A Local Discriminative Model for Background Subtraction

Adrian Ulges[1], Thomas M. Breuel[1,2]

[1] Department of Computer Science, Technical University of Kaiserslautern
a_ulges@informatik.uni-kl.de
[2] Image Understanding and Pattern Recognition Group
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern
tmb@iupr.dfki.de

Abstract. Conventional background subtraction techniques that update a background model online have difficulties with correctly segmenting foreground objects if sudden brightness changes occur. Other methods that learn a global scene model offline suffer from projection errors. To overcome these problems, we present a different approach that is *local* and *discriminative*, i.e. for each pixel a classifier is trained to decide whether the pixel belongs to the background or foreground. Such a model requires significantly less tuning effort and shows a better robustness, as we will demonstrate in quantitative experiments on self-created and standard benchmarks. Finally, segmentation is improved by 18 % by integrating the probabilistic evidence provided by the local classifiers with a graph cut segmentation algorithm.

1 Introduction

Motion-based segmentation in static scenes is targeted at separating moving foreground objects from a static background given a fixed camera position and focal length. For this purpose, a number of background subtraction techniques exist that construct a model of the static scene background and label regions not fitting this model as foreground regions.

Using these methods, background subtraction systems achieve fair segmentation results but do not react properly to sudden intensity changes due to camera gain control, light switches, shadows, weather conditions, etc.. On the one hand, systems should adapt to such phenomena, while on the other hand reliably detecting foreground objects. Our practical experience has been that an "online" adaption as it is proposed by most approaches in the literature is error-prone, that the associated parameters (like feature weights or adaption rates of the background model) are difficult to tune, and that robustness is hard to achieve.

An alternative is to learn a background model during a separate learning phase in the absence of foreground objects (i.e., "offline"). In this way, scene properties can be modeled like weather changes and different light sources as well as characteristics of the camera like gain correction and noise. While conventional global methods following this strategy fail in the presence of pronounced

foreground objects, we propose a *local* and *discriminative* model based on classifiers deciding whether a pixel belongs to the background or foreground. Our contributions are: (1) a background subtraction approach that is simple and – in contrast to conventional methods we tested before – does not require much tuning, (2) quantitative experiments demonstrating that the local discriminative approach performs competitive to hand-tuned state-of-the art segmenters, and (3) it is shown how the probabilistic evidence given by local classifiers can be fused to an improved segmentation using a graph cut algorithm.

2 Related Work

The majority of background subtraction techniques proposed in the literature maintain a pixel-wise background model that is adapted online to illumination changes caused by varying weather conditions, camera gain control, light sources switched on and off, shadows, and background motion like waving trees (for reviews, see [9, 6]). Starting with the work by Wren et al. [12], several ways have been proposed to model the distribution of pixel intensity x given the fact that the pixel belongs to the background, $p(x|b)$. For this density, parametric approaches have been proposed using Gaussians [12] and mixtures of Gaussians [15] as well as non-parametric techniques like kernel densities [10, 11]. $p(x|b)$ is then used for segmentation by thresholding with the difference between background model and observation [18], or by integrating it in a Bayesian decision framework to compute $P(b|x)$ (e.g., [16, 17]).

To adapt to changes of the environment, most systems perform updates of the background model online, i.e. while segmentation is running. To work robustly, these heuristic updates must adapt properly to sudden scene changes while at the same time detecting non-background regions, which makes them error-prone and difficult to tune. To overcome this problem, it has been suggested to use features that are robust to illumination changes, like the gradient direction [8], shadow models [13], and color co-occurrence [7]. Our experience has been that problems with online updates can be overcome to some degree using proper features and careful tuning, but the system is not truly robust. Once segmentation fails, background models tend to be corrupted by foreground regions. Also, scene parts covered by the object cannot be updated properly.

To better adapt to scene changes, an alternative is to learn a model of the scene *offline*, i.e. during a separate learning phase in the absence of foreground objects. The most popular method based on this idea are "Eigenbackgrounds" [14] which view images as vectors of pixel values and perform a global Principal Component Analysis (PCA) decomposition on image level. While we adapt the strategy of learning a scene model offline, it will be shown that a global approach like PCA fails in the presence of large foreground objects. Instead, our model is based on local discriminative patch classifiers.

