Multi View Registration for Novelty/Background Separation

Omid Aghazadeh , Josephine Sullivan and Stefan Carlsson
Computer Vision and Active Perception laboratory
Royal Institute of Technology (KTH)
{omida,sullivan,stefanc}@csc.kth.se

Abstract

We propose a system for the automatic segmentation of novelties from the background in scenarios where multiple images of the same environment are available e.g. obtained by wearable visual cameras. Our method finds the pixels in a query image corresponding to the underlying background environment by comparing it to reference images of the same scene. This is achieved despite the fact that all the images may have different viewpoints , significantly different illumination conditions and contain different objects - cars, people, bicycles, etc. - occluding the background. We estimate the probability of each pixel, in the query image, belonging to the background by computing its appearance inconsistency to the multiple reference images. We then, produce multiple segmentations of the query image using an iterated graph cuts algorithm, initializing from these estimated probabilities and consecutively combine these segmentations to come up with a final segmentation of the background. Detection of the background in turn highlights the novel pixels. We demonstrate the effectiveness of our approach on a challenging outdoors data set.

1. Introduction

A mobile surveillance system or a person with a wearable camera often moves in a geographically limited environment over extended time periods. The problem of identifying the background pixels of a scene from this environment is an interesting and challenging one especially as the background will vary and be partially occluded by different temporary objects each time it is viewed. However, the ability to perform such background/novelty detection would greatly facilitate visual memory processing of wearable camera footage and the monitoring of areas with mobile surveillance systems. In this paper we focus on wearable camera footage and propose a system for detecting these temporary/novel objects in locations repeatedly visited by a person wearing a camera.

We consider images captured over time periods of days

Figure 1. Our system takes as input a query image and multiple reference images. We assume all these images are of the same environment taken from approximately the same view point but at different times. The algorithm segments out objects in the query image which are not part of the environment. The bottom right figure shows the computed segmentation.

and during this time both substantial nuisance and interesting variations can occur in the environment. Given a query image captured from a specific location on a certain day and reference images captured from previous days at approximately the same location, we aim to *distinguish between the novel and background pixels in the query image*.

This is not a trivial task. All the images examined are captured on different days and will have a potentially large variation in their illumination and shading in combination with relatively large variations in their viewpoints. And

also in each image there will be different temporary objects occluding the background. Therefore, it is not feasible to build one clean background image and perform background subtraction. We instead associate background pixels with those which can be consistently and reliably *matched* to the reference images. Our system has two main steps. The first probabilistically classifies each pixel in query image as background or not from appearance consistency features extracted from dense correspondences to the reference images. While the second stage is two-class segmentation of the query image guided by the output of the background classification and consistency of image appearance. Note that the system implicitly relies on the geometric constraints that the query and stored images are captured from approximately the same location.

This problem differs significantly from traditional problems of foreground background segmentation with stationary surveillance cameras where the main source of background variation is changes in illumination. Since we use static images widely separated in time we cannot exploit camera motion constraints as in [4]. Contrary to [14], our images neither allow 3D modelling of the background nor detailed geometric analysis [17] to be used. Our reliance on appearance matching and two class segmentation allows for robust exploitation of our highly varying background images. The use of multiple static images contrasts with segmentation methods using optical flow [1, 16] and allows us to segment out novelty that is not necessarily foreground with high disparity. Co-segmentation approaches [7], though related, are not suitable due to the significant appearance variations in the background - the object of constant appearance for co-segmentation - across the images.

The main contribution of the paper is a robust and generic novelty detection algorithm whose parameters are automatically learnt from annotated data. This allows for the detection of many time-varying scene components such as people, cars... in a robust way that mimics the performance of specifically designed object detection algorithms.

The organization of the paper is: In section 2 we introduce our method for novelty/background segmentation, in section 3 we quantitatively and qualitatively evaluate the proposed method and we conclude the paper in section 4.

