KTH Computer Science
and Communication

Object Segmentation using Spatial and Spatio-Temporal features

OMID AGHAZADEH

Master's Thesis at CVAP
Supervisor: Jan-Olof Eklundh
Examiner: Stefan Carlsson

Abstract

This thesis investigates a variational algorithm for object segmentation using multiple spatial and spatio-temporal cues. The aim is to segment images into two regions: foreground region representing an object of interest or a conspicuous object and background region representing everything else. A variational segmentation framework(region based active contours) is utilized with a functional combining statistics of different cues for segmentation such as color, motion and texture.

Moreover, a classification approach to object boundary (occlusion boundary) detection, which combines appearance and several motion cues is presented. A score function was developed for the purpose of assessing the quality of any edge detection algorithm, which is more reliable than the accuracy of the detections. The concept of localized classifiers is presented and discussed, which leads to significant speedups in the training / testing time of classifiers with the cost of additional bias towards the training data. A localized classifier set and a radial basis kernel SVM were trained and using the mentioned score function, they were compared to each other. Comparable results of the mentioned methods verifies the efficiency of the concept of the localized classifier sets in terms of computational costs and accuracy. Furthermore, the use of Geodesic Active Contours to encourage the level sets to converge to some sparsely defined intermediate boundaries (which in this case would be the detected object boundaries) is investigated.

Additionally, a data-set of 2 frame sequences with the ground truth information is prepared and presented and the results of both algorithms are presented on the mentioned data-set. The data-set consists of 25 sequences, 17 of which were taken carefully by a camera undergoing a translational movement and contain indoor / outdoor sequences with different lighting conditions and different degrees of complexity of the scene as well as 8 sequences chosen from an earlier published data-set. Various concepts of the variational segmentation approach such as sensitivity to the parameters, parameter tuning, performance of the algorithm in different cases were investigated and the usefulness of the proposed methods were investigated using qualitative and quantitative results.

Contents

Chapter 1

Introduction

1.1 Background

Image segmentation is defined as the process of partitioning(segmenting) the images into multiple meaningful regions. As this partitioning changes the representation of the images from the set of all pixels to a set of groups of pixels, which are similar in some sense or possess similar characteristics, image segmentation makes the process of analyzing the images easier. According to its definition, the image segmentation problem is ill-posed(there is no unique solution to the image segmentation problem) [6]. Figure 1.1 depicts some possible segmentations of the same scene using different criteria for the segmentations.

A more limited and constrained problem, the problem of segmenting images into meaningful regions corresponding to separate objects namely the object segmentation problem, has been the focus of many researchers in the field of computer vision. Putting a constraint on the regions: they need to correspond to separate 3D objects, makes the problem less ambiguous. The imaging process, which is a mapping from the 3D world to a 2D image, by its nature loses the depth information and thus, the spatial coherence of 3D objects. This and also the fact that the number of objects is not defined in the object segmentation problem makes the problem ill-posed yet again and very hard to solve without any rich a priori knowledge / information e.g. 3D models of specific objects and without integrating the information from multiple views of the same scene. Figure 1.2 depicts some possible object segmentations of

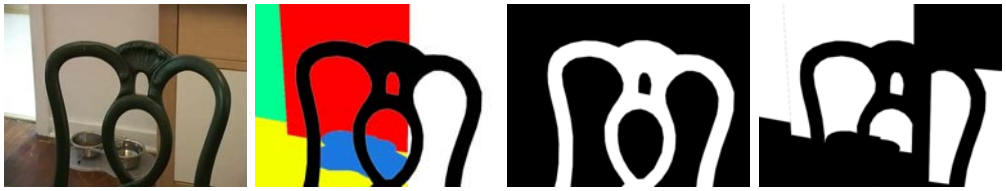


Figure 1.1. Some examples of image segmentation

Figure 1.2. Some examples of object segmentation

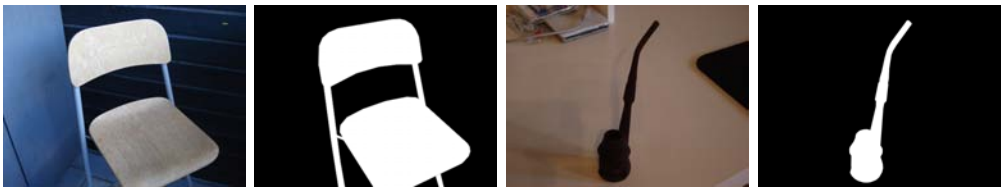


Figure 1.3. Some examples of F/B object segmentation

the same scene with different number of objects.

The problem of object segmentation is interesting both to the robotics community and the computer vision community. As the 3D spatial extent of the objects of interest can be inferred from their corresponding regions in the images, object segmentation can be used in robotics applications such as object grasping / manipulation and obstacle avoidance. Having accurate segmentations can be useful in solving some problems in computer vision: 1- shape description: describing the outline of the objects more accurately and 2- detection / recognition problems: using object segmentation, it is deduced that "something" is located in a specific area in the image; as this limits the number of hypotheses for the location of objects / categories severely(from all possible bounding boxes to the number of segments), this is expected to reduce the number of false positives of classifiers significantly in such problems. However, in this work, the focus is not on the applications of object segmentation but rather on the problem itself.

An even more constrained problem i.e. the problem of segmenting the images into two regions: one corresponding to a conspicuous object(foreground) and the other region(background) corresponding to everything else is particularly interesting, because it constraints the number of segments in the object segmentation problem. Although the Foreground / Background object segmentation problem, is much more limited than the object segmentation problem, it has a specific characteristic which makes it particularly interesting: the 2 region constraint makes the problem well posed for the images(or sequences of images) in which there is one conspicuous object in the image. Figure 1.3 depicts two examples of such cases and the desired segmentation in each case. It can be observed that the F/B object segmentation in

such problems aims to produce segmentations that agree with our perception(and understanding) of the scenes. This is in contrast to the image segmentation problem where the outcome of the segmentation process can vary significantly based on the criterion the image segmentation methods use as measures of similarity or meaningfulness. For the same reasons, in this work, the focus is put more on the F/B object segmentation problem rather than the image segmentation or general object segmentation problem.

A very brief review of the relevant segmentation methods is as follows: Some approaches directly infer the depth associated with visible pixels using two or more views of the same scene(e.g. stereo vision) and use the spatial coherency along with other cues to perform the object segmentation e.g. [3][2]. These approaches usually require calibrated cameras without which a reconstruction of scenes(up to a similarity transformation) is not possible(without a priori information about the scene e.g. perpendicular lines or specific length ratios)[21]. Some approaches estimate the motion associated with each pixel in sequences of two or more images and argue that pixels associated to the same object will have the same motion and thus, for the segmentation, they focus on the motion segmentation [8][57] problem. Motion dissimilarity alone is not a strong cue in cases that objects are not moving independently or have similar motions [58]. An analysis of motion layer segmentation and its limitations in different situations is available in [35].

The exterior visible boundaries of objects in images are referred to as occlusion boundaries[35](as in an image of a scene, closer objects prevent some part of the world to be seen: closer objects occlude some parts of the world). Some approaches estimate the occlusion boundaries(locally or globally) such as [45][44][62][47]. Some approaches aim to infer the object boundaries and their relative depths by assigning the occluded areas to occluded/occluding regions [35][34][63]. Some other approaches try to link the (sparsely) detected edges to infer the object's contour [25].