Other approaches have followed the idea of local classifiers before. Recently, Culibkr et al. [3] have proposed a related approach, in which frequently occurring patterns of pixel features are stored in a Radial Basis Function Network (RBF).

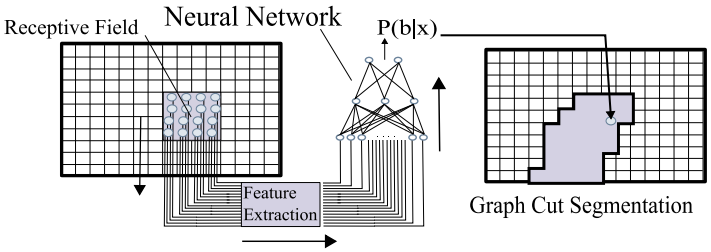


Fig. 1: The proposed setup: Each pixel is associated with a receptive field from which simple features like pixel intensity or gradient strength are extracted. Those form the input of a neural network classifier, which estimates the background posterior $P(b|x)$. These values are finally fed to a graph cut algorithm that determines the segmentation result.

The replacement and update of weights, however, are done using heuristic rules similar to the ones for standard online approaches [15]. In contrast to this work, the classifiers proposed here are trained in a discriminative manner by minimizing classification error.

Closest to our work are systems that truly train patch classifiers based on local information: Criminisi et al. proposed to integrate multiple cues such as motion and color in a Conditional Random Field (CRF) framework using tree-based classifiers [4]. Grabner et al. [5] use an online boosting framework. The key problem with such approaches is the lack of non-background samples for training. While Criminisi's prototype is trained in a fully supervised manner, i.e. ground truth sequences with full segmentations are demanded, Grabner's framework assumes uniform distributions for foreground features. In contrast to both, we propose a third alternative, namely to synthesize virtual foreground samples for training.

3 Our Approach

Most background subtraction systems construct a generative model $p(x|b)$ for the value of a pixel x given the the fact that it belongs to the background. This can then be integrated in a Bayesian framework to obtain the posterior $P(b|x)$.

In contrast to this, we follow a discriminative strategy, i.e. $P(b|x)$ is directly estimated using a local classifier for each pixel x. Since training is carried out in the absence of foreground objects, "virtual" non-background samples are obtained by synthesizing them. Finally, it is demonstrated how the local posteriors $P(b|x)$ can be fused to a global segmentation using a smoothness prior and a graph cut algorithm for optimization.

System Setup: To estimate the background probability $P(b|x)$, a neural network classifier is used for each pixel x, which is associated with a squared receptive field in the surrounding of x (also referred to as a *patch*). From this receptive field, simple features are extracted, which can be (1) pixel gray values or

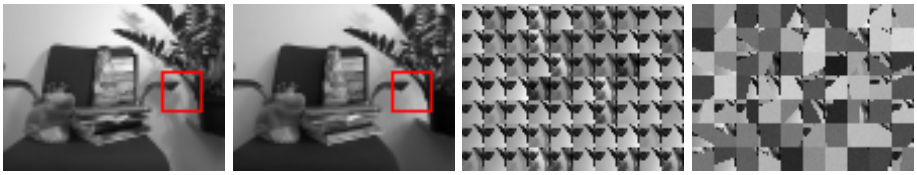


Fig. 2: Left: Two images of a static scene from the `Office` dataset with a highlighted patch (red). **Right**: Background and foreground Samples of the highlighted patch for training the associated classifier. Virtual foreground Samples have been synthesized by covering background samples with random synthetic texture.

color values, (2) gradient strength, or (3) the chroma components in YUV space, a description that has proven robust to illumination changes. These features form the input of a Multilayer Perceptron (MLP) with 1 hidden layer, which estimates $P(b|x)$ [2]. These posteriors can be used for a local, pixel-wise decision by thresholding them, or they can be integrated with a Gibbs prior to obtain smooth region boundaries. See Figure 1 for an illustration of the proposed system architecture.

Training: Most conventional background subtraction systems only maintain a model for the background, which is "trained" by heuristic update rules in a fully unsupervised manner. Our hypothesis is that a better distinction between foreground and background can be achieved if modelling both classes in a discriminative fashion. While background samples can be acquired during an offline learning phase, for the foreground uniform distributions have been assumed [5], or systems have been trained on fully segmented images [4].

