Centralized Information Fusion for Learning Object Detectors in Multi-Camera Networks *

Armin Berger, Peter M. Roth, Christian Leistner, and Horst Bischof

Institute for Computer Graphics and Vision
Graz University of Technology, Austria
armin.berger@student.tugraz.at, {pmroth,leistner,bischof}@icg.tugraz.at

Abstract
Recently, multi-camera networks have proven to be a valuable tool for learning object detectors. By exploiting geometry information given by homographies between different cameras in a co-training framework very valuable training samples can be acquired. However, for a growing number of cameras the existing methods become infeasible for two reasons: (a) The number of required camera-to-camera homographies increases dramatically and (b) the information fusion becomes more and more complicated. To overcome these drawbacks, we propose a centralized approach to fuse the information from different cameras during co-training. In particular, all detections are projected onto the common top view and are merged using a mean shift approach. To finally generate the updates, the thus obtained merged detections are re-projected to the specific camera views. We demonstrate our approach for the task of person detection, where we show that even using only a small number of labeled training samples finally state-of-the-art detection results can be obtained.

1. Introduction

Object detection, in general, is a very important task in computer vision. The most prominent approach in this context is to apply a sliding window technique. Each patch of an image is tested if it is consistent with a previously estimated model or not. Finally, all consistent patches are reported. A lot of research has been focused on efficient training classifiers, but only little attention has been paid to efficiently labeling and acquiring suitable training data. Thus, training data, i.e., positive and negative samples are usually obtained by hand labeling a large number of images, which is a time consuming and tedious task.

Negative examples (i.e., examples of images not containing the object) are usually obtained by a bootstrap approach [15]. Starting with a few negative examples a classifier is trained. The obtained classifier is applied to images that do not contain the object-of-interest. Those sub-images, where a wrong detection occurs (i.e., a false positive) are added to the set of negative examples and the classifier is retrained. This process can be repeated several times. Thus, obtaining negative examples is usually not much of a problem. Automatically obtaining positive examples is a more difficult task. Hence, typically semi-supervised learning methods are applied exploiting the information of (a small amount of) labeled and (a huge amount of) unlabeled data. A sub-set of these methods is *co-training* of Blum and Mitchell [5], which exploits redundant views of the same input data.

*This work was supported by the FFG project EVis (813399) under the FIT-IT programme, by the FFG project SECRECT (821690) under the Austrian Security Research programme KIRAS, and the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209.

Due to its beneficial properties co-training was also applied for many computer vision applications including background modeling [17], learning an object detector [12], or tracking [13]. Later, these ideas were extended for multiple cameras by Leistner et al. [10] regarding different *cameras* as different *views* for the same classification problem. The approach was supported by previous methods exploiting geometry information (i.e., the ground planes and the homographies between the cameras) in order to improve rather simple object or motion detectors ([4, 9]). However, even showing to be beneficial this approach becomes infeasible if the number of cameras is growing. Due to the required pairwise homographies the computational complexity is increased and the information fusion becomes more and more complicated.

The goal of this paper is to overcome these problems in context of training an object detector using only a small number of labeled samples. In particular, we build on the co-training approach of Leistner et al. introducing a new more efficient centralized merging strategy. For that purpose, we estimate projections from each camera onto the top view, only requiring to calibrate the cameras once. In this way avoiding the camera-to-camera-homographies co-training is performed based on the fused projections in the top view. In contrast to other methods we do not rely on a sample grid on the top view, thus, minimizing the projection errors. To show the benefits of the proposed approach we demonstrate it, even though not limited to this application, for learning of person detectors. We show that if a small number of labeled samples is used only, state-of-the-art methods can be outperformed that were trained from 10,000s of labeled samples!

The remainder of this paper is as follows. First, in Section 2., we review the ideas of co-training in multiple camera networks. Next, in Section 3. we introduce our new centralized multi-camera co-training system. Experimental evaluations are give in in Section 4. Finally, a conclusion and a summary are given in Section 5.