2. Foreground/Background Segmentation

As previously stated we have a set of reference images and a query image taken of the same scene. All these images have been captured at different times and relatively different viewpoints. Our goal is to identify background pixels in the query image. This is achieved by summarizing comparisons of the query image to each reference image as follows:

1. Estimate the probability of each pixel **not** belonging to background which we term the *probability of novelty* from the dense correspondences found between the query image and each reference image.

2. Produce multiple segmentations of the query image, given the probabilities of novelty, by varying the parameter settings of the segmentation process.

3. Combine all the segmentations probabilistically to produce a final classification of the query image pixels.

We now describe each step in more detail.

2.1. Estimating the probability of novelty

Crucial to our algorithm's success is the computation of dense correspondences between the query image and each reference image. Establishing such correspondences, when each image has different parts of the scene occluded, is a hard problem. In fact establishing correspondences and occlusion estimation are closely related tasks - knowledge of the image correspondences makes estimation of the occlusions easier and vice versa.

Some authors have exploited this relationship by explicitly including occlusion estimation into their algorithms for finding image correspondences [12]. As such formulations usually rely on expectation-maximization like procedures, they are usually more susceptible to local minima. Therefore, occlusion estimation is usually ignored and more emphasis is instead put on imposing priors - such as smooth displacement fields - when calculating correspondences.

In this work, we do not aim to solve for both occlusions (which in our problem are mainly novelties) and the correspondences simultaneously. Instead, we aim to deduce the background pixels given some noisy correspondences between images. We use *SIFT Flow* [9] to establish such correspondences as we found it more robust to illumination changes, occlusions and large displacements compared to the methods we tried.

We first establish correspondences between the query image I_q and each reference image $I_r \in R$ where R is the set of reference images. Then, we compute the following features on each pixel of I_q using each I_r in turn:

$$\begin{aligned} I^{\text{err}}_{q,r,x} &= \|I_{q,x} - I_{r\to q,x}\| \\ S^{\text{err}}_{q,r,x} &= \|S_{q,x} - S_{r\to q,x}\| \\ H^{\text{err}}_{q,r,x} &= \sum_c QC^A_{0.5}\left(H(I_q,x,c), H(I_{r\to q},x,c)\right) \end{aligned} \tag{1}$$

where

- $I_{i,x}$ is the color (CIE Lab) of pixel x in image I_i,
- $S_{i,x}$ is the SIFT [10] computed at pixel x of I_i,
- $H(I_i, x, c)$ is the histogram of channel c intensity values of the pixels inside a rectangular region centered

Feature	Source	Properties of the Feature		
		Neigh.	Corr. Sens.	Illum Inv.
I^{err}	Color	0	1	0
S^{err}	Sift	1	1	1
H^{err}	Hist	1	0	0,1

Table 1. Features used in the estimation of the *probability of novelty* and their properties. **Neigh.** is 1 if the feature captures information in the neighborhood of a pixel. **Corr. Sens.** is 1 if the feature is affected considerably by small errors in the correspondences. **Illum Inv.** is 1 if the feature is invariant to illumination changes. Different normalizations of H^{err} can make it sensitive or invariant to illumination changes.

at pixel x in I_i and $QC_m^A(.,.)$ is the distance between two histograms computed using the Quadratic Chi kernel with respect to the parameter m and the similarity matrix A [13],

- $I_{r\to q}$ denotes image I_r warped towards I_q.

The measure H^{err} dubbed *Normalized Bagged Similarity* measures neighborhood similarity of pixels similar to Normalized Cross Correlation while unlike NCC it is invariant to the ordering of the pixels and also, it can be made invariant to nonlinear transformations of the intensities using proper histogram normalization techniques and proper similarity matrices (see supplementary material). NBS can be computed very efficiently by the use of Integral Histograms and its computations can be parallelized very efficiently by the use of GPUs. Table 1 describes the properties of the features and Figure 2 shows the features evaluated on an example case.

We compute these three measurement types at multiple scales and stack the resulting feature vectors into $\bar{F}_{q,r,x}$. We then compute for each pixel x at a fixed scale, the algebraic mean, harmonic mean and minimum of each response in $\bar{F}_{q,r,x}$ with respect to the reference images I_r. The resulting feature vector, $F_{q,x}$, for each pixel in I_q is 78 dimensional. This feature vector is used to estimate the probability of novelty as follows.

