

IMPROVING PERSON DETECTION USING SYNTHETIC TRAINING DATA

Jie Yu, Dirk Farin, Christof Krüger

Corporate Research Advance Engineering Multimedia
Robert Bosch GmbH
Hildesheim, Germany

Bernt Schiele

Computer Science Department
TU Darmstadt, Germany

ABSTRACT

Person detection in complex real-world scenes is a challenging problem. State-of-the-art methods typically use supervised learning relying on significant amounts of training data to achieve good detection results. However, labeling training data is tedious, expensive, and error-prone. This paper presents a novel method to improve detection performance by supplementing real-world data with synthetically generated training data. We consider the case of detecting people in crowded scenes within an AdaBoost-framework employing Haar and Histogram-of-Oriented-Gradients (HOG) features. Our evaluations on real-world video sequences of crowded scenes with significant occlusions show that the combination of real and synthetic training data significantly improves overall detection results.

Index Terms— Person detection, 3D model, synthetic training samples

1. INTRODUCTION

Learning approaches to object detection require a large number of training images to achieve good detection rates. These training sets consist of positive samples of the objects to be learned and negative samples of pure background. While the background data is usually easy to collect, the positive samples require significantly more manual work. In fact, it is not sufficient to collect images with positive samples, but these images also have to be aligned in position and scale. Since this is a tedious manual process, learning algorithms like MILBoost [1] have been proposed that can cope with inaccurately aligned input data. However, this approach can only reduce the work for aligning the positive samples, but still requires positive samples with a large variation in object appearances. This variation is difficult to collect, because it requires capturing images at various places under varying illumination conditions and with many different object instances.

A different way to approach this problem is to supplement the manually annotated training data with synthetically generated data. The advantage of synthetically generated data is that a virtually infinite amount of data can be generated,

which also spans a huge variation in object appearance. Finally, with synthetically generated data, annotations can be obtained automatically in the data-generation process. The only manual work that remains is to define 3D models of objects and optionally articulation data, from which the synthetic images are rendered.

Synthetic training data has been used before, e.g. for improving face-recognition applications [2, 3], which have to train detectors for specific faces while only a very limited number of sample images (e.g. three) are available for each face. One approach, for example, is to project these sample images onto a generic 3D face-model, vary the illumination and pose, and use these images as input to train the recognition system. In [4], a synthesized dataset is used to learn the human pose. Another example for using synthetic training data is for improving handwriting recognition [5].

In this paper, we first give a short overview of our person-detection system (Section 2), before we describe our framework for generating synthetic training data (Section 3). Finally, we evaluate variations of detectors, trained with real and synthetic data of different configurations (Section 4).

2. DETECTION FRAMEWORK

We consider the application of surveillance systems, monitoring persons in densely crowded scenes. The cameras are assumed to be mounted at a sufficient height, so that typically the head-and-shoulder part of most persons is visible. For this reason, we have chosen to detect persons with a head-and-shoulder model.

As detection framework, we use a cascade of AdaBoost classifiers [6]. The cascade consists of several stages constructed to reject many of the non-object patches in each stage, while accepting almost all positive patches. Each stage is a strong classifier of boosted weak classifiers. Because a cascade consists of classifiers of increasing complexity, most non-object patches are sorted-out in early stages, resulting in a fast overall detector, which is suitable for object detection with a sliding-window approach.

The features used by weak classifiers are another important component. They should be discriminative and efficient

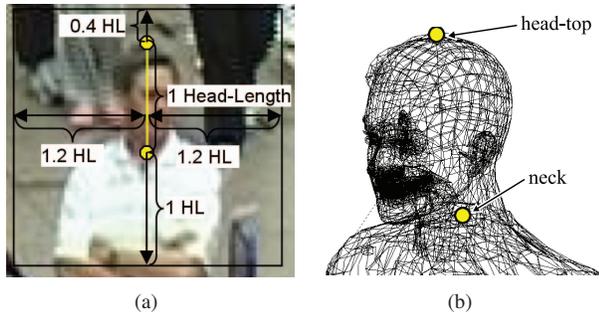


Fig. 1: (a) Example of a head-and-shoulder patch, defined by the head-top and neck key-points. The bounding box is placed around these two points as shown in the figure. The unit HL equals the distance between the two key-points. (b) The virtual key-points are specified in the 3D person model.

to compute. Haar-like rectangle features [6] and Histograms-of-Oriented Gradients (HOG) [7] have shown to be good choices for person detection. A further study in [8] shows that a multi-feature approach provides even better results. In this paper, we use a combination of Haar-like features and HOG features to train the weak classifiers.