This work is different to the stereo based approaches as it does not require camera calibration. It differs from motion layer segmentation approaches in the sense that it combines additional cues to perform the segmentation and does not rely solely on motion information. It diverges from the mentioned relative depth inferring approaches as in this work, neither the occluding/occluded regions are explicitly detected nor the relative depth of the objects is being explicitly inferred.

1.2 Scope

In this work, the focus is on the segmentation of one object(a binary: foreground / background segmentation) which is expected to be present approximately in the center of attention(center of the image). Sequences will be prepared consisting of two or more images from a camera undergoing a horizontal translation and the corresponding ground truth information. A method will be trained on the annotated sequences and is expected to work on somewhat more general sequences not violating

the main assumptions of the method(e.g. translative motion and the spatial location of the object of interest).

1.3 Segmentation methods

While there are several approaches to image segmentation, the most popular ones can be recognized as the following:

Clustering methods were one of the earliest image segmentation approaches. While the clustering approach can be extended to cover other features in addition to intensity information, it will not result in spatially coherent corresponding objects [64].

Region Growing takes as input a set of seeds representing initial members of each segment and using a similarity measure assigns the unlabeled neighbor pixels to one of the segments [1]. Later on, automated region growing methods were developed which did not require the initial seeds [46]. In general, region growing methods are sensitive to the initial seeds and are unable to refine the similarity measure in case of severe change in statistics of the regions.

Edge based methods aim to link some spurious(disconnected) edges into a (smooth) curve [4][25]. The quality of the segmentation of the methods in this category relies on the quality of the detected edges and in general, such methods are not able to integrate multiple cues for edge completion.

Graph partitioning methods define the segmentation problem into a graph partitioning problem. It is possible to directly solve the partitioning problem for the optimal partitions e.g. Ncuts: [43] and its multi scale variant [17] using eigenvalue decomposition. However, the reconstruction of the affinity matrix in those cases limits the direct approach to some pre-defined similarity measures and therefore, the approach cannot utilize region statistics. The alternative is to define an energy functional and iteratively refine the partitioning which minimizes the energy e.g. Graph cuts based formulations: min cut/max flow algorithms [64][5]. In min cut/max flow problems, the images are considered as undirected weighted graphs and a source and a sink(in case of binary segmentation) are defined and the goal is to find the cut which has the minimum cost among all cuts. Graph cuts based approaches are guaranteed to find the global minimum of a wide range of energy functionals [23] using the α-expansion scheme. Such approaches aim to minimize the energy functional

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q) \tag{1.1}$$

in which p represents a pixel, $\mathcal{P}$ represents the set of all pixels, $D_p(f_p)$ represents the data-based cost of labeling p as f_p, $\mathcal{N} \subset \mathcal{P} \times \mathcal{P}$ is a neighborhood system and

$V_{p,q}(f_p, f_q)$ defines the smoothness constraint (the cost of assigning labels f_p, f_q to p, q).
In general, graph cuts based approaches are very similar to their variational counterpart and the difference is the discrete nature of graph cut based approaches vs the continuous optimization in the variational method.

Variational methods are similar to the Graph cuts based approaches aiming to minimize energy functionals. In combination with Level Sets(Φ), these approaches aim to minimize functionals of (in the most general case) the form

$$E(\Phi) = \int_{\Omega} \left(\underbrace{-H(\Phi)\log p_1 - (1 - H(\Phi))\log p_2}_{D(\Phi)} + \underbrace{\nu|\nabla H(\Phi)|}_{V(\Phi)} \right) \mathtt{dx} \qquad (1.2)$$

which resembles its discrete counterpart exactly. Although variational methods can reach sub-pixel accuracy because of the nature of their continuous formulation, they converge to global minima only in the case of convex functionals. These approaches can easily be extended to multi-class segmentation problems [38][12]. The minimization process can be done using calculus of variations. The Euler-Lagrange equations can be solved numerically by proper discretization leading to linear system of equations or simply by a gradient descent approach [6].

Variational segmentation have proven to be a powerful tool for image segmentation [18][49]. As they are easily extendable to multi-class problems and as solutions to some family of variational approaches can be implemented on GPUs providing very fast solutions [38], a variational segmentation method was decided to be utilized.

1.4 Outline

The rest of this thesis is organized as follows:
In chapter 2, the variational segmentation framework in its multi-cue form is introduced, derived and discussed using level sets and it was shown that minimizing the proposed functional is equivalent to maximizing the a-posteriori in a Bayesian framework under some specific assumptions. Two curvature motions are discussed and the use of GAC(Geodesic Active Contour) as a means of encouraging the evolving boundaries of the regions to converge to some edges in the image is motivated. Also, possible ways to estimate the pdfs of the features in the functional are commented on and the possible features, feature smoothing and feature weighting are described.
In chapter 3, a classification / regression based approach to occlusion boundary detection is proposed, which utilizes appearance and motion cues. Different motion based cues are discussed and possible ways to combine the cues and advantages and drawbacks of each approach are elaborated. Also, a measure for quantitative

assessment of results of any [edge] detection algorithm is proposed, which is shown to be more reliable than the accuracy of the detections.
In chapter 4, the results of the proposed algorithms are presented and discussed as well as the drawbacks, limitations and advantages of the proposed algorithms and the robustness of the variational segmentation approach to the assumptions of the method. Finally, in chapter 5, the thesis is summarized and concluded.

Chapter 2

Variational Segmentation

2.1 Early works

The first variational formulation for the segmentation problem, the Mumford-Shah functional [32]:

$$E(u,\Gamma) = \int_{\Omega}(I-u)^2 \ \mathtt{dx} + \gamma \int_{\Omega-\Gamma} |\nabla u|^2 \ \mathtt{dx} + \nu \int_{\Gamma} \ \mathtt{ds} \tag{2.1}$$

aimed to segment the intensity image into piecewise smooth regions(u) separated by the boundary Γ while keeping the length of the separating boundary minimum. While this functional is general and covers many different formulations, there is no numerical approach to minimize this functional uniquely without further assumptions[6]. A simplified version of this functional namely the Cartoon limit of the Mumford-Shah functional, aims to find the piecewise constant approximation of the image eliminating the middle term:

$$E(u,\Gamma) = \int_{\Omega}(u-I)^2 \ \mathtt{dx} + \nu \int_{\Gamma} \ \mathtt{ds} \tag{2.2}$$

For a two-region problem, the cartoon limit reads:

$$E(\Gamma) = \int_{\Omega_1}(I-\mu_1)^2 \ \mathtt{dx} + \int_{\Omega_2}(I-\mu_2)^2 \ \mathtt{dx} + \nu \int_{\Gamma} \ \mathtt{ds} \tag{2.3}$$

Later on, Chan and Vese [14] introduced the level set formulation of the cartoon limit: using the level set function$\Phi : \Omega \mapsto \mathbb{R} : \Phi(x) \geq 0$ if $x \in \Omega_1$ and $\Phi(x) < 0$ if $x \in \Omega_2$(and therefore, the zero level line of Φ determining Γ), the cartoon limit reads:

$$E(\Phi) = \int_{\Omega} \Big(H(\Phi)(I-\mu_1)^2 + (1-H(\Phi))(I-\mu_2)^2 + \nu|\nabla H(\Phi)| \Big) \ \mathtt{dx} \tag{2.4}$$

where H is the regularized version of the Heaviside(step) function. The mentioned functional leads to the evolution equation:

$$\partial_t \Phi = H'(\Phi) \left((I-\mu_2)^2 - (I-\mu_1)^2 + \nu \mathtt{div}\left(\frac{\Phi}{|\Phi|} \right) \right) \tag{2.5}$$

which after proper discretization leads to an iterative approach for energy minimization. This resembles an expectation-maximization-like approach in which at each iteration, the expected value of μ_1 and μ_2 are updated and then, Φ is updated according to the expected μ_1 and μ_2.
The mentioned functionals solely use the intensity feature for the segmentation problem. However, other features can be included in the same framework very easily as will be pointed out in the next section.