2. Co-Training in Multiple-Camera Networks

Supervised learning deals with a labeled dataset $\mathcal{D}^L \subseteq \mathcal{X} \times \mathcal{Y} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{|\mathcal{D}^L|}, y_{|\mathcal{D}^L|})\}$, where $\mathbf{x}_i \in \mathcal{X} = \mathbf{R}^P$ and $y_i \in \mathcal{Y} = \{+1, -1\}$. In contrast, unsupervised methods aim to find an interesting (natural) structure in $\mathcal{X}$, also implying a certain redundancy, using only unlabeled input data $\mathcal{D}^U \subseteq \mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{D}^U|}\}$. Since unlabeled samples can be obtained significantly easier than labeled samples the goal of semi-supervised learning is to exploit both, the information of labeled $\mathcal{D}^L$ and unlabeled $\mathcal{D}^U$ data. One way to exploit the redundancy in unlabeled data is co-training [5]. The main idea is to train two initial classifiers h_1 and h_2 on a small amount of labeled data $\mathcal{D}^L$ and then let these classifiers update each other using the unlabeled data $\mathcal{D}^U$. An update is performed if one classifier is confident on a sample whereas the other one is not. The approach has proven to converge [5] if two assumptions hold: (a) the error rate of each classifier is low and (b) the views must be conditionally independent. However, the second assumption was later relaxed by several authors (e.g., [1, 2]). Thus, co-training can typically also be applied if, in principle, the learners are strong and low-correlated.

Starting with the work of Levin et al. [12], who used co-training for learning a car detector, co-training became also quite popular in Computer Vision. In contrast to many artificial machine learning problems, computer vision offers many physical "real-world" constraints, which can guide a semi-supervised learning process. For that purpose, existing co-training approaches are applied combined with different simple cues based on shape, appearance, or motion (e.g., [3, 12, 14]). In practice, these cues have often shown to be too weak to lead to robust convergence. In contrast, Leistner et al. [10]

proposed to use different *cameras* as different *views* for the same classification problem. Similar to Khan and Shah [9] the key idea is to take advantage of a very strong real-world constraint: *geometry* (i.e., the ground planes and the homographies between the cameras). Considering a setup with n partly overlapping cameras, each of them observing the same 3D scene, the local image coordinate systems can be mapped onto each other by using a homography, that is based on identified points in the ground-plane, as illustrated in Figure 1.

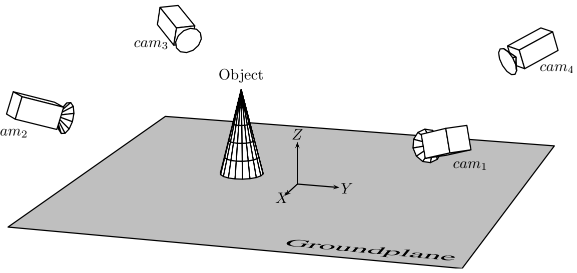


Figure 1. Multi-Camera Co-training: cameras observing a partly-overlapping scene teach each other.

To start the training process, first a general classifier C^0 is trained from a fixed set of labeled positive and negative samples. This initial classifier is cloned and used to initialize all n camera views: $C_1^0, \ldots, C_n^0$. In fact, due to the different camera positions highly independent views required for co-training are obtained; even if exactly the same classifier is applied! Then, these cloned classifiers $C_1^0, \ldots, C_n^0$ are co-trained exploiting the shared geometry.

3. Centralized Fusion of Multi-Camera Co-Training

The approach discussed in Section 2. is based on camera-to-camera homography estimation, which makes it infeasible in practice. Thus, in the following we introduce a multiple-camera co-training method avoiding these problems. Moreover, in contrast to other methods the information fusion algorithm is totally independent from the applied learning algorithm.

3.1. Single View Detection and Camera Calibration

To detect objects within a single view, we apply a sliding window technique. Thus, the detection window represented by a classifier is shifted over the image to reach all locations and different scales are achieved by scaling the detection window. In general, there are no further limitations, however, within the multi-camera learning framework we assume that the objects can move on the ground plane only. Thus, we estimate a ground plane and restrict the sliding window to positions on this plane.

To exchange information between the cameras a common ground plane (top view) for all cameras can be estimated. We use calibrated cameras in our centralized approach instead of using camera-to-camera homographies as described in Section 2. In particular, we determine the intrinsic and extrinsic parameters including the radial lens distortion parameter for each camera using Tsai's camera calibration algorithm [16]. Thus, the detections' base points can be inversely projected onto the 3D scene. Then by intersecting the back projected ray with the ground plane ($z = 0$ plane in the world coordinate frame) the top view coordinates can be obtained. To further increase the accuracy, we additionally correct the radial lens distortion with a first order Taylor polynomial. Having such a calibration the typical way for information exchange within a multiple-camera network would be to discretize the ground plane. Then the points in the common ground plane can be projected to each camera view to generate the corresponding search windows in the specific camera view [8]. Although quite efficient, this procedure fails if the camera angles or the distance between the camera to the observed scene are quite different, since projection errors are introduced, Thus, in our system we estimate the sliding window locations per camera allowing for more precise projections via the top view.