The patches are defined such that they contain the head-and-shoulder region as well as some context around this part. In order to get accurate annotations for the training, we annotate heads with two key-points: the head-top and neck points. With these two points, the head-position and -length (HL) can be determined accurately and the bounding-box for the patch is derived. An example of the estimated head-and-shoulder bounding box based on the key-point annotation is shown in Fig. 1(a). The patch resolution was fixed to 36×36 pixels, since this resolution was sufficient for good detection results.

3. SYNTHETIC DATA GENERATION

The synthetic training data is obtained by superimposing a rendered image of a 3D person mesh on top of a real-world background image. A multitude of training images can be generated by randomly varying parameters of the 3D model (e.g., textures, pose), the camera pose, and lighting. The training-image formation process is shown in Fig. 2; the following sections describe each processing step in more detail.

3.1. 3D Person Model

The 3D person model has been generated with the MakeHuman tool [9]. This tool enables to build 3D meshes of persons for a given set of parameters for articulation and body shape. The articulation of the person model has been kept static as articulation should not be crucial for our head-and-shoulder detector and it keeps the model simple.

In a 3D editor, we additionally altered the exported person model by adding some simple hair styles and clothing,



Fig. 2: The 3D model is varied randomly and passed on to the OpenGL renderer. The output is merged with a randomly chosen background patch in the image composition stage.

optional glasses, and backpacks. The texture of the shirt, the hair-style, and the skin and hair colors are chosen from a set of predefined simple designs. Initially, we designed 7 shirt textures, and 6 hair styles each in one of 6 different hair colors (Fig. 3). Furthermore, we added the two virtual key-points *head-top* and *neck* to the 3D model (Fig. 1(b)). These will enable us subsequently to automatically derive the annotations for the rendered image.

The 3D model is scaled independently in the width and depth to simulate various body proportions. The height is not scaled, because the patch extraction is normalized to the head-length, which would undo the scaling of the height. Both scaling factors are chosen between 1.0 and 1.05. Furthermore, the person model is slightly tilted between -20° and 20° in arbitrary direction to simulate humans bending down or looking upwards.

3.2. Camera, Lighting, and Rendering

The virtual camera is directed at the scene like a typical surveillance camera. To match our set of background images (see Fig. 2), we use a fixed field of view of 45° , and a distance to the world origin of 3 meters. The elevation angle is varied between 10° and 20° , while the azimuth angle (around the object) is equally distributed over the full 360° . The scene is illuminated by two point light sources placed in the upper hemi-sphere. Finally, the final 3D person model is rendered using OpenGL along with an alpha-mask for seamless blending with a background image.

3.3. Background patch selection

The backgrounds to be placed behind the rendered person model are extracted from a database of real-world images recorded in a busy railway station. We do not care whether the image only contains background or there are also persons, because people will also not be isolated from each other in real-world scenarios. The size of the rendered person has been manually adapted to match the size of the other persons in the background. However, this size-adaption could be easily automated by employing camera calibration information.



Fig. 3: The top row shows examples of generated synthetic images. The rows below show the included hair styles and shirt textures.

Since the background image is chosen randomly, the superimposed virtual person will not always appear to be standing on the ground plane realistically. Often, it will look intuitively incorrect (Fig. 3), but since only the top part of the body is considered in the detector, we believe that this has little impact on the performance.

3.4. Image composition

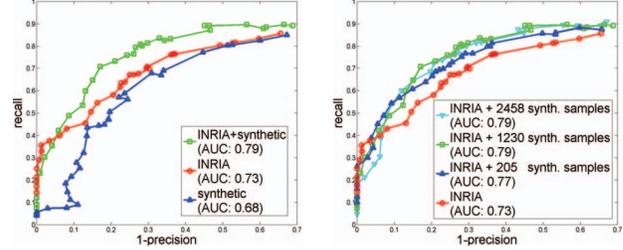
The image composition stage combines a real background image with the rendered person image. The contrast of the foreground object is adapted to match the background lighting conditions. Camera blur, to the extent present in the background, is simulated by smoothing the foreground object and its alpha-mask.

