Object Segmentation using Spatial and Spatio-Temporal Cues

Omid Aghazadeh, Jan-Olof Eklundh, and Josephine Sullivan

Computer Vision and Active Perception Laboratory, KTH, Sweden

Abstract. Determining boundaries of objects is a challenging problem in computer vision. Many approaches reason about object boundaries based on appearance based cues. In this paper we aim to study how informative a second image of a static scene is. The extra image will contain information about the depth discontinuities in the scene when camera undergoes mostly a translational motion. We propose to make use of the parallax in two closely related problems. We first detect object boundaries using motion based cues computed from the optical flow between two images, and appearance information based on the global Pb detector. We then aggregate these boundaries with color and motion cues in an energy minimization framework, building on the well-known Graph Cuts method. We evaluate our proposed methods both for object boundary detection and object segmentation qualitatively and quantitatively and show that the use of motion parallax can resolve difficult cases in static scenes that elude other approaches.

1 Introduction

In this paper we consider static scenes in which there is a prominent object in the field of view. We wish to accurately find the outline of this object and segment it out from the background. The outline of the object provides information about the shape of the object and the geometry of the scene. It can also be integrated into the object segmentation process as a strong cue about the object's extent [1, 2].

If we have no prior information about the background scene and none about the foreground object then finding the occluding contour reliably and automatically is almost impossible using standard segmentation methods. However, methods do exist for highlighting object boundary edges in an image over texture and clutter edges [3, 4]. Unfortunately, such methods cannot be expected to nor can they discriminate robustly the boundary edges of our foreground object. However, if we have access to another image of the object taken from a different viewpoint, then it becomes easier to infer the object's boundary edges from the parallax information provided by this extra image. At the parallax is observable only through a translative camera motion [5], throughout this paper we assume that the viewpoint change between the two images is mostly translative.

The first half of the paper introduces our automatic classification algorithm for detecting a foreground object's occluding contour using both appearance information, via the global Pb detector, and motion cues computed from the optic

Fig. 1. An example result of our method: using two images and a rough estimate of where the object is located, our method segments the object out from the background. (top left) the first frame, (top right) the second frame and the rough estimate of the object's location, (bottom left) the inferred object boundaries and (bottom right) the resulting segmentation(best viewed in color).

flow field obtained from our two images. The main result is that the combination of appearance and weak motion cues can provide much better results than using either the motion cues or appearance cues individually. It is also interesting to note that to exploit the parallax cues an optic flow field is sufficient. Neither explicit camera calibration nor exact disparity maps are required for this task.

While satisfying results are obtained by our object boundary detector, it does not completely delineate the object's boundary and does not segment the image into foreground and background pixels. The second half of the paper shows how to obtain this information via segmentation(see Figure 1), using an energy minimization framework solved efficiently by the graph-cuts algorithm [6]. In the cost function the output of the boundary detector is incorporated within the pairwise terms and the color and motion cues within the unary terms. The method proposed is similar to the GrabCut algorithm [7] with initial models for the color appearance and motion of both the foreground and background pixels being updated after each iteration of the graph-cut minimization. Like GrabCut, initial estimates of these models are needed. These could, be obtained automatically from the estimated object boundaries using an approach similar to [2, 4], however, for this paper we obtain them with manual intervention - a user marks a foreground bounding box. Because unlike [7] and similar approaches, which hard code the initializing background pixels, inference is performed over all pixels according to the probabilistic formulation, the segmentation achieved is quite robust to the initialization process. Also, because both appearance and motion cues are considered simultaneously in the probabilistic formulation, the combination improves results of either of features used alone.

In this paper we only consider the frame-to-frame motion and do not assume object motion. Therefore, while ample work on motion segmentation exists e.g. [8, 9], they are not directly applicable to this problem. Longer motion sequences and/or independently moving objects, as dealt with in [10–12], allow motion segmentation to be performed more reliably. While ideas from these papers have influenced us, they address different problems than we do. However, we need to emphasize that there is no assumption made here which prevents the method to be directly applicable to the case of independently moving objects.

The contributions of this paper are to propose: 1- an accurate object boundary detector based on appearance and two frame based motion cues, 2- a quantitative measure for evaluating sparse/slightly displaced detections, 3- an iterative object segmentation method based on color and two frame based motion cues which integrates object boundary detection into the object segmentation process and to provide 4- a data-set of 14 image pairs with ground truth information for the purpose of object segmentation in static scenes in case of moving cameras.