2.2 Multi cue integration

An optimal partitioning of the image plane[1] $\mathcal{P}(\Omega)$ using feature responses F can be computed by maximizing the a posteriori probability:

$$\max_{\mathcal{P}(\Omega)} p(\mathcal{P}(\Omega)|F)$$

Using the Bayes formula:

$$p(\mathcal{P}(\Omega)|F) \propto p(F|\mathcal{P}(\Omega))p(\mathcal{P}(\Omega))$$

in which the $p(F|\mathcal{P}(\Omega))$ term is the likelihood of the observation of feature set F given the partitioning and the $p(\mathcal{P}(\Omega))$ term is a prior on the favored types of partitioning. Usually, the prior is assumed to be:

$$p(\mathcal{P}(\Omega)) \propto \left[e^{-\nu|C|} = e^{-\nu|\nabla H(\Phi)|}\right]$$

which favors the shorter lengths of the partitioning boundary [18].
As the partitions in the image plane do not overlap, we can simplify the likelihood term(in case of two regions) as:

$$p(F|\mathcal{P}(\Omega)) = p(F|\Omega_1, \Omega_2) = p(F|\Omega_1)p(F|\Omega_2)$$

A common assumption to further simplify the two likelihood terms is to assume that F is iid i.e. feature responses are independently and identically distributed among the partitions. This assumption in general is not valid, specially in case of the features which use neighborhood support(e.g. Texture) [18]. However, it is a common assumption and works well in practice. The mentioned assumption leads to:

$$P(F|\mathcal{P}(\Omega)) = \prod_{i=1}^{2} \prod_{x\in\Omega_i} p_i(F(x))$$

in which p_i is the pdf of the random process which was assumed to generate F in Ω_i.
As maximizing the mentioned a posteriori is equivalent to minimizing its negative logarithm, we get:

$$-\log p(\mathcal{P}(\Omega)|F) = \int_{\Omega_1} -\log p_1(F(x)) \ \mathtt{dx} + \int_{\Omega_2} -\log p_2(F(x)) \ \mathtt{dx} + \nu \int_{\Omega} |C| \ \mathtt{ds}$$

[1] In the case of discrete formulation, the image itself.

and equivalently:

$$E(\Phi) = \int_{\Omega} \left(-H(\Phi)\log p_1(F) - (1 - H(\Phi))\log p_2(F) + \nu|\nabla H(\Phi)|\right) \ \texttt{dx} \tag{2.6}$$

Therefore, under the assumption that the feature responses F is i.i.d, minimizing the functional (2.6) is equivalent to maximizing the a posteriori probability of the determined partitioning given the feature responses F. The Chan-Vese functional (2.4) is a special case of the mentioned functional, where the feature vector is the intensity and the likelihood function is assumed to be gaussian with mean μ and the standard deviation 0.5 [6].
Minimizing the functional (2.6) leads to the evolution equation:

$$\partial_t \Phi = H'(\Phi)\left(\log\frac{p_1(F)}{p_2(F)} + \nu\texttt{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)\right) \tag{2.7}$$

2.3 Curvature Motion

The first and the second term in third term in the energy functional (2.6) (and their corresponding term in the evolution equation (2.7): the first term) evolve the level set with respect to the statistics of the evolving regions while the last term enforces the minimum boundary length prior on the evolving level set. There are in general two possible choices for the prior term in the functional:

Mean Curvature Motion: Can be seen as a balloon-like force in the direction normal to the evolving contour represented by the level set trying to make the region encircled by the contour smaller and its boundary smoother. It can be represented by the functional:[2]

$$E_{MCM}(\Phi) = \int_{\Omega} |\nabla H(\Phi)| \ \texttt{dx} \tag{2.8}$$

leading to the evolution equation:

$$\partial_t \Phi = H'(\Phi) \ \texttt{div} \ \left(\frac{\nabla\Phi}{|\nabla\Phi|}\right) \tag{2.9}$$

Mean curvature motion independent of its initialization shrinks to a circular point as $t \to \infty$ [6].

Geodesic Active Contour: Alternatively, it is possible to slow down(or stop or even reverse the direction of) the curvature motion near strong image edges(or any other specific regions in the image) using a model like the Geodesic Active Contours(GAC):

$$E_{GAC}(\Phi) = \int_{\Omega} g(I_B)|\nabla H(\Phi)| \ \texttt{dx} \tag{2.10}$$

[2]The mean curvature motion has $|\nabla\Phi|$ instead of $H'(\Phi)$ as the scaling factor. However, the nature of both equations is the same and the direction of the evolving force is the same in both.

where I_B is the image containing the desired edges and $g(.)$ is a decreasing function. The functional leads to the evolution equation:

$$\partial_t \Phi = H'(\Phi) \ \texttt{div} \ \left(g(I_B) \frac{\nabla \Phi}{|\nabla \Phi|} \right) \tag{2.11}$$

A similar approach called Geodesic Active Regions is introduced in [36]. As it is pointed out in [6], (2.11) can be written as:

$$\partial_t \Phi = g(I_B)|\nabla \Phi| \ \texttt{div} \ \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) + \nabla g(I_B)^T \nabla \Phi$$

which consists of two different forces: one similar to the mean curvature motion scaled with $g(I_B)$ and the other one attracting the level set in the direction of smaller g. GAC is sensitive to the initialization and it stops the evolution of the curve when the first local minimum(strong image edge) is reached. Therefore, the use of GAC is prohibited in general cases where reasonable initializations are not available[6]. However, if I_B is defined such that it solely contains the regions that the level set is encouraged to converge to, GAC can be used as a prior term without any harm to the final results. This way, an object boundary detector with few false positives can be integrated in the variational segmentation framework. A method for computation of reasonable I_Bs is introduced in chapter 3.

2.4 Estimating the likelihoods

The p_i in the mentioned formulation in (2.6) is a joint probability distribution function of $F : \Omega \mapsto \mathbb{R}^{\left[\sum_{k=1}^{N_F} d_k = D\right]}$ containing N_F features(treated as random variables) and has a dimension equal to the sum of dimension of each feature in F ($D = \sum_{k=1}^{N_F} d_k$).

Given feature vector $f = F(x)$, the joint pdf of f given the ith class: $p(f|c_i) = p_i(F(x))$ can be estimated using different approaches such as the ones mentioned below. In the following, N is the number of D dimensional feature vectors and f_j is the jth feature vector(in a discrete framework).