We propose a third alternative, namely to synthesize virtual training samples for the foreground. In this paper, these training samples are constructed by simply covering the receptive field of a pixel partially or fully with non-background texture (for this, a random color was chosen and Gaussian noise with standard deviation 10 was added). Both background and foreground samples of a training patch are illustrated in Figure 2. Given such samples, the MLP training is carried out using plain backpropagation [2], whereas a fixed learning rate (0.2) and number of epochs (50) are used.

Integration with Graph Cut: Since the scores given by the local MLP classifiers have a probabilistic interpretation as $P(b|x)$, they can be integrated with a smoothness prior as described in the following. For background subtraction, a similar formulation has been proposed before in [4] and combined with so-called "background attenuation" (e.g., [16]).

For an image with pixels $x_1, .., x_n$, the Boolean background variables $b_1, .., b_n$ are determined according to a MAP formulation:

$$\begin{aligned}(\hat{b_1}, .., \hat{b_n}) &= \arg\max_{b_1,..,b_n} \quad P(b_1, .., b_n | x_1, .., x_n) \\ &\propto \arg\max_{b_1,..,b_n} \quad \prod_i \left(\frac{P(b_i|x_i)}{P(b_i)} \right) \cdot P(b_1, .., b_n)\end{aligned}$$

(a) (b) (c) (d) (e)

Fig. 3: A sample result on the `Office` dataset. The Eigenbackgrounds method projects the input image (a) to a too dark background image (b), and thresholding the difference (c) between the observation (a) and background image (b) obviously leads to segmentation errors. In contrast, the posterior $P(b|x)$ returned by our approach (d) gives a good segmentation (e) if integrated with a graph cut technique.

Further, it is assumed that the prior $P(b_1, .., b_n)$ is a Gibbs distribution: If denoting all pairs of 8-connected neighbor pixels with $\mathcal{C}$, we define:

$$P(b_1, .., b_n) \propto \prod_{(x_i, x_j) \in \mathcal{C}} e^{-U(i,j)} \tag{1}$$

with clique potentials $U(i, j) = 0$ (if $b_i = b_j$) and $U(i, j) = \nu$ (else), i.e. the length of the foreground object boundary is penalized with a constant bias. Note that it follows that $P(b_1) = ... = P(b_n) = \frac{1}{2}$. If taking the logarithm, the MAP solution thus minimizes an energy function consisting of a "data fit" term and a "smoothness term":

$$(\hat{b_1}, .., \hat{b_n}) = \arg\min_{b_1,..,b_n} \underbrace{-\sum_i \log P(b_i|x)}_{\text{“}datafit\text{''}} + \nu \cdot \underbrace{\sum_{(x_i,x_j)\in\mathcal{C}} (1 - \delta(b_i, b_j))}_{\text{“}smoothness\text{''}} \tag{2}$$

Via the "smoothness parameter" ν, both constraints are weighted relative to each other. Minimization is carried out efficiently using a graph cut algorithm [1]. The effect of this additional smoothness constraint is illustrated in Figure 3: While the local posteriors give a noisy result with some local misclassifications (d), these are overruled by the smoothness constraint (e).

4 Experiments

In the following experiments, we first analyze the influence of several internal parameters of our system on a self-created dataset. Second, comparisons with other state-of-the-art methods on both self-generated and public benchmarks are presented that demonstrate the competitive performance of the local discriminative approach.

4.1 Experiments on `Office` Dataset

The following evaluations have been done on the self-created *Office* dataset[3]. This benchmark consists of 497 training images taken over several days that show a

[3]The dataset is publicly available at http://www.iupr.org/downloads/data

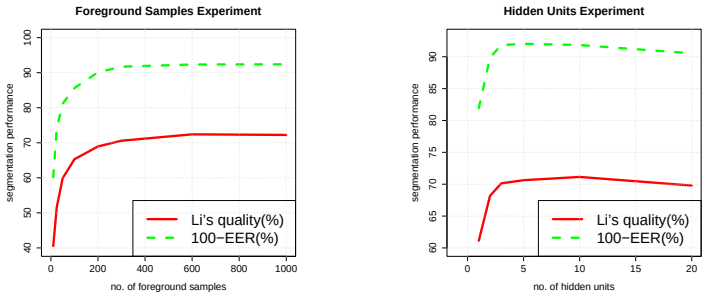


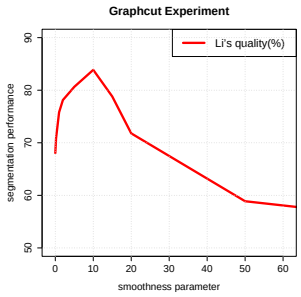
Fig. 4: Experimenting with internal parameters of the system: a saturation of the performance is found for about (a) 300 virtual foreground samples and (b) 5 hidden units.