By projecting the two virtual key-points into the 2D image, we automatically obtain the annotation for the object and can extract the patch for training. Finally, we shift the extracted patch randomly within ± 2 pixels.

4. EXPERIMENTS

In our experiments, we have varied the parameters for camera pose, tilt angle, scaling, lighting, textures, hair/skin colors, hair-styles, and accessory items (backpacks, glasses). The parameters were chosen randomly and uniformly within the empirically specified ranges (Sec. 3). The smoothing and contrast adaptation was kept fixed for the dataset.

For the real training dataset, the INRIA dataset without mirrored images [7] was used (1207 persons). The INRIA dataset was additionally annotated with the two key-points to extract head-shoulder patches. We evaluate the detectors on 49 frames (451 persons) taken from 8 video sequences of crowded scenes, e.g. the PETS2007_S04_V1 sequence [10] is



(a) Purely real, purely synthetic, and combination of both sets. (b) Combination of datasets with different number of synthetic data.

Fig. 4: Comparison of detectors trained on INRIA, synthetic and combined datasets. The AUC (Area Under the Curve) is used to give a summary statistic.

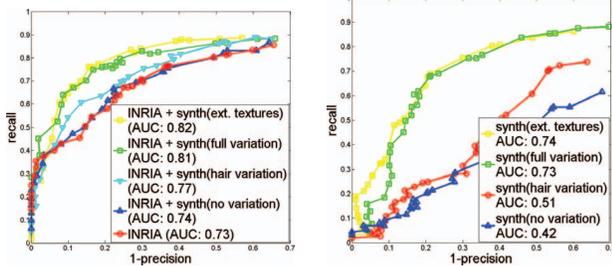
included. Note that the generated synthetic samples are only used for detector training, but not in the evaluation.

In the first experiment, we compare the performances of detectors trained on only the INRIA dataset, trained on only synthetic data (1230 samples), and trained on both datasets combined. Precision-Recall curves of the detection results are shown in Fig. 4(a). The AUC (Area Under the Curve) is used to give a summary statistic. These results show that by combining the real samples with synthetic samples, the overall performance is improved, except for precisions > 0.95 . On the other hand, the detector trained on only synthetic samples is worse than the detector trained on the real dataset. Especially for low recall values, the precision does not reach the expected 100%. This is probably due to some bias of the synthetic person-model compared to the real person-model.

In the second experiment, we consider only the combination of a real dataset with synthetic data and vary the number of synthetic samples added to the real data (Fig. 4(b)). The performances of all combined datasets are significantly better than the real dataset only, quickly improving the results after adding only a few hundred synthetic samples. However, when adding more synthetic data, performance seems to saturate.

The following experiments evaluate which factors in the generation of synthetic samples are critical for the detector performance. We conducted different experiments varying the object textures, camera motion, small geometric transformations, and camera blur to generate different training sets. The synthetic set generated without additional constraints is taken as the reference dataset, i.e. the parameters are equivalent to the dataset used in the previous experiments.

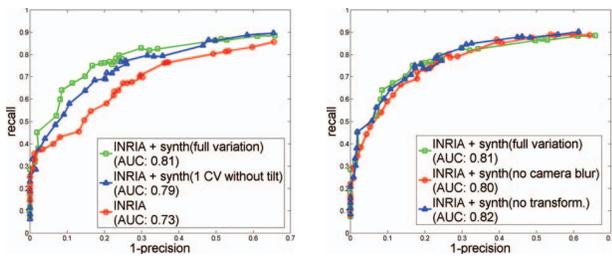
First, we evaluate the influence of the variation of the object textures. As described in Section 3, there is a set of predefined texture designs to increase the variability of the generated samples, such as shirt textures, hair styles, hair colors etc. In this experiment, we use three different synthetic datasets, each with 820 synthetic samples. Each dataset has different restrictions on the numbers of shirt textures, hair styles and hair colors. The reference dataset is with full variation (7 shirt textures, 6 hair styles, 6 hair colors). The second dataset is re-



(a) Results of detector trained on combined datasets.