We use logistic regression to map a pixel's feature vector, $F_{q,x}$, to a scalar between 0 and 1 estimating the pixel's *posterior probability* of being not background. The parameters of this regression function are learnt from our manually annotated ground truth data (see Section 3) which provides many pixel feature vectors and their associated labelling as background or not. L_2 regularization is imposed during learning and LibLinear [6] is used to ensure training takes a reasonable time given the large number of training examples examined (approximately 3 million) which are collected by sub sampling the data every 6th pixel in each direction. In the rest of this paper, we refer to the results of this logistic regression (the *probability of novelty*) evaluated at pixel x in the image i with $\tilde{P}_i(x)$. Figure 4 (top left) depicts a typical evaluation of $\tilde{P}$ on a query image.

2.2. Segmenting out the background

Using the estimated probability of novelty $\tilde{P}$, we iterate between segmentation of the query image's pixels into background and novel regions and updating our models describing the features associated with the background and novel pixels. We do this in a manner similar to Grab Cut [15]. An important difference, though, is that we initialize our foreground and background models automatically from the probability maps indicated by $\tilde{P}$. This iterative process can be viewed as a variant of Expectation Maximization.

For the maximization step, we use an energy minimization approach to segment the images into novelty and background regions. We use Graph cuts [8, 2, 3] to perform the minimization as we use an appropriate energy function in the popular form of a sum of unary and pairwise terms.

$$E(l) = \sum_{x\in\mathcal{X}} D_x(l_x) + \lambda \sum_{(x,y)\in\mathcal{N}} V_{x,y}(l_x, l_y) \qquad (2)$$

where l is a binary labelling assigning each pixel $x \in \mathcal{X}$ a label $l_x \in \{0, 1\}$. Here $D_x(l_x)$ is the data term and determines the cost of assigning the label l_x to pixel x in image I. $\mathcal{N}$ is the set of pairs of neighbouring pixels (8 connectivities) and $V_{x,y}(l_x, l_y)$ is the pairwise smoothness (regularization) term and determines the cost of assigning different labels to neighbouring pixels x and y.

A popular choice of the smoothness term is the Ising prior weighted by some dissimilarity measure to relax the smoothness constraint at image discontinuities. We utilize a similar approach and use a parallelized version [5] of the gPb detector [11] - which utilizes GPUs to estimate the boundaries of objects in natural images - to encourage the cut to go through those boundaries. Therefore, our pairwise term is

$$V_{x,y}(l_x, l_y) = \frac{1}{\|x - y\|} [l_x \neq l_y] \, e^{-\frac{|I_B(x) - I_B(y)|^2}{2\sigma^2}} \qquad (3)$$

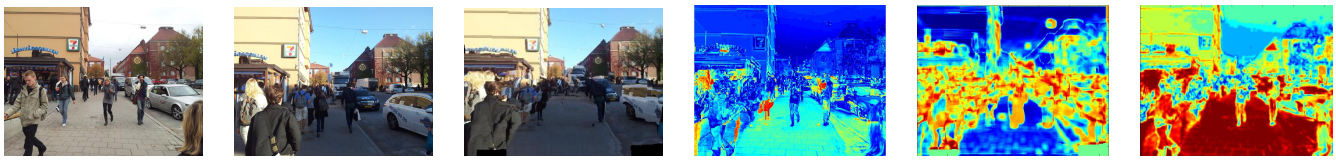
where $[.]$ is the Iverson bracket and $I_B(x)$ denotes the response of the gPb detector at pixel x.

We define the data term to be

$$D_x(l_x) = -\log P(l_x \mid f_x) \qquad (4)$$

where f_x is a feature descriptor of the pixel x and $P(l_x \mid f_x)$ represents the posterior probability of label l_x (novelty or background) conditioned on observing feature f_x. More details of how we estimate this posterior probability are now given.