complex scene in an office environment with several light sources, shadows, and slight background motion of a plant (see Figure 2 for two sample pictures). For testing, objects were presented to the camera in 6 short sequences taken at about 1 fps.. These test sequences represent difficult situations with shadows cast by the objects, abrupt light switches, and objects in varying distance to the camera. For 90 randomly sampled frames, ground truth segmentations were provided manually.

A Sample Result: We start with a first qualitative result that compares the local discriminative approach with the global Eigenbackgrounds approach (quantitative results will be given later in Figure 5). Given a test frame with an object held close to the camera (Figure 3, the Eigenbackgrounds approach projects the input image to a low-dimensional Eigenspace and thresholds the distance between the observation and the projection. Due to the influence of the large foreground object, the input (a) is projected to a too dark background image (b), and thresholding with the resulting distance (c) obviously gives a poor segmentation. In contrast, since the proposed pixel classifiers are local, the foreground region has negligible influence on the rest of the image, and the resulting posterior $P(b|x)$ (d) allows for a proper segmentation (e) when integrated with a graph cut as outlined in Section 3.

Number of Synthesized Foreground Samples: A core idea of the proposed approach is to synthesize samples for the non-background class. In this experiment, we tested how many such samples are needed per patch. The system was trained and tested on the data described above, whereas images were scaled to a width of 80 pixels. Note that the local discriminative approach scales linearly with the number of image and patch pixels - since the proposed approach proved robust to downscaling in our experiments, it was decided to use subsampled images such that the system runs in near-realtime at 5 fps. on a 2.4 GHz Opteron processor.

As features, chroma components in YUV color space were used with a patch width of 7 and an MLP with 5 hidden units.



method / features used	Li's Quality	EER
PCA	52.04	18.22
ONLINE	63.23	—
gray	63.56	17.61
grad	67.20	6.99
uv	71.41	8.09
grad+uv	69.76	6.25
uv + graph cut	83.87	—

Fig. 5: Left: When using graph cut segmentation with a proper smoothness parameter ν, the segmentation quality (Li's measure) can be improved significantly from 71 % (best pixel-wise thresholding) to 84 %. **Right**: Overall results for baseline methods and our system with different feature types on the *Office* dataset.

For the assessment of segmentation quality, two different measures were used:

1. the equal error rate (EER) for foreground and background pixels
2. the quality measure used by Li et al. [17]: If the foreground regions in the result and ground truth mask are denoted with A_t and B, the segmentation quality is defined as $S(A_t.B) := \frac{A_t \cap B}{A_t \cup B}$. Note that the result A_t can be obtained by a graph cut algorithm or by locally thresholding the posterior $P(b|x)$. In the latter case, we choose the threshold t that maximizes S on the test set.

Figure 4 plots both segmentation quality measures against the number of synthesized samples. As expected, the segmentation quality increases with the number of samples, but it converges against an optimum approximately reached at 300 samples (this number will be used in the following experiments).

Number of Hidden Units: A similar experiment was done for the number of hidden units, whereas the setup from the last experiment was copied and the number of foreground samples per patch was set to 300. Our results illustrated in Figure 4 show that the overall performance of the system depends less on the number of hidden units than it does on the number of training samples. Also, we find the optimal performance for about 5 hidden units (when increasing this number further, the performance drops slightly due to overfitting).

Influence of Graph Cut: As outlined in Section 3, one key feature of the proposed framework is the probabilistic integration of posteriors with a graph cut optimization. In this experiment, we study quantitatively whether this optimization actually improves the overall performance of the system (the setup from the previous experiment was kept). In Figure 5, Li's quality measure is plotted against the smoothness parameter ν from Equation (2). The top performance can be observed at $\nu = 10$. More interesting, however, is that the overall segmentation performance is improved to 84 %, i.e. by 18 % relative to the best pixel-wise thresholding with the posterior (ca. 71 %, as can be seen in Figure 4).