2 Object Boundary Detection

Many approaches estimate the object boundaries from one image such as [3, 4] while other approaches estimate the occlusion boundaries during (or after) the motion estimation process from two or more frames such as [13]. Our boundary detector is closely related to that of Stein et al [13] as we also use motion(more precisely, parallax) and appearance cues. It differs from [13] as: 1- we only use two frames, 2- we use global motion estimation instead of spatio-temporal filters, 3- we make use of sophisticated appearance based cues to build a more accurate and robust object boundary detector.

We formulate the object boundary detection as a classification problem. We aim to learn a function which, given a set of appearance and motion cues, classifies each pixel to belonging to object's boundaries or to some other regions such as interior of the object or background. The components of the detector are now described.

2.1 Features

First, we introduce the features that we used in the classification framework.

Appearance based cue: We build upon the global Pb detector([3]) which is a powerful appearance-only based object boundary detector. Using a model statistically learnt from a large data set of annotated images, it combines color and texture cues to provide spatially consistent estimates of object boundaries.

Motion based cues: Our approach is applicable for arbitrary number of available frames but, we limit our discussion to a two-frame case. We consider the forward and backward flow fields(u_b and u_f):

$$\begin{aligned} I(x,0) &= I(x+u_f(x),1)+\epsilon_f(x) \\ I(x,1) &= I(x+u_b(x),0)+\epsilon_b(x) \end{aligned} \tag{1}$$

where $I(.,0)$ and $I(.,1)$ represent the first and second images of an image pair, u_f and u_b are the forward and backward flow fields between the two frames and ϵ_f and ϵ_b are the forward and backward warping errors which optical flow estimation methods usually aim to minimize.

Correspondences do not exist for pixels in regions which are occluded in one image but visible in the other one. The motion of such pixels is usually estimated using a smoothness assumption on the flow field. Violations of such assumptions can be assumed to be caused by depth discontinuities at occlusion boundaries. This makes the problem of occlusion boundary detection particularly interesting as a strong cue about depth discontinuity in images. For more information regarding the occlusion boundaries, the reader is strongly encouraged to refer to [9, 13].

While it might appear that the warping error can be used as a cue for object boundary detection, we have found it to be too noisy and unreliable for this purpose and instead, inspired by the vector field regularization techniques, we make use of 3 cues to highlight the *motion discontinuities*. Let $\mathcal{U} = \{u_f, u_b\}$ contain the forward and backward flow fields, the cues computed from $\mathcal{U}$ are:

$$\begin{aligned}
\tilde{P}_{\texttt{Div}}(\mathcal{U}) &= |\texttt{div}\ \ (u_f^{\mathbf{w}(-u_f)})| + |\texttt{div}\ \ (u_b)| \\
\tilde{P}_{\texttt{Inc}}(\mathcal{U}) &= |\texttt{div}\ \ (u_f^{\mathbf{w}(-u_f)} + u_b)| \\
\tilde{P}_{\texttt{GM}}(\mathcal{U}) &= ||(\nabla||u_f||)||^{\mathbf{w}(-u_f)} + ||(\nabla||u_b||)||
\end{aligned} \tag{2}$$

where $X^{\mathbf{w}(u)}$ denotes feature X warped according to the displacement field u. We use the method of [14] to compute the optical flow between two frames because of their image based anisotropic regularization of the flow field which adapts the discontinuities in the flow field to that of the image edges and also because of their fast GPU based implementation.

While image based anisotropic regularizers can result in better localization of the flow field discontinuities near strong edges in images, they are more prone to spurious discontinuities(see Figure 4). For this reason, we make use of anisotropic image based regularizers in addition to flow based isotropic regularizers to compute two sets of flow fields and average the cues computed from the flow fields to be less sensitive to these spurious discontinuities. For more information about different regularizers, the reader is referred to [15].

Let $\mathcal{U}_I$ and $\mathcal{U}_F$ denote the flow fields computed using image based and flow based regularizers. We define the final flow based cues to be the geometric mean between the mentioned 3 cues computed over flow fields using image based and flow based regularizer:

$$P_{\texttt{Cue}}(\mathcal{U}_I, \mathcal{U}_F) = \sqrt{\tilde{P}_{\texttt{Cue}}(\mathcal{U}_I)\tilde{P}_{\texttt{Cue}}(\mathcal{U}_F)} \tag{3}$$

In summary, we use an appearance based cue which was chosen to be the gPb detector and three motion based cues. See Figure 4 for a visualization of these cues.