Let $l^{(t)}$ represent the current best estimate of the pixel labellings. Define $\mathcal{X}_k^{(t)} = \{x \mid l_x^{(t)} = k\}$ for $k \in \{0, 1\}$ to



I_q I_r $I_{r\to q}$ $I^{\text{err}}_{q,r}$ $S^{\text{err}}_{q,r}$ $H^{\text{err}}_{q,r}$

Figure 2. The features used to calculate the probability of novelty $\tilde{P}$ for pixels in the query image I_q when compared to a reference image I_r. $I_{r\to q}$ is I_r warped towards I_q using SIFT Flow. The corresponding pixels of I_q and I_r are then compared via $I_{r\to q}$ as in equation (1).

be the set of pixels with label k according to labelling $l^{(t)}$. For the expectation step, we collect some statistics about the distribution of some features in I_q conditioned on the current estimate of the segmentation[1]. The features we use for the segmentation are (a subset of) the color, a dimensionality reduced version of the sift feature vector (to 3 dimensions) and the position of each pixel. We use *Kernel Density Estimation* to estimate the likelihood $P(f_x \,|\, l_x)$

$$P(f_x \,|\, l_x) = \frac{1}{|\mathcal{X}^{(t)}_{l_x}|\, h^d} \sum_{y \in \mathcal{X}^{(t)}_{l_x}} K\left(\frac{f_x - f_y}{h}\right) \qquad (5)$$

where d is the dimensionality of the f_x (8 in case of all 3 features), h is the bandwidth(window width) and $K(\mu)$ is the multivariate Gaussian density function with identity covariance matrix evaluated at μ. Whitening the feature data is performed before any likelihood computations are made. We sub-sample the pixels on a fixed grid, evaluate a homogeneous KDE on the same subset of pixels and use bilinear interpolation to estimate the likelihood maps on all pixels. The KDE evaluation is quadratic in the number of (sub sampled) pixels and can be parallelized very efficiently by the use of GPUs. Evaluating the likelihood maps at each iteration takes around 1 second on an NVIDIA GTX 470 for a sub sampling of once every 3 pixels in both directions.

To convert the estimated likelihoods to posteriors, based on $\tilde{P}$, we consider three types of class priors for each pixel: a uniform prior (P_H) and two spatially varying priors (P_{SF} and P_S). Table 2 shows the details of these priors. The prior P_{SF} allows more deviation from the relatively noisy probability estimates in $\tilde{P}$ compared to P_S which strictly promotes the segmentation suggested by $\tilde{P}$. Note the way we define the posterior is different to [1] as we do not marginalize over model parameters but instead use a pixelwise prior computed from $\tilde{P}$. We then, use the negative log of the posterior $P(l_x \,|\, f_x) \propto P(f_x \,|\, l_x)P(l_x)$ as the data term:

$$D_x(l_x) = -\log\left(P(f_x \,|\, l_x)P(l_x)\right) + \log Z_x \qquad (6)$$

[1] In the first iteration, we collect statistics only from the pixels whose $\tilde{P}$ is more than a desired margin m_0 away from 0.5. This way, we can collect the initial statistics about segments with the desired level of certainty and avoid collecting statistics from uncertain regions if $m_0 > 0$.

Prior name	$P(l_x = 1) \propto$	$P(l_x = 0) \propto$
P_H	$\sum_x \tilde{P}(x)$	$\sum_x (1 - \tilde{P}(x))$
P_{SF}	$\tilde{P}(x)$	1
P_S	$\tilde{P}(x)$	$1 - \tilde{P}(x)$

Table 2. The three types of priors used for the labelling of a pixel.

