Object Detection with Bootstrapped Learning [*]

Peter M. Roth* , Horst Bischof*, Danijel Skočaj** and Aleš Leonardis**

* Graz University of Technology, Institute for Computer Graphics and Vision
Inffeldgasse 16/II, 8010 Graz, Austria

** University of Ljubljana, Faculty of Computer and Information Science
Tržaška 25, SI-1001 Slovenia

e-mail: {pmroth, bischof}@icg.tu-graz.ac.at,
{danijel.skocaj, alesl}@fri.uni-lj.si

Abstract

This paper proposes a novel framework for learning object detection without labeled training data. The basic idea is to avoid the time consuming task of hand labeling training samples by using large amounts of unsupervised data which is usually available in vision (e.g. a video stream). We propose a bootstrap approach which starts with a very simple object detection approach, the data obtained is fed to the next level which uses a robust learning mechanism to obtain a better object detector. If necessary this detector can be further improved by the same mechanism at the next level. We demonstrate this approach on a complex person detection task. We show that we can train a person model without any labeled training data.

1 Introduction

Starting with face detection [12, 18] there has been a considerable interest in visual object detection in recent years, e.g., pedestrians [19], cars [1], bikes [10], etc. At the core of most object detection algorithms is usually a classifier, e.g., AdaBoost [3], Winnow [7], Neural network [12] or support vector machine [17]. The task of the classifier is to decide if the cropped window contains the object of interest or not. The search is repeated for all locations and scales, therefore, the classification has to be very fast. The proposed approaches have achieved considerable success in the above mentioned applications.

However, a requirement of all these methods is a training set which in some cases needs to be quite huge (several thousands of scaled and aligned images). The problem of obtaining enough

[*] This work is funded by the Austrian Joint Research Project Cognitive Vision under sub-projects S9103-N04 and S9104-N04 and by a grant from Federal Ministry for Education, Science and Culture of Austria under the CONEX program.

training data increases even further because the methods are view based, i.e., if the view-point of the camera changes significantly (e.g. car from the side and car from the back) the classifier needs to be retrained. Training data is usually obtained by hand labeling a large number of images which is a time consuming and tedious task. It is clear that this is not practicable for applications requiring a large number of different view-points (e.g. video surveillance by large camera networks). Therefore, the main limitation of these approaches is to obtain a representative set of labeled object data. Negative examples (i.e., examples of images not containing the object) are usually obtained by a bootstrap approach [15]. One starts with a few negative examples and trains the classifier. The obtained classifier is applied to images not containing the object. Those sub-images where a (wrong) detection occurs are added to the set of negative examples and the classifier is retrained. This process can be repeated several times. Therefore, obtaining negative examples is usually not much of a problem.

The main contribution of this paper is to propose a novel framework to avoid hand labeling of training data for object detection tasks which is demonstrated on a pedestrian detection task. The basic idea is to use the huge amount of unlabeled data that is readily available for most detection task (i.e., just mount a video camera and observe the scene).

In particular, we propose the following (see Fig. 1 for an illustration): We start training the detection framework with a simple classifier (in our particular example we use a simple motion detection by background subtraction and classification based on the size and aspect ratio of the motion blobs). It is clear that this simple classifier will produce wrong classifications (if not, our task would be solved and we can stop). Nevertheless we can use this classifier to produce a labeled training set. Since we have excessive data available we can pick only those samples where the classification is correct with a high likelihood (i.e., we have only a few false detections but might miss many persons). Fig. 2 shows the results of such a simple classifier. The person in (a) is detected correctly, while the persons in (b) and (c) are not detected because the person with the buggy and the biker produce a too large motion blob. The thus obtained training set can now be used to train another classifier.