2.2 Supervised Learning

We train a linear SVM classifier to map a transformation of the aforementioned cues to binary values representing object boundaries. We first augment the 4 dimensional feature vector with the geometric means of $\binom{4}{2} = 6$ pairwise selections of the features, introducing non-linearity to the linear-SVM. In order to locally integrate neighborhood information in our feature vector, we perform a Gaussian smoothing of the 10 dimensional feature vector with a Gaussian filter with standard deviation of 2 pixels in both directions and augment the original feature vector with its smoothed version to get a final 20 dimensional feature vector for each pixel.

We then use LibSVM [16] to train an ℓ_2 regularized linear SVM using the ground truth information as the target labels after proper division of the the data set randomly to equally sized training and testing pairs. We repeat this process 5 times and find the classifier with optimal parameters using the average cross validation performance.

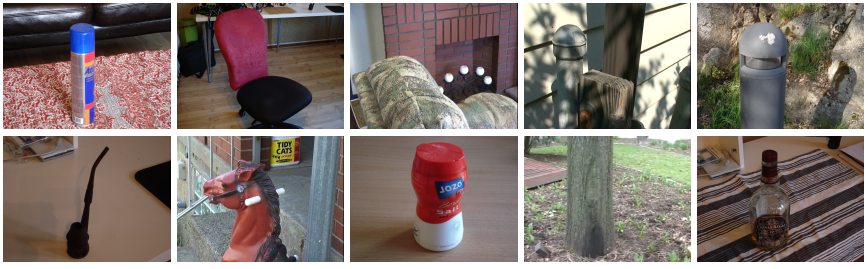

Fig. 2. Some samples of the second frame of image pairs in our data set: from top left to bottom right: Spray1, Chair1, couch_corner, fencepost, Light1, Pipe1, rocking_horse, Salt1, tree and Whisky1. Pairs in lower cases are from data set published by Stein et al [13].

2.3 Quantitative Measure

As previously stated, the flow computed for occluded regions is usually inferred from a smoothness prior (the regularization process). Therefore, the discontinuities in the flow fields computed from two images do not exactly match the object boundaries and instead, are usually displaced according to the motion of the occluding objects. In the case of small motions between immediate frames, this translates to a few pixels. Thus, the information regarding the spatial extent of objects inferred from the motion discontinuities will not be perfectly localized to object boundaries without some higher level reasoning process. Different approaches to address this issue can be considered. For instance, [17] uses an oriented disk which ignores the information in a few pixels around the central axis of the disk. However, we aim to capture *slightly displaced object boundaries* and defer this high level reasoning to the object segmentation process.

Therefore, a measure needs to be defined for slightly displaced detections which scores slightly displaced labellings better than completely incorrect ones. Considering a detection $\rho : \mathcal{X} \rightarrow [0,1]$ and the ground truth $\rho^* : \mathcal{X} \rightarrow \{0,1\}$ where $\mathcal{X}$ represents the set of all pixels in the image: the mean absolute error between ρ and ρ^* is defined as:

$$D_H = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} |\rho_x - \rho^*_x| \tag{4}$$

where ρ_x denotes the value of ρ at pixel x. We introduce a soft approximation to the mean absolute error which allows displacements:

$$\begin{aligned} D(\rho, \rho^*) &= \tfrac{1}{2|\mathcal{X}|} \sum_{x \in \mathcal{X}} (P(x) + R(x)) \\ P(x) &= \min_{y \in \mathcal{X}} |\rho_x - \rho^*_y \exp\{-\tfrac{||y-x||^2}{2\sigma^2}\}| \\ R(x) &= \min_{y \in \mathcal{X}} |\rho^*_x - \rho_y \exp\{-\tfrac{||y-x||^2}{2\sigma^2}\}| \\ Q(D) &= e^{-\gamma D} \end{aligned} \tag{5}$$

The proposed measure resembles a balanced precision-recall measure as $P(.)$ and $R(.)$ resemble soft measures of precision and recall. The displacements are penalized softly by the parameter σ. We chose $\sigma = 2$ so that a perfect detection displaced by 1, 2, 3 and 4 pixel gets penalized approximately by a factor of 1.1, 1.6, 3.1 and 7.4 respectively. Furthermore, the measure is able to cope with partial detections i.e. if half of the equidistant pixels on a ground truth line are detected, the detection will be penalized by a factor of 1.1 instead of a factor of 2. These characteristics make D useful for measuring the quality of sparse or slightly displaced detections of the object boundaries.