Figure 3. Segmentation results with different class priors: from top left to bottom right: initializing with $m_0 = 0.1$ and segmentation results using P_H, P_{SF} and P_S class priors. In the figure illustrating the initialization, regions inside blue and red boundaries represent initial estimates of background and novelty regions. The margin $m_0 = 0.1 > 0$ on $\tilde{P}$ (refer to Figure 4) leads to gaps between the regions.

where the normalization factor is

$$Z_x = \sum_{k \in \{0,1\}} P(f_x \,|\, l_x = k)P(l_x = k) \qquad (7)$$

Figure 3 shows the different segmentations achieved using the different priors P_H, P_{SF} and P_S on the pixel labels. In this example the parameters were set to $m_0 = 0.1, \lambda = 5, h = 0.5$ and each f_x was composed of pixel x's color, dimensionality reduced sift representation and its position. We iterate between the expectation and maximization steps until the solution converges for a maximum of 25 iterations.

2.3. Combining Multiple Segmentations

The segmentation process of the previous section will converge to a stable segmentation. However, the final segmentation achieved will greatly depend on the setting of the explicit and implicit parameters in the energy function defined in equation (2). The explicit parameter corresponds to the regularization parameter λ, while the implicit parameters include the initialization margin m_0, the bandwidth of the KDE h in the likelihood function $P(f_x \mid l_x)$, the features extracted to define f_x and the prior used in the calculation of the posterior $P(l_x \mid f_x)$. For clarity let $\mathcal{S} = \{\lambda, h, \ldots\}$ denote the set of all the parameters which influence the segmentation process and $\mathbf{s}$ a vector containing the values assigned to each parameter in $\mathcal{S}$.

The question then is which $\mathbf{s}$ should we use when we segment a new image? We could potentially use the $\mathbf{s}$ which optimizes performance on a validation set. However, the choice made in this way will be highly influenced by the images in the validation set and how performance is measured and also the best parameter setting can vary drastically across individual images. Ideally, we want to perform multiple segmentations, corresponding to $\mathbf{s}_1, \ldots, \mathbf{s}_K$, and aggregate the results. One drawback of this approach is the extra computational cost if K segmentations must be performed and this becomes computationally impractical for a large K. Another issue is how to aggregate the results.

We propose the following solution. We start with a large pool $\{\mathbf{s}_1, \ldots, \mathbf{s}_K\}$ of parameter settings ($K = 50$ in the experiments). Each image in our training set is segmented K times, once for each $\mathbf{s}_k$. Then for a pixel x in a training image we get a binary vector of length K whose kth entry is l_x and l_x is its labelling returned by the segmentation process with parameter setting $\mathbf{s}_k$. We then, learn a logistic regression function with L_1 regularization which maps this binary vector to a probabilistic estimate of its ground truth labelling. The parameter controlling the regularization, in the regression learning, is set to ensure a sparse solution is found. An immediate consequence of this sparse solution is that only a small proportion of the original K segmentations need to be computed when a novel image is encountered. We denote the evaluation of this learnt logistic function on image i at pixel x with $\hat{P}_i(x)$. The top right image of 4 shows an example of a computed $\hat{P}(x)$.

The final segmentation of the query image is found by minimizing an energy function similar to 2 but with the data term based on $\hat{P}$:

$$\hat{D}_x(l_x) = \begin{cases} -\log\left(1 - \hat{P}(x)\right) & \text{if } l_x = 0 \\ -\log\left(\hat{P}(x)\right) & \text{if } l_x = 1 \end{cases} \quad (8)$$

We use Graph Cuts to minimize this energy globally[2]. The bottom left image of Figure 4 shows the final segmentation found for a query image.

[2]This minimization step is not iterated as the data term is fixed.

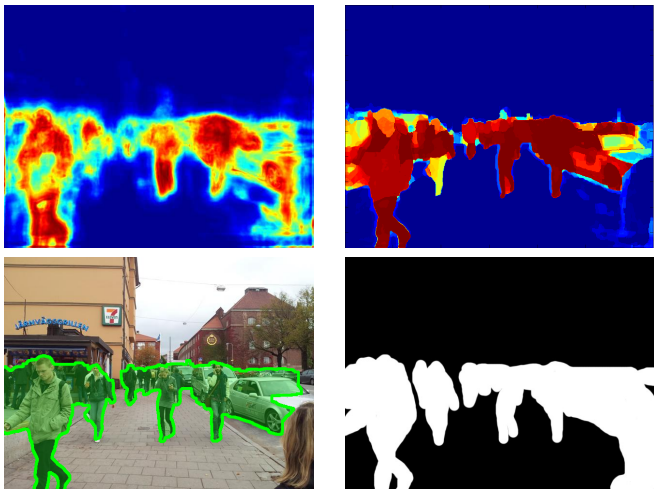


Figure 4. Evaluating the logistic regression function $\hat{P}$ combining multiple segmentation results and the final segmentation acquired from $\hat{P}$. From top left to bottom right: $\tilde{P}$, $\hat{P}$, final segmentation and the ground truth labelling.