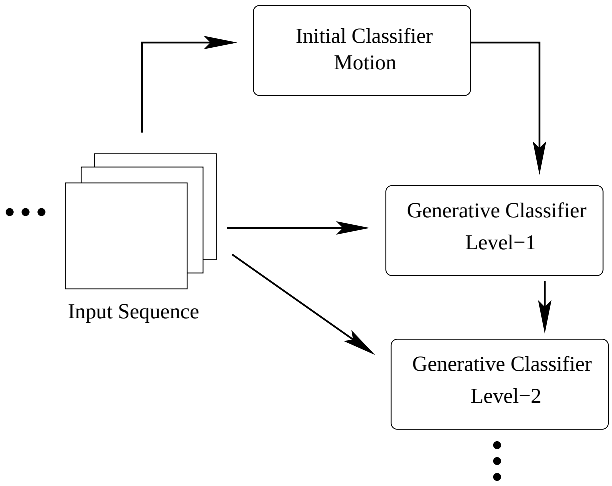


Figure 1: A sketch of the proposed framework of bootstrapped learning

In particular we propose to use a generative (reconstructive) method because it allows us to apply a robust detection scheme. Since we expect also wrong examples in the training set (misclassifications) we emphasize the importance of a robust training algorithm that can cope with outliers in the training data. To demonstrate the approach we use robust PCA as a generative robust classification procedure. The outlined process can be continued because with the new classifier we can produce a better labeled training set which we can use in turn to train another classifier etc. In fact, we can even use voting of the multiple classifiers obtained in the process (they are trained on different data) to generate another classifier. In the more general case we can also use different features for the different classifiers. In this case we increase the diversity of the classifiers which in turn would support voting. Later in the process we can also introduce discriminative classifiers like AdaBoost (the problem with discriminative methods is that they are inherently non robust (which is particularly true for AdaBoost) and are therefore not well suited in the early stage of training, but later when we have an almost correct training set we can obtain better detection rates with discriminative methods). The negative examples needed for training discriminative classifiers can be obtained by standard bootstrapping.

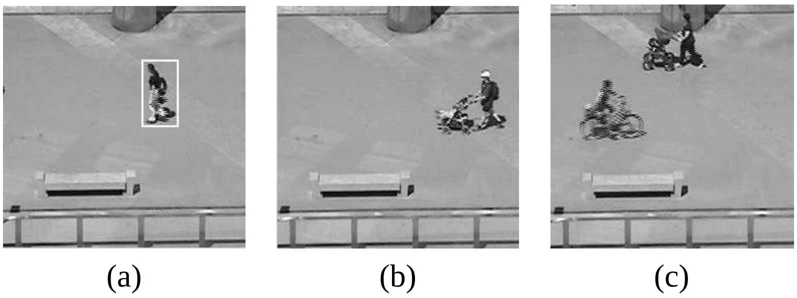

(a) (b) (c)

Figure 2: Match and mismatches of the motion classifier

The outlined approach is similar to the recent work of Nair and Clark [9] and Levin et al. [6]. Nair and Clark propose to use motion detection for obtaining the initial training set and then Winnow as a final classifier. Their approach does not include a robust training procedure which we show is quite beneficial, nor does it iterate the process to obtain more accurate results. In that sense our framework is more general. Levin et al. use the so called co-training framework to start with a small training set and to increase it by using a co-training of two classifiers operating on different features. This is similar to our voting proposal. However, they do not emphasize the role of robust classification, nor do they include more than two classifiers in the process. In this sense our approach is a generalization of both approaches. We emphasize the role of robust training which achieves better classification results in shorter time (i.e., make better use of the data), we use both generative and discriminative classifiers in the same framework.

The rest of the paper is organized as follows. In section 2 we detail our approach. In order to make the discussion concrete we will discuss it within the framework of person detection from videos. We will demonstrate the framework on using motion detection (a simple approximated median background model) and PCA and robust PCA as generative classifiers. The experimental results in section 3 demonstrate the approach on some challenging outdoor video sequences with groups of people and occlusions. Finally, we present some conclusions and outlook.