To re-scale D to an illustrative interval, we use $Q \in [0,1]$ which resembles the quality of a detection. We use $\gamma = 200$ to report the results in the following. Figure 3 depicts two cases where the detections are clearly informative about the object's extent. While the displacements and/or partial detections are penalized by a hard measure such as D_H (4), the D measure copes with the imperfect detections.

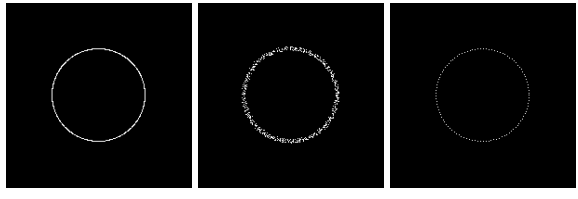

Fig. 3. The proposed quantitative measure on a synthetic test case: (left) ground truth ρ^* (middle) noisy detection with on average two pixel displacement and (right) partial perfect detections. A hard measure such as D_H evaluates the detections poorly(0.25 and 0.53) while the soft measure D evaluates them reasonably(0.82 and 0.92).

2.4 Data Set and Results

We use a data set consisting of 22 pairs of images 8 of which are from the data set published earlier by Stein et al[13]. The reason that we do not consider the rest of the 30 sequences is that we are limiting this study to static scenes in which there is one (or more) objects clearly standing out in a way that a human observer is able to pick *the one object* without much hesitation.

Figure 2 shows the second frame of a few image pairs from our data set and Figure 4 depicts the results of our classifier on a few sample pairs. It can be observed that utilizing the motion information helps the classifier to achieve precise and accurate detection of object boundaries.

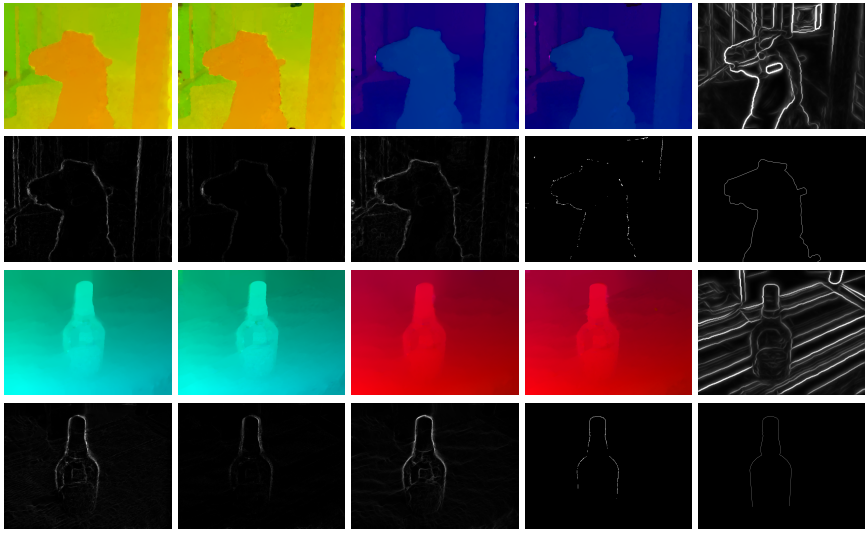

Fig. 4. Qualitative evaluation of the cues and the object boundary detector on two image pairs. First row: forward and backward flow fields using Flow based and Image based regularizers ($\{u_f\} \in \mathcal{U}_F$, $\{u_f\} \in \mathcal{U}_I$, $\{u_b\} \in \mathcal{U}_F$, $\{u_b\} \in \mathcal{U}_I$) and gPb thick. Second row: $P_{\texttt{Inc}}$, $P_{\texttt{Div}}$, P_{GM} motion based cues, the result of the detector and the ground truth. (Best viewed electronically.)