3. Experiments

3.1. Data Set

Our data set consists of 12 images of 12 different places making a total of $12 \times 12 = 144$ images. Figure 5 shows 3 images of one of the places from our data set. Note that as the images of the same place were captured on different days, they contain significant (non-linear) changes in lighting conditions - strong shadows and bright regions appear and disappear and occlusions and viewpoints change between images.

The definition of novelty depends on *the memory* we provide the system i.e. which images are used as reference images to detect novelties in a query image. But it also, from the design of our system, depends on the manual annotations we provide for training. However, an accurate annotation is very expensive to obtain manually and is subject to choices made by the annotator. Annotators were not given strict rules but were simply asked to annotate what they thought was not a part of the environment in disjoint subsets of images. They did not consider what actually changes in the other images in our data set. Therefore, we do not have entirely consistent annotations that strictly follow objective rules: in some annotations, we have strong shadows labeled as novelty while in some cases, some parts of the environment that appear multiple times at the same physical place are labeled as novelty. While it is impossible for any algorithm to agree completely with the ground truth, we expect a reasonable algorithm to statistically agree with the majority of the annotations.

Figure 5. Three images of the same place from our data set and the ground truth labeling of the last image. Note the variation in lighting conditions, strong shadows, occlusions and changes in viewpoints.

In the following evaluations, we divide our images into training and testing sets, use the training set to fit our models and to cross validate its parameters and we report the results on the testing set.

3.2. Estimating the probability of novelty

Figure 6 (left) shows the results of using different combination of features in computing $\tilde{P}$. The beginning capital letters in the figure denote which features are used e.g. `I` denotes the I^{err} measure and `ISH` refers to the combination of I^{err}, S^{err} and H^{err}. The subsequent letter refers to a single scale "`s`" or a multi scale "`m`" version of the mentioned features. The final letters after the "`-`" sign ("`a`", "`h`" and "`m`") refer to the aggregation function applied to different pairwise error measures (the algebraic mean, harmonic mean and the minimum respectively).

It can be observed that by taking the minimum of the most basic measure, I^{err}, over 5 different reference images `Is-m`, the Average Precision (AP) of 41.2 can be achieved. By including more aggregating functions, the harmonic and algebraic means, the AP improves to 43.8 `Is-ahm` while by considering the multi scale version of the same measure `Im-m`, the AP improves considerably to 62.9. Using a multi scale version of the same feature I^{err} with multiple aggregation `Im-ahm` function achieves an AP of 66.8. Therefore, we use multiple aggregations and multi-scale versions of the features in the remaining part of the evaluations.

To evaluate the contribution of each feature, we report the performance measure when the feature is removed from the feature pool: in order to evaluate the contribution of I^{err} measure, we report the performance of `SHm-ahm` and compare it to a logistic regression based on all three measures `ISHm-ahm` with an AP of 70.4. We expect features with more information to have more contribution to the performance of `ISHm-ahm`. Therefore, the results suggest that the H^{err} measure contains more information than the other two: AP of 68.5 for `ISm-ahm` compared to 69.9 for `SHm-ahm` and 69.1 for `IHm-ahm`. For the rest of the evaluations, we use the entire feature pool (78 dimensions) unless stated otherwise.

Figure 6 (middle) shows the results of using a different number of reference frames to compute $\tilde{P}$. Using only one reference frame (one pairwise comparison) results in an AP of 43.9 while increasing the number of reference frames increases performance. Due to computational issues we do not consider using more than 5 reference frames (AP of 70.4) but the figure suggests that increasing the "memory" of the system i.e. by increasing the number of reference images compared to a query image, the performance of the system increases.

