially to about 75%. Also the first ten classifiers in this second set of experiments are almost always selected from the augmented variety. This of course is to be expected: as the data set is enriched in pose, augmented features stand a far better chance to model the variations.

In addition to the presented results, we supply corresponding video-sequences as supplementary material to demonstrate the benefits of our method. Video sequences showing detection performance are provided for our first experiment.

6. Concluding Remarks

We introduced a novel object-detection strategy to handle target poses going beyond classical location and scale. Our method consists of designing a series of pose estimators, able to directly compute the target orientation in the image plane, and to allow the learning process to chose the most efficient combinations of pose estimators and pose-indexed features.

This procedure produces a detector able to modulate its features according to the image signal, hence adapting to variations in appearance and local deformations without the need for fragmenting the data during training, nor visiting additional pose parameters during detection.

A simple class of features truly invariant to rotation would compute the maximum proportion of edges over all possible orientations in a fixed sub-window. The pose estimators we use, as defined in § 4.3, provide the same operator when the windows of the pose-estimator and the pose-indexed features are identical. Hence, the features we have designed form a super-set of simple truly invariant features, as they are able to estimate the orientation in a window, and evaluate the response for that orientation in another one.

Extension of this work can follow two different axes. The first is to consider the use of more complex pose-estimators, going beyond the direct use of the edge counting features. The second axis will consist of investigating the relationship between standard invariant features and alternatives of the combination of pose-indexed features and pose-estimator we propose here, as stated above. By deconstructing standard image invariants in the same way, we may exhibit new valuable classes of both pose-indexed features and pose estimators.

References

[1] F. Fleuret and D. Geman. Stationary features and cat detection. *Journal of Machine Learning Research*, 9:2549–2578, 2008.

[2] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[3] M. Kolsch and M. Turk. Analysis of rotational robustness of hand detection with a viola-jones detector. *International Conference on Pattern Recognition*, 3:107–110 Vol.3, Aug. 2004.

[4] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. *European Conference on Computer Vision*, pages 67–81, 2002.

[5] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *Conference on Computer Vision and Pattern Recognition*, June 2009.

[6] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. *Conference on Computer Vision and Pattern Recognition*, page 963, 1998.

[7] P.-F. Ruedi, P. Heim, F. Kaess, E. Grenet, F. Heitger, P.-Y. Burgi, S. Gyger, and P. Nussbaum. A 128 x 128 pixel 120-db dynamic-range vision-sensor chip for image contrast and orientation extraction. *Solid-State Circuits, IEEE Journal of*, 38(12):2325–2333, Dec. 2003.

[8] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. *Conference on Computer Vision and Pattern Recognition*, 1:746–751 vol.1, 2000.

[9] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Filtering Using a Tree-Based Estimator. In *International Conference on Computer Vision*, volume 2, pages 1063–1070, 2003.

[10] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Estimating 3d hand pose using hierarchical multi-label classification. *Image Vision Comput.*, 25(12):1885–1894, 2007.

[11] P. Viola and M. Jones. Fast Multi-view Face Detection. Technical Report TR2003-96, MERL, 2003.

[12] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.

[13] D. G. Y. Amit and B. Jedynak. Efficient focusing and face detection. *Face Recognition: From Theory to Applications*, 1998.

[14] Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff. Multiplicative kernels: Object detection, segmentation and pose estimation. *Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[15] J. Zhang, S. Zhou, L. McMillan, and D. Comaniciu. Joint real-time object detection and pose estimation using probabilistic boosting network. *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.