Cue	gPb thick	gPb thin	$P_{\texttt{Div}}$	$P_{\texttt{Inc}}$	$P_{\texttt{GM}}$	Detector
100* Q	17.48	49.02	55.46	51.21	58.22	**78.12**

Table 1. The performance of the appearance and motion based cues based on (5).

Table 1 shows the quantitative evaluation of the individual appearance and motion based cues in addition to the performance of the classifier on the entire data set. Both the quantitative and qualitative results suggest that the pro-

posed object boundary detector is able to infer object boundaries using strong appearance and motion based cues robustly and precisely.

3 Object Segmentation

In this section, we propose an iterative Graph Cuts[6] based object segmentation method which is able to integrate the object boundary detector proposed in the previous section. Our method is similar to Grab Cut [7] as it is iterative and it is based on Graph Cut for global energy minimization. It differs from similar methods in many ways: 1- we utilize the parallax information embedded in motion to disambiguate between regions with similar appearances. 2- unlike many tracking methods we do not assume independently moving objects and nor do we require more than two frames. 3- our method is much less sensitive and more robust to imperfect initializations, 4- we utilize object boundary detection in the energy minimization framework. 5- unlike [11] and similar methods, we do not start from an over-segmentation as the precision of such approaches will always be limited by the accuracy of the initial over segmentation.

3.1 An Energy Minimization Approach

As stated in [6], many modern object segmentation approaches are based on minimizing some kind of energy. One type of energy that can be globally minimized efficiently and exactly is:

$$E(l) = \sum_{x \in \mathcal{X}} D_x(l_x) + \lambda \sum_{(x,y) \in \mathcal{N}} V_{x,y}(l_x, l_y) \tag{6}$$

where l is a binary labelling assigning each pixel $x \in \mathcal{X}$ a label $l_x \in \{0, 1\}$, $D_x(l_x)$ is the data term(also called the unary term) and determines the cost of assigning the label l_x to pixel x, $\mathcal{N}$ is the set of all neighboring pixels(usually 4 or 8 connected neighborhood) and $V_{x,y}(l_x, l_y)$ is the pairwise smoothness(regularization) term and determines the cost of assigning different labels to pixels x and y.

Many authors such as [6] and [7] model the data term as:

$$\bar{D}_x(l_x) = -\log P(f_x|l_x) \tag{7}$$

where f_x is the feature descriptor of the pixel x and $P(f_x|l_x)$ is the likelihood term and represents the probability of observing feature f_x conditioned on pixel x assuming label l_x. A common choice of the smoothness term is the weighted Ising prior:

$$\bar{V}_{x,y}(l_x, l_y) = \frac{1}{\|x - y\|}[l_x \neq l_y] \exp\{-\frac{||f_x - f_y||^2}{2\sigma^2}\} \tag{8}$$

where $[P]$ is the Iverson bracket.

We use the software available online which is based on [18], [19] and [20] to minimize an energy similar to the energy mentioned above. Although such

an approach is not limited to specific type of features, in this work we use the color (3 dimensional, CIE Lab color space) and motion (2 dimensional) features. Below, we elaborate on the details of the energy function that we minimize.

3.2 Density Estimation

The object segmentation process is directly affected by how the likelihood functions $P(f_x|l_x = 1)$ - the likelihood of pixel x lying on the object - and $P(f_x|l_x = 0)$ - the likelihood of the pixel x belonging to background - are estimated. As the spatial extent of the object is not known beforehand, usually the pixels are assumed to be i.i.d and the PDFs are estimated from an estimate of the spatial extent of the object(s).

While it is possible to assume that the observed features of the object, conditioned on the class label, are independent and while such an assumption leads to simpler density estimation problems, when dealing with noisy features such as optical flow, this assumption leads to spurious undesired effects of the noisy features (in terms of e.g. imperfect localization) on the likelihood function. This, in turn leads to ambiguous segmentation boundaries. For this reason, we avoid the independency assumption of the color and motion features and directly model the class conditional joint PDF.