Parameter Settings						
Feature	h	λ	m_0	$P(l_x)$	log	**Acc**
CSP	0.66	10	0.4	P_{SF}	1	**91.86**
CSP	0.5	1	0.4	P_{SF}	0	**91.76**
CSP	0.5	10	0.3	P_{SF}	1	**91.75**
CSP	0.66	10	0.4	P_{SF}	0	**91.72**
CSP	0.5	1	0.3	P_{SF}	0	**91.71**
CSP	0.75	10	0.4	P_{SF}	1	**91.69**
CSP	0.5	0.5	0.3	P_{SF}	0	**91.67**
CSP	0.5	10	0.2	P_{SF}	1	**91.56**
CSP	1	10	0.4	P_{SF}	1	**91.50**
CSP	0.5	5	0.1	P_{SF}	1	**91.43**

Table 3. Evaluation of different parameter settings for the segmentation process. The pixel-wise accuracy of the 10 best performing settings are presented. Compare with the accuracy of thresholding $\tilde{P}$ (the initialization for the segmentations) at 0.5 : 90.64.

3.3. The Segmentation Method

Table 3 shows quantitative evaluation of the segmentation step using the 10 best performing parameter settings from the 50 we tried where *best* is defined relative to the pixel-wise accuracy measure. From the results the following observations can be made. All the three feature measurement types used in the KDE likelihood computations have a positive role in improving the segmentation. One should avoid using information from uncertain regions ($m_0 > 0$) when initializing the likelihood model and that P_{SF} performs better than the other two priors imposed on the pixel label.

It should be emphasized here that our annotations do not match the data exactly. Large brush strokes were used

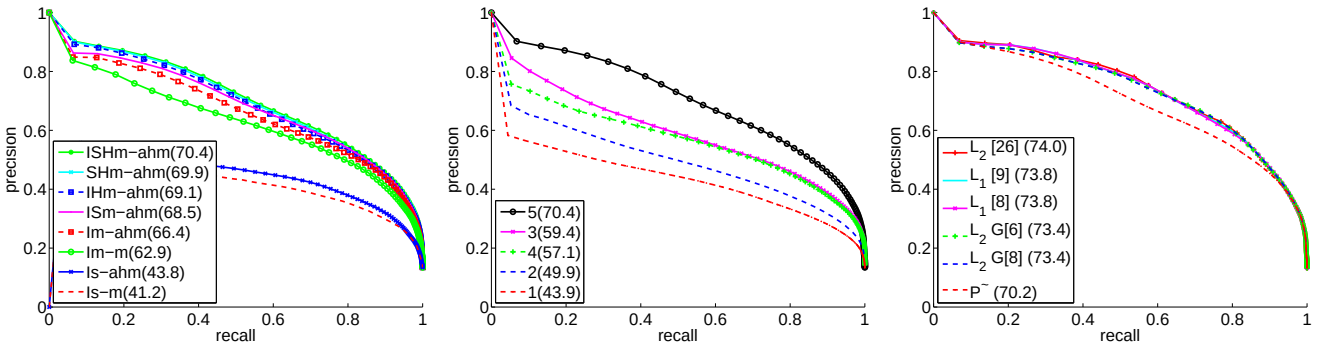


Figure 6. Quantitative evaluation of the individual pixel classifier. The effect of using different features and different combinations of features (left), the effect of using a different number of reference frames (right) and the final probability measure $\hat{P}$ combining multiple segmentations and $\tilde{P}$ using 5 reference frames.

to manually label the novelties, therefore our annotations over-estimate the extent of the true novelties. Our annotations therefore agree more with smoother and slightly over extended estimations. Therefore, by fitting boundaries of the segmentation to their exact locations, we have probably decreased the accuracy measure compared to a slightly over extended estimation! This probably accounts for the small quantitative improvements in accuracy and AP measures over the estimations achieved by thresholding $\tilde{P}$.