Histogram based: It is possible to represent $p_i(f)$ using D dimensional histograms. Following this approach, the feature space can be quantized into $(N_Q)^D$ bins and the bins corresponding to features of pixels belonging to each class are increased by 1(or by the membership weight of the pixel) in the histogram corresponding to each class. Afterwards, the histogram is normalized such that it has norm of 1 and then, the probability of f belonging to each class is the value of corresponding bins in each histogram. Using this approach, the number of bins in the histogram grows exponentially with D and therefore even for reasonable numbers of N_Q and D, this approach runs out of samples to populate the histogram. To verify this, one can consider a feature vector with D as low as 9 and each dimension of the feature vector normalized between 0 and 1 gets quantized to 10 regions, the

histogram will have 10^9 bins which will be useless in practice. Histogram based approaches can be thought of as approximations to kernel density estimates with the quantization having a non-linear smoothing effect [6].

A more efficient approach is to use clustering(e.g. k-means) to find k cluster centers of dimension D leading to a 1-dimensional histogram per class. Afterwards, the feature vector of each pixel is assigned to a cluster center using the minimum Euclidean distance criterion and the corresponding bin in the histogram of the corresponding class is increased by 1(or again, the membership function of the pixel to the class). This approach can be formalized as:

$$\begin{aligned} H_i(k) &= \sum_{j \in c_i} \left[k = \arg\min_{t \in C_i} ||t - f_j|| \right] \\ p_i(f) &= \frac{H_i(\arg\min_{t \in C_i} ||t - f||)}{\sum_{t \in C_i} H_i(t)} \end{aligned} \tag{2.12}$$

where C_i is the set of cluster centers for the ith class and the operator $[x_1 = x_2] = \delta_{x_1,x_2}$ is the Iverson bracket. It is also possible to cluster the entire feature vectors regardless of their memberships to come up with the same cluster centers for both classes $C_1 = C_2 = C$. This approach is specifically not good if the feature vectors of one class dominate the feature space, as the resulting cluster centers will be in favor of the dominating class, leaving the representation of the other class much less accurate.

This approach is non parametric and fast to compute specially with efficient implementations of k-means(e.g. [20]). The parameter k in this approach should be known beforehand and therefore, it is possible to use heuristics or specific criteria mentioned below to find reasonable ks. Furthermore, k-means clustering assumes hard memberships of the feature vectors to the cluster centers and also uncorrelated covariance matrices with the same standard deviation on all dimensions for cluster centers. Therefore, data scaling is crucial in k-means clustering approaches.

Gaussian Mixture Models: Alternatively, it is possible to utilize GMMs computed from an Expectation Maximization approach to model the desired pdf. GMMs compared to the k-means clustering approach with the same number of cluster centers have some advantages:
1- GMMs do not assume hard membership functions(each feature vector can be assigned to a cluster center with a certain probability) neither in the training time nor in the evaluation time.
2- GMMs can model correlated Gaussians with different standard deviations on each dimension. As a result, this approach is scale-invariant.
3- Gaussian Mixture Regression models continuous pdfs among the entire space and therefore, pdfs represented by GMRs are smooth and differentiable in the entire feature space.
The main drawback of using GMMs is their computational costs compared to the k-means approach. The pdf defined by a GMMs with k Gaussians on f_j for the ith

class is defined as:

$$\begin{aligned} p_i(f) =& \sum_{k=1}^{K} p_i(f|k)\bar{p_i}(k) \\ p_i(f|k) \sim& \mathcal{N}(f;\mu_{i,k},\Sigma_{i,k}) \\ =& \frac{1}{\sqrt{(2\pi)^D|\Sigma_{i,k}|}} e^{-\frac{1}{2}\left((f-\mu_{i,k})^T\Sigma_{i,k}^{-1}(f-\mu_{i,k})\right)} \end{aligned} \tag{2.13}$$

where $\bar{p_i}(k)$ is the prior for the kth Gaussian modeled by the GMM.
Similar to the k-means approach, GMM requires the number of Gaussians to be known in advance. One can utilize different criteria(e.g. Minimum Description Length(MDL), Bayesian Information Criterion(BIC) or heuristics) to chose the number of Gaussians. As BIC assumes exponential underlying data distribution([42]), and as GMMs do assume the same, a suitable criterion here is to use BIC similar to [13]. The optimal K for the ith class according to the BIC criterion($K_{BIC}^{*(i)}$) is defined as:

$$\begin{array}{lcl} S_{BIC}^{(i)}(k) & = & -\mathcal{L}^{(i)} + \frac{n_p(k)}{2}\log|c_i| \\ \mathcal{L}^{(i)} & = & \sum_{j\in c_i}\log p(f_j) \\ n_p(k) & = & k-1+k(D+\frac{1}{2}D(D+1)) \\ k_{BIC}^{*(i)} & = & \arg\min_k S_{BIC}^{(i)}(k) \end{array}$$

where $\mathcal{L}^{(i)}$ is the log likelihood of the model for ith class, $|c_i|$ is the number of pixels labeled as belonging to the ith class, $n_p(k)$ is the number of parameters for a mixture model of k Gaussians(k parameters for priors -1 for overall scaling + kD parameters for means + $kD(D+1)/2$ parameters for (symmetric, positive definite) covariance matrices). The first term measures how well the model fits the data and the second term is a penalty factor, which tries to minimize k([13]). Following this approach, GMMs with different number of components are trained and the one with the minimum score is chosen.

It is obvious that using one Gaussian to represent the pdf is a special case of GMM using $k = 1$. It is also worth noting that GMMs are considered in the parametric family of estimators and therefore, they are less powerful to represent arbitrary pdf compared to the non-parametric methods(for reasonably small ks). However, for sufficiently large ks, GMMs show the desired data-adaptivity as the non-parametric methods [48].

Parzen window: It is also called Kernel Density Estimation(KDE) which is a non-parametric way of estimating the pdf of random variables. The unscaled pdf in this approach is represented by:

$$\tilde{p_i}(f) = \frac{1}{|c_i|\sigma}\sum_{j\in c_i} K(\frac{f-f_j}{\sigma}) \tag{2.14}$$

where the kernel function $K(.)$ is usually assumed to be a Gaussian with standard deviation equal to 1[3]. The scaled pdf is then defined as:

$$p_i(f) = \frac{\tilde{p_i}(f)}{\sum_t \tilde{p_t}(f)}$$

Parzen window is somewhat sensitive to the bandwidth parameter σ and performs poorly if the data is very sparse. If the parameter σ is large, the pdf will be over-smoothed and if it is too small, it will overfit the data. The Parzen window with such kernel function, can be thought of as a special case of GMM with $k = N$ and with isotropic covariance matrices with standard deviation σ and the same priors for each component $1/N$ [48]. Furthermore, as it is mentioned in [6][10], Parzen window comparatively produces more local minima compared to the other methods in the minimization process, provided that σ is not large.