2 Bootstrapped Learning for Person Detection

Let us put the whole procedure in more formal terms. Let $X = \{\mathbf{x}_1, \dots \mathbf{x}_n | \mathbf{x}_i \in \mathbb{R}^m\}$ be a set of unlabeled training data. In fact, we assume that this set is very large (our scenario is video surveillance where we can easily obtain huge amounts of unlabeled data). We start the bootstrap procedure with a very simple classifier C_0 (in this paper we use motion detection and classification according to the bounding box of the motion blob see section 2.1). This classifier will label some of the images in the sequence, i.e., we obtain a set of positively labeled samples $X_0 = \{\mathbf{x}_i | \mathbf{x}_i \in X, C_0(\mathbf{x}_i) = 1\}$, where we denote a positive classification with $C_0(\mathbf{x}_i) = 1$. It is important that we take only those classifications that are quite certain, therefore the set X_0 will contain a lot of correctly labeled examples (we assume that the majority of samples will be labeled correctly), but a lot of correct samples from the set X will not be included in X_0, in addition there will be also some mislabeled samples (minority of samples)[1].

To obtain a better classifier in the next round we can now use the set X_0 for training a classifier C_1. There are two important things to note, we have only positively labeled samples and the training set might contain outliers. Therefore, we propose to use a generative (reconstructive) classifier in level C_1, in particular we will use PCA (see section 2.2) but any other generative classifier can be used as well. This choice is important because only with generative methods we can learn from positive samples only, moreover we have the possibility to perform robust training, i.e., reject entire training samples or take only sub-parts of images for training (e.g. detect the person pushing the buggy and not the buggy). Having trained C_1 we can employ it on the parts of the set X not used by any other classifier (i.e., in fact we have a sequence and we can just use the next images), producing a labeled set X_1', the new training set is then $X_1 = X_1' \cup X_0$. This process can be continued to obtain C_2 and X_2 etc. The two important things to note are that we produce a diverse set of classifiers since we are using different data to train them (this is the same argument as used in boosting). If the classifiers produced are not diverse enough (i.e., the improvements from step to step are slow) we can increase the diversity by following options:

1. We can use a different classifier in the process (in fact when we have already a large training set we can also use a discriminative classifier).
2. We can use voting of classifiers $C_1 \dots C_k$ to obtain a new one.
3. We can use a different set of features and employ ideas from co-training [6] to increase the diversity.

The critical question is if the outlined process will improve the classifier in each step and when will it converge. A proof under general conditions is difficult to obtain (and perhaps not even possible). But there are several hints which show that the proposed approach is powerful. The process resembles in some aspects Boosting, which has been proven to improve the classification accuracy of weak classifiers [3]. The process is also closely related to co-training [2]

[1] If we are not able to obtain an "unsupervised" initial classifier we could start the whole process using a few hand labeled samples to generate an initial classifier.

which has been proven to increase the accuracy of the underlying classifiers (unfortunately the assumptions for the proof are rarely satisfied in practice). As the experiments in section 3 demonstrate the proposed method works for a a challenging person detection task.

2.1 Motion Detection

Having a stationary camera a common approach to detect moving objects is to threshold the difference image between the currently processed image and a background image. A widely used and simple method for generating a background model is a pixel-wise median filter of length L:

$$\mathbf{b}_t(m,n) = median_L(\mathbf{x}_{t-L}(m,n), \ldots, \mathbf{x}_t(m,n)) \tag{1}$$

This implicitly assumes that an object will not stay at the same position for more than $L/2$ frames. This simple method has two main disadvantages:

1. The median must be computed at each time step *t*.
2. *L* frames must be stored in the memory.

An alternative computationally more efficient method was developed by McFarlane and Schofield [8]. The approximated median filter computes an approximation of the median by incrementing the current estimate by one if the input pixel value is larger than the estimate and by decreasing it by one if smaller:

$$\mathbf{b}_{t+1}(m,n) = \begin{cases} \mathbf{b}_t(m,n) + 1 & \mathbf{b}_t(m,n) < \mathbf{x}_t(m,n) \\ \mathbf{b}_t(m,n) - 1 & \mathbf{b}_t(m,n) > \mathbf{x}_t(m,n) \end{cases} \tag{2}$$

This estimate eventually converges to the real median. Thus, after the initial model was computed only one reference image must be stored in memory. Considering the size and the geometry of a detected blob, the motion information can be used as a simple classifier.

2.2 PCA and Robust PCA

As a classifier we use PCA and Robust PCA. The main reason is that they are simple and efficient generative methods and that there are robust and on-line variants of these algorithms available (even robust and on-line algorithms see [13]).