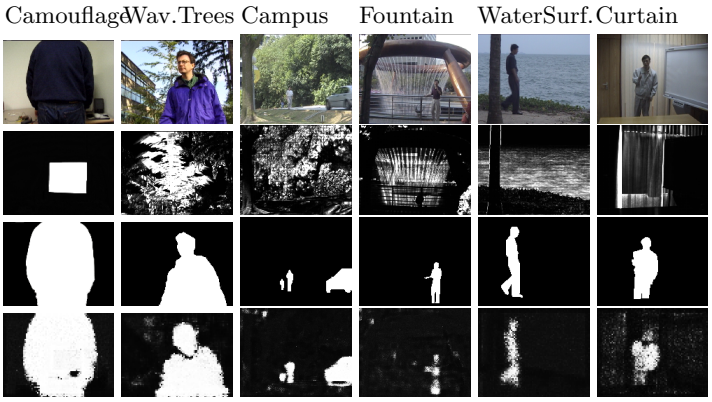


Fig. 6: Sample results on public datasets as reported in [17, 18]. Row 1 shows test frames, row 2 variance images that illustrate where motion occurs. Ground truth masks are given in row 3, and the bottom row shows the posterior given by our approach.

Comparison with other Methods: Finally, we compare the proposed framework with two other methods, namely Eigenbackgrounds [14] ("PCA") and a thoroughly tuned implementation of a state-of-the-art online background subtraction approach ("ONLINE"). This method weighs two background models, namely a shadow model [13] and histograms of gradient directions [8], and integrates them with a background attenuation and graph cut optimization as in Sun's Background Cut [16].

Quantitative results for both methods as well as for the proposed approach tested with several features are illustrated in Figure 5 (the same setup was used as in the experiments before). Our approach gives a statistically significant improvement compared to PCA, which is revealed by a paired t-test and corresponds to the observations made in Figure 3.

The online algorithm performs comparably to our system when using gray pixel values ("gray") – it has particular problems with light switches and strong gain control (note that test sequences were taken at only 1 fps., which simulates sudden light changes and stresses the online system).

The local discriminative approach does even better when using more robust features like gradient strength ("grad"), chroma information ("uv"), or both ("grad+uv"). Finally, the top performance is achieved when integrating the posteriors with a graph cut optimization.

4.2 Experiments on Other Datasets

In this experiment, the proposed framework is tested on publicly available benchmark sequences from the literature representing difficult situations with motion in the background, like waving trees, flickering monitor sequences, and water surfaces. We demonstrate that our approach is capable of learning an adequate background model in such situations and show competitive results.

Table 1: Quantitative results on public datasets from [18, 17]

	Error Rate (from [18])	
sequence	our approach	best res. in [18]
Camoufl.	5.41	9.54
W.Trees	5.31	5.02

	Li's quality (from [17])		
sequence	our approach	Li's system	MOG
Campus	60.01	68.3	48.0
Fountain	54.87	67.4	66.3
W.Surf.	72.66	85.1	53.6
Curtain	40.51	91.1	44.5

From [18] we use the "WavingTress" and "Camouflage" Sequences, and from [17] "Campus", "Fountain", "WaterSurface", and "Curtain". Other sequences are available, but do not satisfy our need for a background-only training phase showing all lighting conditions that occur in testing.

The system was run by using a pixel-wise thresholding of the posterior, i.e. without graph cut integration. "uv" features were used with a patch radius of 3, 5 hidden units and 300 training samples. Images were scaled to a resolution of width 160 (for "Campus" and "Fountain", which show very small objects) or 80 (for all others). 200 training frames in the beginning of each sequence were used.

Some sample results are illustrated in Figure 6, and quantitative performance measures are given in Table 1, whereas we stick with the error measures from the corresponding publications. For [18], this is the rate of pixel errors (we use the equal error rate threshold). Our approach performs comparably ("WavingTrees") or significantly better ("Camouflage") than the best results reported in [18].

For the sequences from [17], we choose the threshold that optimizes Li's quality. Here, the local discriminative approach is outperformed by Li's for all sequences. Compared to a standard mixture-of-Gaussians (MOG) system, it performs better for "Campus" and "WaterSurface", comparable on "Curtain", and does worse for "Fountain". An in-depth analysis revealed that for the "Fountain" Sequence, our system reacts sensitive to a small camera shake during the test phase of the sequence.