As the membership function $p_i(x) = \delta(x \in \Omega_i)$ is updated after each iteration of the segmentation algorithm(at least for some regions/pixels), one needs to be aware of the computational costs of each approach. Parzen window approach does not suffer this fact as it is a lazy learner and does not process any information during the update of the membership function postponing all processes to the estimation of the likelihood itself. On the other hand, approaches like GMM(Expectation Maximization) are computationally much more involved and re-initializing the entire clustering process in such approaches would lead to extremely redundant computational overhead. However, it is possible to initialize the EM(or k-means) approach, using the last estimates of the cluster centers, which are much closer to the new local minima compared to a random initialization leading to reduction of the update time drastically. Furthermore, searching for optimal number of k in GMMs is not possible in each iteration. Therefore, one needs to define a measure or heuristics about when to update the estimate for k^*. A simple heuristic can be defined such as: update k^* as soon as the number of pixels in either class changes by a constant κ_j or in the case of multi scale segmentation, as soon as the scale changes.

Independency Assumption: In order to further simplify the likelihood estimation, one can assume that the features F_k composing $F(x) = F_1(x) \times F_2(x) \times ... \times F_{N_F}(x)$ are independent leading to:

$$p_i(F(x)) = \prod_{k=1}^{N_F} p(F_k(x)|c_i) = \prod_{k=1}^{N_F} p_{i,k}(x)$$

which reduces the problem to the estimation of N_F independent probability distribution functions of less dimensions(d_k). However, this assumption in general is not valid as the features of an object(e.g. its color and its texture) are not independent. This can be verified easily by considering a case of two objects with equal sizes each one consisted of two equal sized regions: object 1: red texture 1 + blue

[3]In this case, the kernel function works on the norm of its argument.

texture 2 and object 2: red texture 2 + blue texture 1. In such cases, independent color and texture features cannot distinguish between the two objects. However, as the joint probability density estimation of a high dimensional random variable is computationally expensive, the features are usually assumed to be independent.

In order to further simplify the pdf estimation, usually the feature channels are assumed to be independent too, simplifying the problem to the estimation of D one dimensional pdfs:

$$p_i(F(x)) = \prod_{k=1}^{D} p_{i,k}(x)$$

Assuming independent feature channels, the functional reads:

$$E(\Phi) = \int_\Omega \left(-H(\Phi) \sum_{k=1}^{D} \log p_{1,k} - (1 - H(\Phi)) \sum_{k=1}^{D} \log p_{2,k} + \nu |\nabla H(\Phi)| \right) \texttt{dx} \quad (2.15)$$

This way, the curse of dimensionality is avoided making very fast and accurate 1-D pdf estimation applicable to the problem. In practice, the feature vectors are scaled such that they have the same dynamic range in all dimensions and afterwards, the pdfs derived from a KDE are sampled on a fine resolution 1-D grid. Therefore, the update of the algorithm is very efficient($O(n)$) as well as the estimation of the likelihoods(a division per pixel per feature channel: $O(Dn)$). As the spatial bins are fixed in this approach, the update process can be made even more efficient by considering only the pixel that changed their membership functions.

2.5 Smoothing, re-scaling and weighting the Feature Space

Regardless of the approach used to estimate the likelihoods, noisy pdfs(or pdfs with several peaks) lead to several local minima in the energy minimization process. Therefore, in order to get less peaks in the process, it was suggested in [6] to smoothen the feature space prior to the pdf estimation. A vector valued isotropic coupled nonlinear diffusion of the feature space was suggested:

$$\partial_t F_i = \texttt{div} \left(g \left(\sum_j |\nabla F_j|^2 \right) \nabla F_i \right) \quad (2.16)$$

where the ϵ-regularized diffusivity function $g(|\nabla u|^2) = \frac{1}{(|\nabla u|^2 + \epsilon^2)^{\frac{p}{2}}}$ with the control parameter $p \in \mathbb{R}^+$ determines the behavior of diffusion process[4]:

- $p = 0$ leads to linear diffusion(i.e. gaussian filtering)

[4]For more information regarding nonlinear diffusion the reader is referred to [59]

- $0 < p \leq 1$ constraints the evolving field to solely lose mass but with different speeds(determined by the mass of the point and the mass of the neighboring points) for different points in the feature space. Non-linear diffusion with $p = 1$ is called Total Variation(TV) flow.

- $p > 1$ lets the points in the feature space to both gain and lose mass with different speeds. The non-linear diffusion with such ps is called Edge Enhancing flow.

The stopping criterion for the diffusion process can be either a fixed time or a more dynamic criterion such as a threshold on some measure of smoothness of the evolving feature space. Such a measure of the smoothness of the feature space can be the average total variation of the field(after discretization) per pixel per channel [6]:

$$\rho(F) = \frac{1}{N_F M} \sum_x \sum_{i=1}^{N_F} |\nabla F_i(x)|$$

where N_F is the number of channels(dimensions) of the feature vector F and M is the number of pixels. ρ is independent of the dimension and the extent(the number of pixels) of F. As the coupling of the different channels depends on the magnitude of the gradient of the channel at each pixel, different channels need to be scaled properly so that they have the desired effect on the evolution process. Features can be scaled such that they have the same dynamic range so that they have the same effect in the diffusion process. They can also be scaled to different scales leading to more focus on specific channels.
It is also possible to modify the functional (2.15) so that it weights different feature channels irrespective of the feature scaling and the choice of the kernel bandwidth. Using such approach, the functional reads:

$$E(\Phi) = \int_\Omega \left(-H(\Phi) \sum_{k=1}^{D} \alpha_k \log p_{1,k} - (1 - H(\Phi)) \sum_{k=1}^{D} \alpha_k \log p_{2,k} + \nu |\nabla H(\Phi)| \right) \texttt{dx}$$

This can be thought of as repeating kth channel α_k times in the feature space before computing the likelihoods putting more stress on the channels with bigger weights. However, this approach does not have a direct Bayesian interpretation and therefore, it is not investigated more in this thesis.

While re-scaling of feature channels, one needs to also consider the tuning of the kernel bandwidth in case of 1-D KDE. Reasonable scales and corresponding bandwidths for each channel can be learned using cross-validation of some hypotheses using some measure of goodness of the resulting segmentations such as the Normalized Probabilistic Rand(NPR) index [50]. However, in this work, searching for optimal scales are not performed and instead, feature channels are scaled to the same dynamic range i.e. [0,255].

2.6 Initialization and Convergence

As the gradient descent approach converges to the first local minimum of the functional being minimized, continuous variational methods do not guarantee convergence to the global optimum of the functionals. However, if the feature space is smoothed enough(using the techniques such as the one mentioned in Section 2.5), and the bandwidths are chosen large enough, the pdfs will not be very peaky and it can be expected that solutions sufficiently close to global optima are reached using the gradient descent approach. The Chan-Vese method is independent of the initialization and always converges to the global optimum. However, it should be noted that the feature vector in that case is one dimensional and by nature produces less minima in a 1-D KDE approach compared to the same situations with multiple dimensions. Assuming independent channels, since the channels are multiplied together, the number of total minima grows exponentially with the number of dimensions. This is the case in the texture segmentation problem(e.g. [11][12]), as well as the case with more complicated and higher dimensional feature spaces. Therefore, in such situations a smoothing scheme and larger bandwidths are necessary in order to avoid local minima.

Another way to prevent local minima is the idea of coarse to fine(multi scale) processing of the data. Starting at the coarsest level, the feature space is much smoother leading to fewer local minima while speeding up the computations as the coarse level contains fewer pixels. At the end of each level, the level set is re-scaled to the size of the next(finer) scale and the evolution of the level set is resumed in that scale. This approach is proven to be very effective in most of the variational approaches such as optical flow estimation or texture segmentation. Therefore, a similar approach is utilized here.