Let $\mathbf{x}_i = [x_{1i}, \ldots, x_{mi}]^T \in \mathbb{R}^m$ be an individual image represented as a vector, and $X = [\mathbf{x}_1, \ldots \mathbf{x}_n] \in \mathbb{R}^{m \times n}$. To simplify the notation, we assume X to be normalized, having zero mean. The eigenvectors (principal axes) obtained from X are denoted by $\mathbf{e}_i = [e_{1i}, \ldots, e_{mi}]^T \in \mathbb{R}^m$; $E = [\mathbf{e}_1, \ldots \mathbf{e}_n] \in \mathbb{R}^{m \times n}$. The columns of E, i.e., eigenvectors, are arranged in decreasing order with respect to the corresponding eigenvalues. Usually, only k, $k < n$, eigenvectors (those with the largest eigenvalues) are needed to represent $\mathbf{x}_i$ to a sufficient degree of accuracy as a linear combination of eigenvectors $\mathbf{e}_i$:

$$\tilde{\mathbf{x}} = \sum_{i=1}^{k} a_i(\mathbf{x})\mathbf{e}_i = E\mathbf{a} \quad , \tag{3}$$

where $\tilde{\mathbf{x}}$ denotes the approximation of $\mathbf{x}$. The entire set of images X can thus be represented as $\tilde{X} = EA$ where $A = [\mathbf{a}_1, \ldots \mathbf{a}_n] \in \mathbb{R}^{k \times n}$ consists of coefficient vectors $\mathbf{a}_i = [a_{1i}, \ldots, e_{ki}]^T \in \mathbb{R}^k$.

Having an eigenspace encompassing the training images and being given an input image $\mathbf{y}$, recognition occurs as an estimation of the parameters $a_i(\mathbf{y})$. These can be calculated by a standard projection

$$a_i(\mathbf{y}) = \mathbf{e}_i^T \mathbf{y} = \sum_{j=1}^{m} e_{ji} y_j \quad , \quad i = 1 \ldots k \quad , \tag{4}$$

or, as a robust procedure [5], by solving a system of linear equations

$$y_{r_i} = \sum_{j=1}^{k} a_j(\mathbf{y}) e_{r_i,j} \quad , \quad i = 1 \ldots q \quad , \tag{5}$$

evaluated at $q \geq k$ points $\mathbf{r} = (r_1, \ldots r_q)$.

Once we have obtained the parameters $a_i(\mathbf{y})$ we can reconstruct the image using (3), and determine the reconstruction error $\epsilon = ||\mathbf{x} - \tilde{\mathbf{x}}||$, based on this error we can perform object detection.

In order to train PCA when we have noisy samples we need a robust algorithm for obtaining E. Several methods to robustly extract the principal axes in the presence of outliers have been proposed in the statistical community; see [4] for a nice overview. The major drawback of these methods is that either they rely on the calculation of a "robust" covariance matrix, which is due to the high dimensionality of image data not feasible, or that they discard entire data vectors [20]. In the later case, a whole image would be eliminated just because of a single outlying pixel. For the tasks we envision to tackle, this would mean that no images would be usable since, in general, each of them might contain some outliers.

Two recent papers one by De la Torre and Black [4] and the other by Skocaj et al. [14] have presented methods which are robust and suitable for high dimensional image data. The method of De la Torre and Black is based on robust M-estimator. Their formulation yields a high dimensional non-linear optimization problem which has to be solved in an iterative manner. Therefore, the computational complexity of the algorithm is very high. The algorithm of Skocaj et al. is based on the EM formulation of PCA [11, 16] and is computationally much cheaper.

3 Experimental Results

3.1 Test Data

For our experiments we used a challenging surveillance video consisting of approximative 15000 frames. The video is available in uncompressed true color with a resolution of 720×576

pixels (see Fig. 3). For computational reasons we converted the frames to gray-values and cropped only subimages of the size 240×240. For evaluation purpose we generated a new video from several interesting subsequences (containing multiple persons, occlusions etc). This video consists of 1650 frames and was manually annotated (all together about 1750 persons), where the average size of the people is about 60×30 pixels. For training we used only every fifth frame to avoid too many similar training objects.