3.2. Information Fusion

To merge information from multiple cameras, we first run a detection step using the current camera-specific classifiers and apply, in contrast to existing approaches such as [8], a non-maximum suppression (NMS). This post-processing leads to a very sparse list of detections for each camera view reducing the complexity of the further processing queue. The obtained detection results are then projected onto the central top view map to obtain a centralized confidence map. Additionally, we retain the information in which camera view a specific detection occurred. However, due to imperfect aligned detections in the single camera views the projections in the top view map are misaligned too. To overcome this problem, in a local neighborhood around a specific location we check the Euclidean distance between all detections in the top view map and introduce a camera count c. The camera count is incremented if the distance to another detection from a different camera view is smaller than a threshold Θ.

Next, based on this camera count map we create a probability density map considering the corresponding confidence values in the top view map. In particular, we transfer only points from the top view map to the density map if $c \geq 2$ at the point's location (i.e., an agreement of at least two views is required). To estimate the final detection results, we apply a mean-shift clustering on the resulting density map to find all local maxima $P_\chi = \{p_{\chi,1}, \ldots, p_{\chi,k}\}$. Finally, P_χ represents locations on the ground plane where the desired objects are most likely located. The whole procedure is summarized more formally in Table 1.

```
  • Requires $N \geq 2$ calibrated cameras; a small set of positive and negative samples
  • Train an initial on-line classifier
  • Clone the classifier $N$ times
For each input image:
  • For $n = 1, \ldots, N$:
      1. Detect objects in camera view $n$.
      2. Apply NMS on the detection output $\rightarrow \mathcal{D}_n$.
      3. Transfer $\mathcal{D}_n$ into the top view map.
  • If $(\exists\ d(\mathbf{p}_i, \mathbf{p}_j) < \Theta\ \ \forall\ j \neq i,)$ then
      1. Increment camera count at $\mathbf{p}_i$
  • Top view map $\rightarrow$ probability density map $\forall\ \mathbf{p}_i$ with camera count $\geq 2$
  • Mean-shift on probability density map $\rightarrow$ local maxima: $P_\chi = \{p_{\chi,1}, \ldots, p_{\chi,k}\}$.
  • Output: $P_\chi$
```

Table 1. Centralized detections fusion from multiple cameras generating ground plane locations which might be used for a co-training update.

3.3. Generating Co-Training Updates

Once we have generated a centralized detection mask represented as a set of positions P_χ as introduced in Section 3.2., the remaining problem is to generate the updates. For that purpose, we project all points from P_χ back onto the original camera views and extract positive examples from the single camera views. Additionally, we restore their corresponding bounding boxes in these views (see Section 3.1.). A remaining problem, which would impede robust learning, are occlusions. Thus, we perform an additional occlusion check ensuring that the back projected points and the corresponding bounding boxes contain fully visible objects. In detail, each camera view's occlusion check takes all detections that survived the post-processing and computes the overlap with the back projected bounding boxes. If the overlap between any detection and a back projected bounding box is smaller than a threshold Φ

$$overlap(bbox_1, bbox_2) = \frac{area(bbox_1 \cap bbox_2)}{area(bbox_1) \cup area(bbox_2)} \quad , \tag{1}$$

we perform a positive update on all camera classifiers using the restored sub-window in the camera. To additionally acquire negative updates for each positive update a negative update is bootstrapped from a negative dataset. The overall update procedure is summarized in Table 2.

```
• Requires $P_\chi$ with $K$ locations from $N$ cameras.
• For $n = 1, \ldots, N$:
    1. For $k = 1, \ldots, K$:
        (a) Project $p_{\chi,k}$ into camera view $n$.
        (b) Compute the corresponding bbox.
        (c) Compute overlap between $p_{\chi,k}$ and $\mathcal{D}_n$.
        (d) If $overlap(bbox(\mathcal{D}_{n,j}), bbox(p_{\chi,k})) < \Phi \; \forall \; j \neq k$ then
            i. Positive update with $bbox(p_{\chi,k})$ for all classifiers
            ii. Negative update for all classifiers (bootstrapped)
```

Table 2. Generating positive updates using locations estimated by centralized information fusion (see Table 1).

4. Experimental Results

In the following, we demonstrate the proposed approach when learning a person detector[1]. We established a setup consisting of three synchronized static cameras having partly overlapping views, which were calibrated using Tsai's camera calibration software[2]. The experiments are split into two parts: first, we give an analysis of the learning behavior during training; second, we show that a person detector learned on this specific setup can be applied for a general (single camera) detection task. Although the co-training strategy proposed in this paper is quite general, in particular for our experiments we used an on-line GradientBoost-based[3] learner [11]. To increase the robustness and to capture different appearances, we applied a logistic loss-function and used Haar-like as well as HOG features as low level representation.