As the object of interest is expected to roughly be located in the center of attention in this problem, the level set is initialized using a box function located at the center of image. However, it is not necessary for the algorithm to cover the entire object or to contain solely the object of interest at the initialization step.

2.7 Features

The set of features F in (2.6) consists of :

Color: While color alone is not a very strong cue, in combination with other cues(e.g. texture) it is proven to be a very strong cue for object classification and similar problems [33]. In this work, color with CIE Lab color space is used as a cue(as it is also suggested in [18]).

Texture: It has been shown that a simple descriptor such as [52] can be used to recognize between different textures(and thus, objects with different textures). While more sophisticated approaches such as [31] and [24] exist, they are mostly

used as a means for texture classification rather than discrimination of local texture. Also, the dimension of the resulting feature vector affect the likelihood pdf estimation process and the focus should be on low dimensional discriminative texture descriptors in a variational segmentation framework. Therefore, although features like Histogram of Oriented Gradients [19] have been used to discriminate the texture of objects, their high dimensional feature responses make them less useful directly in the current framework. Although it is possible to use dimension reduction approaches such as PCA to limit the dimension of the feature descriptors, it is not clear how this transformation will affect the discriminative power of the feature vector as there is no reason to expect the discriminative information to lie in the direction of maximum variance of the data in this case.

A descriptor such as [52] can be utilized with sufficiently small spatial support leading to a low dimensional feature vector. Reasonable choices for the spatial support for feature vectors in this case are 1 and 2 pixel neighborhoods leading to 9 and 25 dimensional feature vectors respectively. However, one needs to be aware of the fact that discriminative power of such feature vectors can only be achieved where feature channels are considered jointly and if the feature channels are to be treated independently as it was suggested in Section 2.4, such feature vectors which do not propagate information between feature channels will perform poorly.

A very low dimensional feature vector based on the structure tensor has been proposed [40][11] which consists of:

- The three channels of the structure tensor:

$$J_0 = \nabla I \nabla I^T = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

  undergone a non-linear coupled isotropic diffusion(2.16). The three channels reflect the texture magnitude, orientation and homogeneity.

- Inverse texture scale acquired from the average speed of mass exchange of pixels in a total variation smoothing of the intensity of the image:

$$\frac{1}{\bar{m}} = \frac{1}{4} \frac{\int_0^T |\partial_t I| dt}{T - \int_0^T [\partial_t I == 0] dt}$$

  Because of the properties of the TV flow[6], as the pixels belonging to larger regions(and therefore larger texture scales) change their intensities slower, their inverse scale will be smaller and vice versa [5].

- Texture intensity to count for the contrast of the texture. Also, including the intensity of pixels in the diffusion process plays a "filling in" effect for the areas where gradient does not exist [11].

[5] for a more detailed motivation, the reader is referred to [6].

Figure 2.1. A sample image and the corresponding texture feature(threshold on the average total variation per channel per pixel $\theta = 2$). From top left to bottom right: original image, the intensity channel, structure tensor component I_x, structure tensor component I_y, structure tensor component I_xI_y and scaled inverse texture scale.

This 5 dimensional feature vector has proven to be a powerful cue for the segmentation purpose in a variational framework [18] and therefore, it was chosen to be the texture descriptor in this work. Similar to the smoothing of the feature space, the texture feature space is smoothed(instead of just the component of the structure tensor) after proper scalings of each dimension. For more information the reader is referred to [6] or [11]. Figure 2.1 depicts the texture feature computed for a sample image.

Motion: Many approaches estimate the apparent motion between two or more images. Some approaches define steerable spatio-temporal filters [44]. Some approaches use feature matching to compute the motion of pixels. A recent approach following this scheme is [26] which uses SIFT features [27] to compute the motion field associated with two images. Optical flow computation is an alternative approach to compute the motion of pixels in two or more images and it has been the focus of many researches for many years. Although optical flow is considered a solved problem for small displacements, in the case of large displacements it is still an unsolved problem [9]. However, recent advances in optical flow estimation makes them a powerful and reliable tool to compute the apparent motion.

Optical flow estimation algorithms are generally divided into two categories: local([28] and its derivations) and global methods([22] and its derivations). Global methods have proven to be more accurate but require much more time to converge([7], [61], [65]). However, recently, efficient implementation of global methods have acquired near-real-time performance on GPUs [61]. Global methods produce dense

flow estimates and are considered the state of the art algorithm in optical flow estimation having sub-pixel accuracy. In this work, variational optical flow algorithms are utilized as motion cues.

The brightness constancy assumption(that is the main assumption of optical flow algorithms[6]) states that the intensity of a moving pixel does not change over time. Taylor expansion of the brightness constancy assumption: $I(x,t) = I(x+u(x), t+1) = u(x)^T \nabla I + I_t$ leads to the well known optical flow constraint: $u^T \nabla I = -I_t$, which is one equation in two unknowns(the x and y components of u). This is known as the aperture problem and basically, it means that the optical flow can only be determined in the direction normal to the direction of the gradient of the image. Therefore, additional assumptions or information need to be integrated in the optical flow estimation process. One option is to integrate the information from the neighboring pixels which leads to the local methods. An alternative which leads to global methods is to put a prior into the algorithm and favor specific types of functions(e.g. smoother results). Global methods aim to minimize a functional of the form:

$$E(u) = \int_\Omega \left(\underbrace{|I(x,t) - I(x+u(x), t+1)|^L}_{\texttt{Data Term}} + \lambda \underbrace{g(\nabla I, \nabla u)}_{\texttt{Smoothness Term}} \right) \texttt{dx} \tag{2.17}$$

The smoothness term reflects the underlying assumption of the global methods and hence, the choice of the smoothness term affects the outcome of the algorithm considerably. The first global method [22] used a homogeneous regularizer: $g = |\nabla u_x|^2 + |\nabla u_y|^2$. Later on, more sophisticated smoothness terms were suggested such as [56][65]. The smoothness term in general falls into two different categories: Image based and Flow based[7]. Image based regularization uses image edges to favor specific types of flow while flow based regularization uses the evolving flow field itself to direct the evolution of the flow field. A taxonomy of possible choices for the smoothness term are available in [60]. Figure 2.2 depicts the forward and backward flows(Section 3.1) computed between two frames.

2.8 Summary

In summary, the functional is defined as:

$$E(\Phi) = \int_\Omega \left(-H(\Phi) \sum_{k=1}^{D} \log p_{1,k} - (1 - H(\Phi)) \sum_{k=1}^{D} \log p_{2,k} + \nu g(I_B) |\nabla H(\Phi)| \right) \texttt{dx} \tag{2.18}$$

[6]Although recent developments in the optical flow estimation field involve gradients of the image to deal with intensity shifts and add color information, they still assume the constancy of some component(s). Also, although there are methods that postpone the linearization of the data term, they do linearize the data term at some level and therefore, the aperture problem still is a part of any optical flow algorithm.

[7]More sophisticated smoothness terms usually combine the information from both cues: image and the evolving flow and/or add other types of priors to the smoothness term.