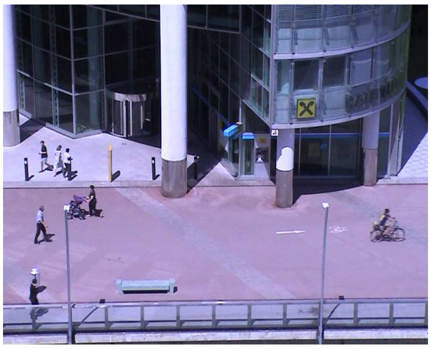


Figure 3: Full video frame: 720×576 pixels

3.2 Description of Experiments

We have evaluated three steps on the video sequence described above to show that the detection rate can be increased from step to step.

Step 1: The first classifier C_0 is based on motion information. Moving pixels are estimated by thresholding the difference image between the current frame and an estimated background model. Connected foreground pixels are grouped into blobs that are geometrically tested. Only blobs that fulfill the size and aspect ratio restrictions are accepted as a person (these parameters are derived from a rough calibration of the ground plane). For estimating the background model the approximated median method is applied. As a result we get the labeled set X_0.

Step 2: The second classifier C_1 is obtained by standard PCA training on X_0 (there is not enough data for robust PCA in this step). For detection subimages are cropped and transformed into the eigenspace, where the PCA coefficients are estimated by the robust procedure [5] (see equation (5)). If the reconstruction error is below some threshold θ_{person} a subimage is accepted as a person. To save computing time, only those regions are processed where the motion detection has detected blobs. To ensure that partially occluded persons are also included in this processing step the detected blobs are enlarged based on the geometry of the blob. As a result we get the labeled set X_1'.

Step 3: The third classifier C_2 is obtained by PCA training on X_1, where $X_1 = X_1' \cup X_0$. In contrast to Step 2 the eigenspace is now computed using the the robust PCA learning algorithm [14]. The detection is performed as described in Step 2.

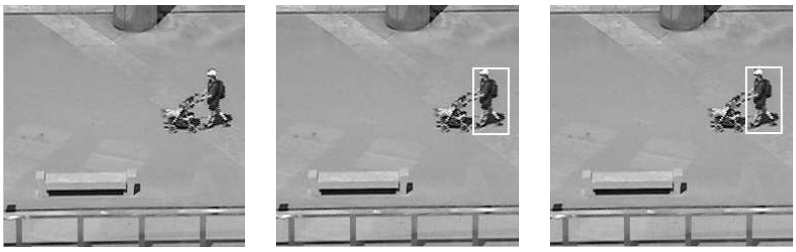
3.3 Results

The simple classifier based on motion information (Step 1) detects only about 34% of all pedestrians[2], but due to the rather restricted parameter settings there are no false positives. Using the standard PCA method (Step 2) we get a detection rate of 64%, almost two times as much detections as in Step 1. But note that there also a few false positives now, where partially detected persons are counted as errors. The results of Step 3 show an increase in performance using the robust algorithm. The detection rate is increased to 75%, while the misclassification rate is decreased. This demonstrates the advantage of using a robust learning algorithm. These results are summarized in Table 1.

	matches %	mismatches %
Step 1	34.3%	0.0%
Step 2	64.3%	1.9%
Step 3	75.5%	1.1%

Table 1: Detection/False Detections rates on the whole annotated video sequence

Let us now look at a few examples. Fig. 4 shows the father with the buggy. It was not detected in Step 1 because the motion blob is too big. But it was detected in Step 2 and also in Step 3.



(a) (b) (c)

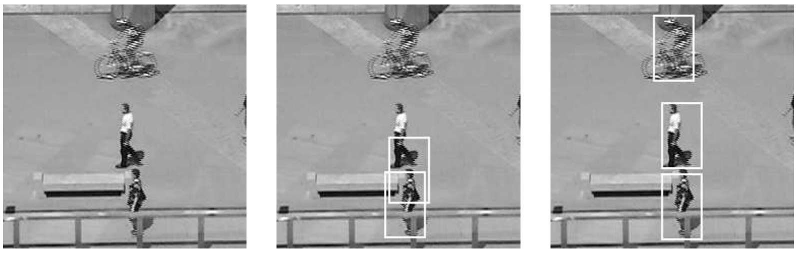
Figure 4: Father pulling the buggy is detected: (a) Step 1, (b) Step 2, (c) Step 3

The next example shows the benefit of using a robust training procedure. Fig. 5 shows two persons close together. They are not detected in Step 1, in Step 2 a false detection occurs, but using the robust PCA method both persons are correctly detected. Additionally the biker is detected in Step 3.