3.4. Combining Multiple Segmentations

Figure 6 (right) shows a quantitative evaluation of the combination of multiple segmentations approach. The figure presents the results for combinations of $\tilde{P}$ with different segmentations using different priors. The L_2 [26] refers to an L_2 regularized logistic regression fitted to 25 of the best performing parameters, L_2 G[x] to the greedy selection of x out of the best 8 and L_1 [x] to an automatic feature selection of x features using L_1 regularization.

It can be observed that the suggested approach efficiently combines different segmentations (compare $\tilde{P}$ with the rest) and that L_1 regularization based feature selection outperforms the greedy approach for the same level of sparsity in the solution (compare L_1 [8] and L_1 [9] with L_2 G[8] and L_2 G[6]). In summary, we can achieve more than 3.5 percent increments on the AP measure by combining multiple segmentations. However, the argument we made earlier about the over-extension of the ground truth labelling still holds here and therefore, we believe the true gain to be greater than is reflected in these numbers.

4. Discussions and Conclusions

Figure 7 shows some qualitative results of our method. While most of the results are quite compelling and convincing, some depict the limitations of the method. In particular, as is the case with any correspondence method, large homogeneous regions cause problems as they are ambiguous to register. While our method can overcome incorrect established correspondences to a reasonable extent, the algorithm will have difficulty in detecting novel textureless segments occluding textureless background regions if the wrong correspondences are established consistently across different reference images. This probably accounts for most of the missed novelties.

Although our method is robust to illumination and moderate view point changes, it cannot cope with large changes in the appearance such as strong textures induced by strong shadows. However, as more reference images are added to the system e.g. with the passage of time in wearable systems, scenes will be represented under various illumination conditions and view points and this issue will become less important. Figure 6 (middle) provides evidence for this argument.

In conclusion, we presented a system which uses multiple images of the same environment captured at different times, viewpoints and lighting conditions to implicitly learn a background model and segment out the novel objects. As for future work, it would be interesting to also consider temporal information and to consider an extra constraint of consistency across different view points. Using such an approach, we would be able to explicitly learn the underlying 3D model and its projection in each view point, which would allow us to make a dense 3D model of the environment and to automatically remove the novelties, and fill them in with the learnt background model.

Acknowledgements: We want to thank Jan-Olof Eklundh for his constructive comments about the segmentation process. This work has been funded by the Swedish Foundation for Strategic Research (SSF); within the project VINST.

References

[1] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *ECCV*, 2008. 2, 4

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in

Figure 7. Qualitative results of our algorithm. The first three rows show one result per different place that we have collected data (12 places in total). The last row shows some failure cases where most likely either parts of objects are missed or strong changes in appearance (e.g. strong shadows) are detected as novelties.

vision. *PAMI*, 2004. 3

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001. 3

[4] A. Bugeaue and P. Perez. Detection and segmentation of moving objects in highly dynamic scenes. In *CVPR*, 2007. 2

[5] B. Catanzaro, B. Su, N. Sundaram, Y. Lee, M. Murphy, and K. Keutzer. Efficient, high-quality image contour detection. In *ICCV*, 2009. 3

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, pages 1871–1874, 2008. 3

[7] A. Joulin, F. R. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, pages 1943–1950, 2010. 2

[8] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *PAMI*, 2004. 3

[9] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, pages 28–42, 2008. 2

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, pages 91–110, 2004. 2

[11] M. Maire, P. Arbelaez, C. C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008. 3

[12] A. S. Ogale, C. Fermuller, and Y. Aloimonos. Motion segmentation using occlusions. *PAMI*, pages 988–992, 2005. 2

[13] O. Pele and M. Werman. The quadratic-chi histogram distance family. In *ECCV*, 2010. 3

[14] T. Pollard and J. L. Mundy. Change detection in a 3-d world. In *CVPR*, 2007. 2

[15] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 3

[16] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011. 2

[17] A. Taneja, L. Ballan, and M. Pollefeys. Image based detection of geometric changes in urban environments. In *ICCV*, 2011. 2