[1] This scenario was chosen since various reference implementations and benchmark datasets are publicly available.

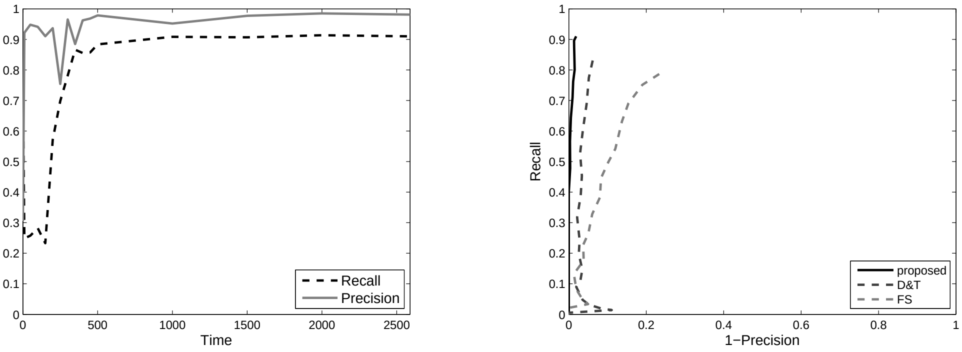
[2] http://www.cs.cmu.edu/ rgw/TsaiCode.html which is based on Tsai's camera calibration algorithm [16].

[3] Since Liu et al. [13] give a proof for error bounds for on-line boosting in co-training this is a reasonable choice.

4.1. On-line Learning

First of all, we demonstrate the on-line learning behavior of the proposed approach. For that purpose, we generated a training set consisting of 2500 frames and a independent test set consisting of 250 frames. To start the learning process, we trained an initial classifier using a fixed training set of 20 positive and 20 negative samples, which were randomly chosen from a larger dataset (i.e., MIT-CMU). This classifier was cloned and used to initialize the co-training process for each of the three cameras. To demonstrate the learning progress, after a pre-defined number of processed training frames we saved the corresponding classifiers, which were then evaluated on the independent test sequences (i.e., the current classifier was evaluated but no updates were performed!). Please note, there is no collaboration between different cameras during the evaluation. Hence, using a calibrated camera setup is unnecessary.

The corresponding results for view 3 are shown in Figure 2(a). It clearly can be seen that the final performance for both, recall and precision, is already achieved after less than 500 frames and that this performance keeps stable over time. In addition, we compared the final classifier obtained by our approach to two different state-of-the-art detectors[4], i.e., the Dalal and Triggs (D&T) person detector [6] and the deformable part model of Felzenszwalb et al. [7] (FS). These results are given in Figure 2(b), which show that these detectors can be clearly outperformed in terms of recall and precision. For reasons of completeness, finally in Table 3 for all three cameras the precision, the recall, and the F-measure are given for the initial and the final classifier, respectively. These show the same trend as can be recognized from Figure 2(a) – the recall as well as the precision can be drastically improved.



(a)

(b)

Figure 2. Learning scene specific classifiers: (a) improvement over time and (b) comparison of the final classifier to state-of-the-art methods.

	recall	prec.	F-M
view 1	0.72	0.25	0.37
view 2	0.71	0.29	0.41
view 3	0.75	0.31	0.43

(a)

	recall	prec.	F-M
view 1	0.87	0.96	0.91
view 2	0.88	0.96	0.92
view 3	0.91	0.98	0.94

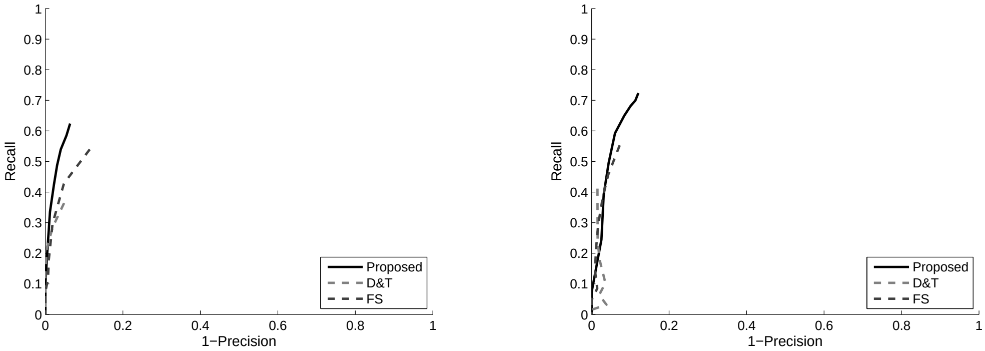
(b)

Table 3. Performance characteristics for all three cameras: (a) initial classifiers and (b) final classifiers.