4 Summary and Conclusions

We have presented a novel framework for unsupervised training of an object detection system that can be applied for automatically labeling a huge amount of data. Therefore the time consuming task of hand labeling the data can be reduced to an initial estimation of the parameters

[2] Here all bikers are accepted as a persons. The total number of bikers is below 1% of all detections.



(a) (b) (c)

Figure 5: RPCA yields matches of higher quality: (a) Step 1, (b) Step 2, (c) Step 3

needed for motion detection. The basic idea is to use a bootstrap approach, starting with a very simple object detection system and then using a sequence of classifiers to generate better object detectors. We have demonstrated the framework on a surveillance task where we have learned a pedestrian detector. We have started with a simple moving object classifier and then used PCA and Robust PCA to obtain more complex classifiers. Compared to the initial classification we have increased the performance of the system by more than a factor of 2. The final detection rate of 75% is not overwhelming, but so far we have due to limited amount of data not trained another classifier and stopped after Step 3. We have stressed the importance of robust training. As the examples have shown we will usually have errors in the training set, therefore it is important to be robust during training. We have used the system in an offline fashion, but for both PCA and Robust PCA there are online algorithms available, so we could in principle perform the training also online while the system is in operation.

The framework we have presented is quite general. Our next steps are to increase the diversity of classifiers and to include also voting in the process. Another interesting point is to add a discriminative classifier (e.g. AdaBoost) later in the process to further increase the detection rate.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proceedings of the 7th European Conference on Computer Vision*, volume 4, pages 113–130, 2002.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, pages 92–100. Morgan Kaufmann, 1998.

[3] Y. Freund and R. Shapire. A decision-theoretic generalization of online learning and an application to boosting. *J. of Computer and System Sciences*, 55:119–139, 1997.

[4] F. D. la Torre and M. Black. Robust principal component analysis for computer vision. In *Proc. ICCV01*, pages 362–369. IEEE Computer Society Press, 2001.

[5] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78(1):99–118, 2000.

[6] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *Proc. ICCV*, pages 626–633. IEEE CS, 2003.

[7] N. Littlestone. Learning quickly when irrelevant attributes abound. *Machine Learning*, 2:285–318, 1987.

[8] N. J. McFarlane and C. P. Schofield. Segmentation and tracking of piglets. *Machine Vision and Applications*, 8(3):187–193, 1995.

[9] V. Nair and J. Clark. An unsupervised, online learning framework for moving object detection. In *Proc. CVPR*, pages 317–324. IEEE CS, 2004.

[10] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. ECCV2004*, volume II, pages 71–84, 2004.

[11] S. Roweis. Em algorithms for pca and spca. In *NIPS*, pages 626–632, 1997.

[12] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[13] D. Skočaj. *Robust subspace approaches to visual learning and recognition*. PhD thesis, University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, 2003.

[14] D. Skočaj, H. Bischof, and A. Leonardis. A robust PCA algorithm for building representations from panoramic images. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proc. ECCV02*, volume IV, pages 761–775. Springer, 2002.

[15] K. Sung and T. Poggio. Example-based learning for view-based face detection. *IEEE Trans. PAMI*, 20:39–51, 1998.

[16] M. Tipping and C. Bishop. Probabilistic principal component analysis. Technical report, Microsoft Research, 1999.

[17] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Conference on Computer Vision and Pattern Recognition*, pages 511–518. IEEE CS, 2001.

[19] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proceedings of the Ninth IEEE Conference on Computer Vision (ICCV'03)*, volume 2, pages 734–741, 2003.

[20] L. Xu and A. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Trans. Neural Networks*, 6(1):131–143, 1995.