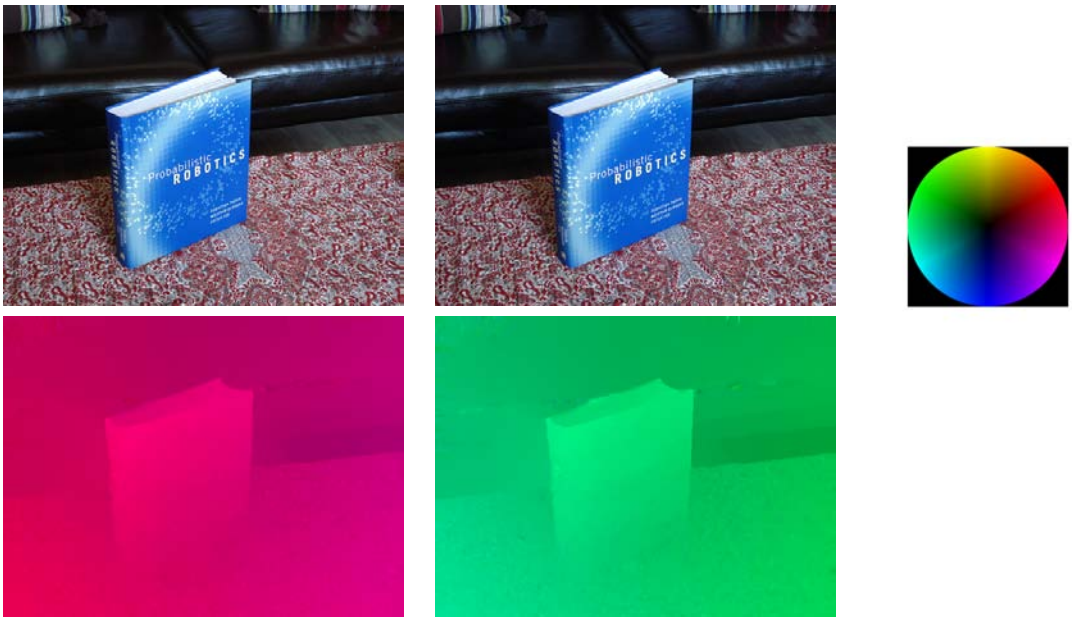


Figure 2.2. Two frames and the forward and backward flows(motion) computed for the two frames using [61] and the corresponding color code. The hue represents the direction of the flow vector and the saturation represents its magnitude.

in which H is a regularized step function, Φ represents a level set, F represents the D dimensional feature function, $p_{i,k}$ represents the conditional probability(the likelihood) of the k-th dimension of the feature vector F given the class i, ν represents the weight of the regularizer and I_B is an image containing the desired sets of edges the segmentation is encouraged to converge to. The mentioned functional leads to the following evolution equation:

$$\partial_t \Phi = H'(\Phi)\left(\sum_{k=1}^{D} \log \frac{p_{1,k}}{p_{2,k}} + \nu \texttt{div}\left(g(I_B)\frac{\nabla\Phi}{|\nabla\Phi|}\right)\right) \tag{2.19}$$

Using the mentioned evolution equation, the algorithm is summarized in Algorithm 1.

Algorithm 1 2-Frame Variational Segmentation using Level Sets

Inputs: Images I_0, I_1 with dimensions $s_y \times s_x$
Compute backward flow u_b between I_0 and I_1
Compute the intermediate boundaries I_B on I_1
Initialize $\Phi_{(s_y \times s_x)}$
for scale $s = s_0$ to 1 **do**
  re-scale $\Phi \xrightarrow{s} \Phi$, $I_1 \xrightarrow{s} I_s$, $u_b \xrightarrow{s} u$ and $I_B \xrightarrow{s} I_B$ using bi-linear interpolation
  Compute feature vector F_0 from appearance based cues(from I_s) and u
  Filter F_0 and store the smoothed version in F
  repeat
    Estimate the pdfs of the feature channels for each class
    Evolve Φ using (2.19)
  until Φ is stable or iteration limit exceeds
end for
return Φ

Chapter 3

The Intermediate Boundaries

In this section, a classification / regression approach to object boundary (occlusion boundary) detection is proposed using appearance and motion cues. Although the application of such detectors is not limited to the segmentation process, a possible approach to use such detector in the segmentation process is to use GAC (2.11) to absorb the contour of the evolving level set to the detections of such detector. Although it is possible to use a simple edge detection algorithm such as the Canny edge detector on the image itself or the smoothed image using sophisticated approaches such as [37] or the TV smoothing or basically any appearance based edge / boundary detector to get I_B (2.18), one has to consider the situation where I_B has many strong edges which do not correspond to the actual boundaries of the objects. Having many false positives[1] in I_B introduces many unwanted local minima to the energy functional and might force the algorithm to converge to unwanted regions. Therefore, by refining I_B, one can avoid the unwanted local minima. The refining method suggested here is to combine multiple motion and appearance based cues using a classification or a regression framework.

3.1 Cues for intermediate boundaries

Initially, we have five cues for the intermediates boundaries($g(I_B)$). The forward and backward flow fields are defined as:

$$\begin{aligned} I(x,t) &= I(x + u_f(x), t+1) \\ I(x,t+1) &= I(x + u_b(x), t) \end{aligned}$$

Obviously, the forward flow field(u_f), operates on the second image and the backward flow field operates on the first image. In the forward flow field, flow vectors correspond to the pixels in the first image(e.g. a pixel located at x_1 in the first image moves to $x_1 + u_f(x_1)$ in the second image). Therefore, warping a quantity

[1]Here, what is meant by false positives is the detections which do not have any correspondence in the desired ground truth information.