[4] We abstained from a comparison to [10]. Since the method can not cope with the occlusions and the upcoming projection errors, it totally fails and no curves can be generated.

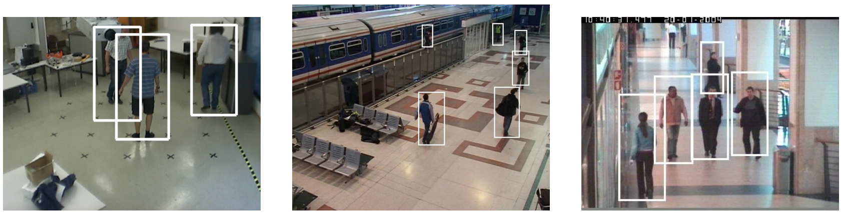
4.2. General Pedestrian Detection

Since we have shown that we can train performative scene-specific classifiers, i.e., the test sequences were taken from the same scenario as the training sequences, in the following we show that the acquired information is also beneficial for more general setups. In particular, we show that a classifier that was trained on the laboratory setup introduced in Section 4.1. also yields competitive results on publicly available standard benchmark datasets, i.e., the *PETS 2006* dataset[5] and the *Caviar* dataset[6]. The thus obtained RPCs, again compared to the two general detectors, are shown in Figure 3. It clearly can be seen that we obtain (more than) competitive results for both datasets, especially, considering the recall! Finally, in Figure 4 we show illustrative detection result for the three different scenarios.



(a) (b)

Figure 3. Evaluation on publicly available datasets: (a) PETS 2006 and (b) CAVIAR.



(a) (b) (c)

Figure 4. Illustrative detection results: (a) laboratory-scenario, (b) *PETS 2006*, and (c) *CAVIAR*.

5. Conclusion

The acquisition of data for training a classifier is an important, however, often ignored problem. In this paper, we proposed a method for training a detector from only a limited number of labeled data. For that purpose, we exploit constraints given by geometry for co-training in a multiple-camera-network. In contrast to existing methods our method differs in two main points: (a) The goal is to learn a general detector from a multiple-camera-setup which can be applied for an arbitrary single-camera setup. (b) To overcome problems arising from a larger number of cameras, an efficient centralized information fusion method is applied. In fact, due to the geometric constraints very valuable samples are selected for updating the classifiers. Thus, the autonomous learning process is quite stable and the classifiers are generalizing to different setups. For our experiments we applied a boosted on-line classifier, finally yielding state-of-the-art detection results.

[5] http://www.pets2006.net

[6] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1

References

[1] S. Abney. Bootstrapping. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 360–367, 2002.

[2] M.-F. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *Advances in Neural Information Processing Systems*, pages 89–96, 2004.

[3] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *Proc. ECCV*, volume I, pages 299–308, 1994.

[4] J. Berclaz, F. Fleuret, and P. Fua. Principled detection-by-classification from multiple views. In *Proc. VisApp*, pages 375–382, 2008.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. COLT*, pages 92–100, 1998.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, volume I, pages 886–893, 2005.

[7] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. CVPR*, 2008.

[8] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE Trans. PAMI*, 30(2):267–282, 2008.

[9] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proc. ECCV*, volume IV, pages 133–146, 2006.

[10] C. Leistner, P. M. Roth, H. Grabner, A. Starzacher, H. Bischof, and B. Rinner. Visual on-line learning in distributed camera networks. In *Proc. ICDSC*, 2008.

[11] C. Leistner, A. Saffari A. A., P. M. Roth, and H. Bischof. On robustness of on-line boosting - a competitive study. In *Proc. IEEE On-line Learning for Computer Vision Workshop*, 2009.

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

[13] R. Liu, J. Cheng, , and H. Lu. A robust boosting tracker with minimum error bound in a co-training framework. In *Proc. ICCV*, 2009.

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

[15] K.-K. Sung and T. Poggio. Example-based learning for view-based face detection. *IEEE Trans. PAMI*, 20(1):39–51, 1998.

[16] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3:323–344, 1987.

[17] Q. Zhu, S. Avidan, and K.-T. Cheng. Learning a sparse, corner-based representation for background modelling. In *Proc. ICCV*, volume I, pages 678–685, 2005.