We utilize a non-parametric method to estimate the class conditional PDFs - sacrificing computational efficiency[1] for accuracy. Let $l^{(t)}$ represent the current estimate of pixel labellings and define $\mathcal{X}_k^{(t)}$ to be the sets of pixels with label k according to the labelling $l^{(t)}$. Utilizing Kernel Density Estimation with a homogeneous Gaussian kernel, the likelihood function then reads:

$$P(f_x|l_x) = \frac{1}{|\mathcal{X}_{l_x}^{(t)}|h^d} \sum_{y \in \mathcal{X}_{l_x}^{(t)}} K\left(\frac{f_x - f_y}{h}\right) \tag{9}$$

where d is the dimensionality of the f (in the case of color and motion: 5), h is the bandwidth(window width) and $K(\mu)$ is the multivariate Gaussian density function with identity covariance matrix evaluated at μ. We whiten the feature space prior to the segmentation process in order to remove the correlation between features and to normalize the standard deviation of each dimension.

Instead of working with the negative log of class conditional likelihoods(7), similar to [12], we propose to use the posterior probability of the label given the observation. Assuming equal priors for the background and foreground, we define the normalized data term as

$$D_x(l_x) = P(\neg l_x|f_x) = \frac{P(f_x|\neg l_x)}{P(f_x|l_x) + P(f_x|\neg l_x)} \tag{10}$$

[1] The computational cost of the KDE is quadratic in the number of pixels which is quite expensive for single core computations. However, using our GPU based implementation and on an NVIDIA GTX 470, we are able to perform the density estimation with a sub sampling of once every two pixels in each direction, in less than a second within each iteration.

The reason for equal priors over class labels a is two fold: 1- updating the prior over class labels after each iteration would introduce additional local minima to the energy function and 2-as we aim to build a method which is not over-sensitive to the initializations (see Figure 7 for some examples of the initialization), we do not use the initialization to estimate class priors. The data term (10) is therefore less demanding on the initializations and is more robust to unavoidable fluctuations of the class conditional likelihoods that stem from the use of noisy features such as optical flow.

3.3 Integrating Detections

The smoothness term suggested in (8), relaxes the smoothness assumptions on neighboring pixels with dissimilar descriptors. While it is a reasonable assumption to expect strong discontinuities in the appearance descriptors on the object boundaries, one needs to consider the cases that the interior region of the object contains several strong edges. Relaxing the smoothness constraints on those areas might lead to unwanted discontinuities in the segmentation process. Figure 5 depicts such a case.

Fig. 5. The effect of using different smoothness terms. The Ising prior (left), weighting the Ising prior using color dissimilarity (middle left), color+motion dissimilarity (middle right) and using our object boundary detector through (11) on (right). Using the motion dissimilarity in addition to the color dissimilarity makes the smoothness term less sensitive to the edges that are not on object boundaries. Note the robustness of the smoothness term based on the object boundary detector(11). (Best viewed in color)

Instead of relaxing the smoothness constraint based on image edges, we propose to make use of a more reliable cue about object boundaries:

$$V_{x,y}(l_x, l_y) = \frac{1}{\|x - y\|}[l_x \neq l_y][\rho_x \neq \rho_y] \tag{11}$$

where ρ_x represents the classification result of the object boundary detector (described in Section 2) evaluated on pixel x. The proposed smoothness term nullifies the smoothness constraints on the pixels which are believed to be on the object's boundary while enforcing the smoothness constraint uniformly on

all other pixels. Figure 5 depicts the effect of using the object boundary detector in the segmentation process.

3.4 Results

In the following, we present qualitative and quantitative results of the method using different features for the segmentation process on the data set introduced in Section 2.4. We also present the results of some of the interactive segmentation methods to give an indication of how informative the human cognition process is and how much the motion feature helps the segmentation process.

Figure 6 depicts the results of three interactive segmentation methods on four pair of images. The figure shows the manual information provided to three interactive segmentation methods,[21], [7] and [3], and the results achieved by each method. The figure does not by any means aim to make any comparison between the depicted methods and nor to draw any conclusions about the performance of the methods. Instead, we aim to show that while the appearance based cues are strong cues for image and object segmentation, without further reasoning about the geometry of the world, they do not contain enough information for reliable estimates of object extents.