(b) Results of detector trained on purely synthetic datasets.

Fig. 5: Comparison of synthetic datasets with varied numbers of shirt textures, hair styles and hair colors.



(a) Varied camera views.

(b) Blur and transformations.

Fig. 6: Comparison of synthetic datasets with varied options of camera views (CVs), camera blur, object transformations.

stricted to only one shirt texture, but all hair styles and colors, and the third dataset includes only a single shirt texture, hair style and color. The results in Fig. 5 show that the performance depends significantly on the appearance variability of the model. In order to see whether we could improve performance with even more variability, we added 9 more shirt textures to the 7 already present in the “full variation” set (see “ext. textures” in Fig. 5). However, we could observe a saturation effect.

Next, we evaluate the influence of the number of camera views. Compared to the reference dataset, another dataset was generated by fixing the azimuth angle of the camera, i.e. only a side-view of the person is used. Besides, the tilt angle of the person model is also deactivated, as it would have similar influence on the rendered image as the azimuth angle. Figure 6(a) shows that the performance degrades if the training set is limited to side views.

Finally, we evaluated the influence of the camera blur parameter and small geometric object transformations (scaling of object, tilt angle, and 2D in-plane shifts). The first dataset again is the reference dataset with all variations. For the second dataset, camera blur was disabled, and in the third dataset all small geometric transformations were disabled. As shown in Fig. 6(b), all of these parameters seem to have only minor influence on the detection performance.

Summarizing the experiments, we conclude that the best

performance can be obtained by combining synthetic training data with a real dataset. The most important parameters appear to be the variation in object appearance (textures), the number of viewing angles, and the number of samples. However, increasing the number of samples does not increase the detector performance without limit. The small geometric transformations and camera blur turned out to have little impact on the performance.

5. CONCLUSIONS

This paper presented an approach for generating synthetic training data for person detection. The synthetic data is used to complement the manually annotated real data, effectively reducing the amount of manual work needed for annotation. Our results showed that the detection results could in fact be significantly improved compared to a detector trained with real data only. However, we also showed that training on only synthetic data does not reach high precision. Hence, it appears sensible to extend real-world datasets with synthetic data to boost their performance with little extra work. Since we still use a quite simple model without articulation and advanced rendering, we expect that improving the variability of the synthetic person-model would lead to even better performance. This is an important direction of future work.

6. REFERENCES

- [1] B. Babenko, P. Dollár, Z. Tu, and S. Belongie, “Simultaneous learning and alignment: Multi-instance and multi-pose learning,” in *ECCV*, 2008.
- [2] J. Huang, B. Heisele, and V. Blanz, “Component-based face recognition with 3d morphable models,” in *AVBPA*, 2003, pp. 27–34.
- [3] X.G. Lu, R.L. Hsu, A.K. Jain, B. Kamgar Parsi, and B. Kamgar Parsi, “Face recognition with 3d model-based synthesis,” in *ICBA*, 2004, pp. 139–146.
- [4] R. Okada and S. Soatto, “Relevant feature selection for human pose estimation and localization in cluttered images,” in *ECCV*, 2008, pp. II: 434–445.
- [5] T. Varga and H. Bunke, “Comparing natural and synthetic training data for off-line cursive handwriting recognition,” in *IWFHR*, 2004, pp. 221–225.
- [6] Paul Viola and Michael Jones, “Robust real-time object detection,” *IJCV*, vol. 57, no. 2, pp. 137–154, 2002.
- [7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, 2005, vol. 1, pp. 886–893.
- [8] Christian Wojek and Bernt Schiele, *A Performance Evaluation of Single and Multi-feature People Detection*, vol. 5096 of *LNCS*, Springer, 2008.
- [9] MakeHuman project, “Open Source tool for making 3D characters,” <http://www.makehuman.org>.
- [10] UK EPSRC REASON Project, “PETS 2007,” <http://pets2007.net/>.