5 Discussion

In this paper, we have presented a background subtraction approach based on training local discriminative classifiers that assign pixels to foreground and background. A competitive performance on self-created and standard benchmarks has been demonstrated. Further, our experience has been that the system is more robust and easier to tune than online algorithms implemented previously.

While segmentation with our prototype can be done near-realtime (for an image width of 80 pixels, an unoptimized implementation runs at 5 fps.), training takes significantly longer (about 1 sec. per pixel) and is thus not really suitable for online updates such as the system in [5]. Note, however, that our prototype can be parallelized by treating pixels separately.

An interesting open question is the influence of more realistic samples for the "foreground model" of the system. So far, a very simple sampling strategy has

been used (random colors with additive noise). It might be interesting to test whether our approach can do better with samples from real images, or even from foreground objects that are known to occur in the scene[4].

References

1. Y. Boykov, V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE PAMI*, 26(9):1124–1137, 2004.
2. R. Duda, P. Hart, D. Stork. Pattern Classification (2nd Edition). Wiley Interscience Publications, 2000.
3. D. Culibrk, O. Marques, D. Socek, H. Kalva, B. Furht. Neural Network Approach to Background Modeling for Video Object Segmentation. *IEEE Trans. Neur. Netw.*, 18(6):1614–1627, 2007.
4. P. Yin, A. Criminisi, J.M. Winn, I.A. Essa. Tree-based Classifiers for Bilayer Video Segmentation. In *CVPR 07*, pp. 1–8, 2007.
5. H. Grabner, P.M. Roth, M. Grabner, H. Bischof. Autonomous Learning a Robust Background Model for Change Detection. In *Int. Workshop Perf. Eval. Track. Surv.*, pp. 39–46, 2006.
6. S. Cheung, C. Kamath. Robust Background Subtraction with Foreground Validation for Urban Traffic Video. *EURASIP Appl. Sign. Proc.*, pp. 2330–2340, 2005.
7. L. Li, W. Huang, I. Gu, Q. Tian. Foreground Object Detection in Changing Background Based on Color Co-Occurrence Statistics. In *WACV 02*, pp. 269–274, 2002.
8. P. Noriega, O. Bernier. Real Time Illumination Invariant Background Subtraction Using Local Kernel Histograms. In *BMVC 06*, pp. III:979–988, 2006.
9. M. Piccardi. Background Subtraction Techniques: A Review. In *IEEE SMC/ICSMC*, Vol. 4, pp. 3099–3104, 2004.
10. B. Han, D. Comaniciu, L. Davis. Sequential Kernel Density Approximation through Mode Propagation: Applications to Background Modeling. In *ACCV 04*, 2000.
11. A.M. Elgammal, D. Harwood, L.S. Davis. Non-parametric Model for Background Subtraction. In *ECCV 00*, pp. 751–767, 2000.
12. C.R. Wren, A. Azarbayejani, T. Darrell, A. Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE PAMI*, 19(7):780–785, 1997.
13. T. Horprasert, D. Harwood, L.S. Davis. A Statistical Approach for Realtime Robust Background Subtraction and Shadow Detection. In *IEEE Framerate Workshop*, pp. 1–19, 1999.
14. N. Oliver, B. Rosario, A. Pentland. A Bayesian Computer Vision System for Modelling Human Interactions, *IEEE PAMI*, 22(8):831–843, 2000.
15. C. Stauffer, E. Grimson. Learning Patterns of Activity Using Real-Time Tracking, *IEEE PAMI*, 22(8):747–757, 2000.
16. J. Sun, W. Zhang, X. Tang, H.-Y. Shum. Background Cut, In *ECCV 06*, pp. 628–641, 2006.
17. L. Li, W. Huang, I. Gu, Q. Tian. Statistical Modelling of Complex Backgrounds for Foreground Object Detection. *IEEE PAMI*, 13(11):75–104, 2004.
18. K. Toyama, J. Krumm, B. Brumitt, B. Meysers. Wallflower: Principles and Practice of Background Maintenance. In *ICCV 99*, pp. 255–261, 1999.

[4]This work was supported in part by the Stiftung Rheinland-Pfalz für Innovation, project InViRe (961-386261/791)