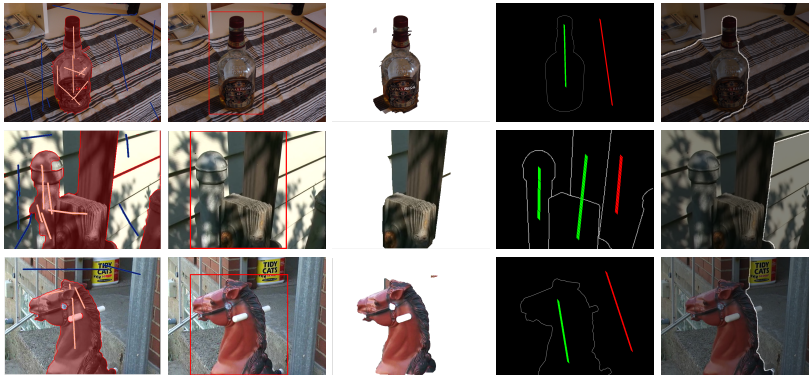

Fig. 6. Qualitative results of three interactive methods on three pair of images. The first and last results are from state of the art interactive segmentation methods. From left to right: TVSeg[21], initial frame for Grab Cut[7], the result of the Grab Cut, initial seeds for the method of [3], the result of [3](best viewed in color).

Figure 7 presents a qualitative comparison of the method in the cases of using color and motion features and color and motion features together in the segmentation. It can be observed that our method can make use of the motion feature to provide more robust estimates of the objects' extents compared to the case of solely using the color feature. Such results suggest that the parallax information embedded in the motion feature can be used to disambiguate the boundaries of objects.

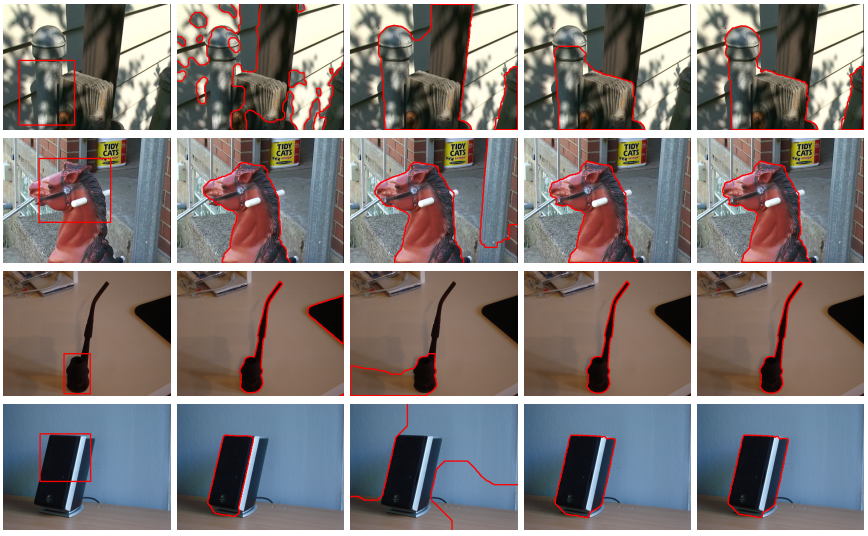

Fig. 7. Motion cue can disambiguate object's extent and lead to less sensitivity to the initializations. (left): the initialization, segmentation using (middle left) color feature with tuned parameters, (middle) motion feature with tuned parameters, (middle right) color and motion features with constant parameters $\lambda = 5$, $h = 0.5$ and (right) color and motion feature with tuned parameters (best viewed in color).

In order to quantitatively measure the performance of our method, we report the average accuracy(pixel-wise) of the segmentations over the 22 pair of frames in our data set for cases that color or the combination of color and motion features are used, with or without the integration of the object boundary detector in the segmentation process. We evaluated the method using the parameters $(\lambda, h) \in \{0.5, 1, 2, 5, 10\} \times \{0.5, 1, 2\}$ and report the results in two conditions: 1- the overall best performing parameters were used for all pairs; we refer to this with constant parameter setting and 2- the best performing parameters were picked for each pair individually which we refer to with tuned parameter setting. Table 2 shows the evaluation results. It can be observed from the results that

Feature	Detector	Fixed Params	Acc (Fixed)	Acc (Tuned)
Color	Not used	$\lambda = 2$, $h = 1$	0.9198	0.9436
Color	Used	$\lambda = 5$, $h = 0.5$	0.9317	0.9551
Color+Motion	Not used	$\lambda = 2$, $h = 1$	0.9387	0.9578
Color+Motion	Used	$\lambda = 5$, $h = 0.5$	**0.9490**	**0.9715**

Table 2. The quantitative evaluation of the segmentations using color and motion features. Acc (Fixed) refers to the result achieved by the fixed parameter setting mentioned in Fixed Params column and Acc (Tuned) refers to the results achieved if the best performing parameters from a fixed set of parameters(see the text) were selected individually for each pair.

integrating the object boundary detector increases the performance of the segmentation process in all cases. By removing the smoothness constraint on areas which were believed to be on object boundaries, stronger regularization($\lambda = 5$ instead of $\lambda = 2$) could be used without causing the boundaries to be over-smoothed or mixed with neighboring objects. The stronger regularization enables the method to cope with changes in appearance/motion caused by e.g. lighting conditions. It can also be observed that the use of motion information is robustly beneficial in both the constant and tuned parameter settings. Moreover, it is evident from the results that further feedback to the method in terms of the choosing the optimal parameters, will lead to more accurate segmentations.

In summary, we have showed that with subtle modifications to the state of the art methods, it is possible to achieve satisfactory results using only the color information(0.920 accuracy). It is possible to go further, and with the use of one extra frame, achieve an accuracy of 0.949 without any extra manual effort. With little manual effort in terms of choosing a good parameter settings, or through the use of a method which does so automatically, it is possible to achieve 0.972 accuracy. We find these results promising as we believe with little improvements in terms of automatic parameter tuning, we can reach close to perfect segmentations for similar data sets using only a pair of images. We need to emphasize here that unlike other interactive methods, our method has the potential to automatically segment out the prominent object without any user interaction, provided that a reasonably good initialization can be acquired via some other method such as [22].

4 Conclusion and Future Work

In this paper, we addressed two problems i.e. the problem of detecting object boundaries and the problem of segmenting out prominent objects from the background, and showed that integrating the object boundary detection into the segmentation problem improves the results.

We have addressed the object boundary detection problem in a supervised learning framework using the robust appearance based global Pb detector and three motion based cues. When using regularized features such as optical flow, small spurious displacements in the regularized features are difficult to avoid. Therefore, we introduced a measure which is preferable for evaluating slightly displaced boundary detections and evaluated the proposed object boundary detector using this measure.

We have addressed the problem of object segmentation in an iterative energy minimization framework. We achieved promising results using color and motion cues in combination with the output from our detector. The reason for success of our method are the modified data and smoothness terms and non parametric multi dimensional joint feature kernel density estimations.

Future work includes a tracking system based on pairs of images to perform robust and accurate object segmentation in video using initialization from one

frame, studies of adaptive parameter selection, automatic initialization based on multiple hypotheses and applying the system to the case of moving objects.

References

1. Ogale, A.S., Aloimonos, Y.: A roadmap to the integration of early visual modules. IJCV (2007)
2. Stein, A., Stepleton, T., Hebert, M.: Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In: CVPR. (2008)
3. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour Detection and Hierarchical Image Segmentation. PAMI (2010)
4. Hoiem, D., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from an image. IJCV (2011)
5. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. (2004)
6. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. IJCV (2006)
7. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. In: SIGGRAPH. (2004)
8. Brox, T., Bruhn, A., Weickert, J.: Variational motion segmentation with level sets. In: ECCV. (2006)
9. Ogale, A.S., Fermuller, C., Aloimonos, Y.: Motion segmentation using occlusions. PAMI (2005)
10. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: ECCV. (2010)
11. Huang, Y., Liu, Q., Metaxas, D.: Video object segmentation by hypergraph cut. In: CVPR. (2009)
12. Bibby, C., Reid, I.: Robust real-time visual tracking using pixel-wise posteriors. In: ECCV. (2008)
13. Stein, A.N., Hebert, M.: Local detection of occlusion boundaries in video. IVC (2009)
14. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic huber-l1 optical flow. In: BMVC. (2009)
15. Weickert, J., Schnörr, C.: A theoretical framework for convex regularizers in pde-based computation of image motion. IJCV (2001)
16. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
17. Sundberg, P., Brox, T., Maire, M., Arbelaez, P., Malik, J.: Occlusion boundary detection and figure/ground assignment from optical flow. In: CVPR. (2011)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI (2001)
19. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. PAMI (2004)
20. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. PAMI (2004)
21. Unger, M., Pock, T., Trobin, W., Cremers, D., Bischof, H.: Tvseg - interactive total variation based image segmentation. In: BMVC. (2008)
22. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M.: Global contrast based salient region detection. In: CVPR. (2011)