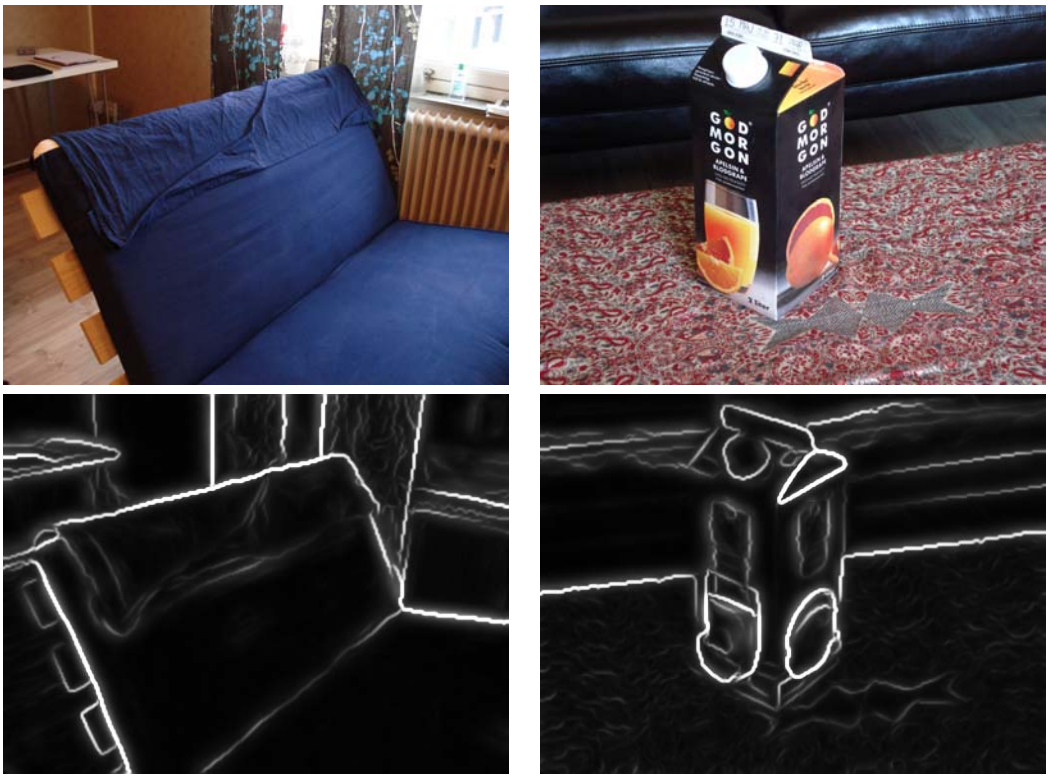


Figure 3.1. Two images and the result of gPb detector on each sequence

in image one according to $-u_f$, moves it to its corresponding location in the image two. The boundaries are to be estimated to match the boundaries of the objects in the second frame($I(x, t+1)$). Figure 2.2 shows the forward and backward flows computed for a sequence.

In the following, $P(F)$ represents the probability of pixel x being on the desired object boundary given the feature F and f_F represents an arbitrary function taking as input the mentioned value(s) used in the feature F. If not stated otherwise it is assumed that: $f_F(F_1, F_2) = \frac{1}{2}(F_1 + F_2)$ and $f_F(F_1) = F_1$.

Appearance based edge detector (gPb): The gPb edge detector [29] is currently one of the most robust and accurate edge detectors, which utilizes color and texture information. Figure 3.1 shows the results of the gPb detector on two sample images. It can be observed that while the detector is able to distinguish the boundaries between regions with different textures and colors, such boundaries do not necessarily correspond to object boundaries.

Warping error: Incompatibility between the first image and the second image warped toward the first image can have its roots in the incorrect flow estimated at

occlusion boundaries [62]:

$$P(W_0) = f_W\left(|I(x,t) - I(x+u_f(x),t+1)|, |I(x+u_b(x),t) - I(x,t+1)|\right)$$

in which, u_f represents the forward flow field and u_b represents the backward flow field. As the forward flow field is applied to $I(x+t)$, for the first and the second terms to be well-localized, the second term needs to get warped towards the first term once more:

$$\begin{aligned} P(W_1) =& f_W\left(|I(x,t) - I(x+u_f,t+1)|(x-u_f), |I(x+b_f,t) - I(x,t+1)|\right) \\ =& f_W\left(|I_1 - I_2^{\mathbf{w}(u_f)}|^{\mathbf{w}(-u_f)}, |I_1^{\mathbf{w}(u_b)} - I_2|\right) \end{aligned}$$

The contrast invariant form of the mentioned cue is:

$$P(W) = f_W\left(||\nabla I_1 - \nabla I_2^{\mathbf{w}(u_f)}||^{\mathbf{w}(-u_f)}, ||\nabla I_1^{\mathbf{w}(u_b)} - \nabla I_2||\right) \tag{3.1}$$

Divergence of the flow field: As the flow field is usually assumed to be piecewise-continuous, the discontinuities of the flow field can be used as a cue for existence of depth-discontinuities. Similar to the previous cue, the second term needs to be warped towards the first term:

$$P(D) = f_D\left(|\texttt{div}\ \ (u_f^{\mathbf{w}(-u_f)})|, |\texttt{div}\ \ (u_b)|\right) \tag{3.2}$$

Incompatibility of the forward and backward flow fields: Similar to the previous cue, the areas in which the forward and the backward flow fields are incompatible can reflect a depth-discontinuity in the image. A slightly different version of this cue was presented in [44]:

$$P(I) = f_I\left(|\texttt{div}\ \ (u_f^{\mathbf{w}(-u_f)} + u_b)|\right) \tag{3.3}$$

Gradient Magnitude of the norm of the flow field: As we have the translative motion assumption, norm of the flow vector at each pixel represents its disparity if the problem is viewed in a stereo configuration. Thus, it has a direct relation with the inverse depth of the corresponding real world point. Therefore, discontinuities in the norm of the flow field can be interpreted as depth discontinuities:

$$P(N) = f_N(||\nabla\sqrt{u_{f,x}^2 + u_{f,y}^2}||^{\mathbf{w}(-u_f)}, ||\nabla\sqrt{u_{b,x}^2 + u_{b,y}^2}||) \tag{3.4}$$

3.2 Combining the cues

Using the ground truth information of N samples $Y_{1\times N}$ in either the binary form or in a probabilistic form, and feature vectors $X_{|F|\times N}$ where $|F|$ is the number of features, a classifier(or a regressor) can be defined which could detect if an object is on an object's boundary(or estimate the probability of a pixel being an occluding boundary). There are many possible methods to combine the cues in a strong

classifier. In general, the approaches fall into two categories below.

Regression methods learn the relation between one or more independent variables and one or more dependent variables. The relation is usually expressed as a conditional probability distribution function. In this case, a regression framework should infer the relation between the cues(independent variable) and the probability of a pixel being an occluding boundary. Regression methods in general estimate parameter vector b such that $Y = f(X, b) + \epsilon$.

Classification methods learn a function which maps an input to an integer number representing the class of the object. They are not designed to give probability estimates but rather the integer numbers. However, some of them are able to give confidence bounds based on the distance of the input to the decision boundary. Classification methods in general estimate parameter b such that $Y = \texttt{sign}(f(X, b))$(in 2-class problems).

Some possible approaches for the purpose of combining the mentioned cues to compute the intermediate boundaries are:

Linear Regression is the simplest form of regression and has the shortest training time of the rest of the methods. The simplest form of linear regression is the ordinary least squares in which a weight vector $b_{|F|\times N}$ is to be learned that minimizes $||Y - b^T X||$. While ordinary least squares is a powerful method to find a hyperplane fitted to the data, it is sensitive to outliers.

Generalized Linear Regression as a generalization of linear regression can be used to find b such that $Y = g(b^T X)$, where g is a link function(the desired distribution function for Y) such as Normal, Binomial or Poisson distributions [30]. It is in general more robust to outliers while being able to model the conditional PDF of Y to have specific forms.

Logistic Regression models the conditional probability of each class in Y given X using logistic functions. It assumes the form of $Y_i(X) = \frac{1}{1+e^{-b^T X}}$.

Neural Networks are well known for their capability to represent arbitrary functions provided that the network is given enough neurons and hidden layers to model the complexity of the data. While radial basis neural networks(generalized regression networks and probabilistic neural networks [55]) are powerful tools for nonlinear regression, their limitations e.g. memory consumption and model complexity prevent their usage in large problems. Multi Layer Perceptrons on the other hand do not have such limitations and can be used to perform nonlinear regression in large problems. Their over-fitting characteristics can be dealt with to some extent using cross validations(e.g. with 10 fold cross validation).

Boosting is well known for its power to combine many weak learners to create a strong classifier while requiring reasonable resources(computational and memory resources). It was originally introduced by [41] and later improved to AdaBoost, Real AdaBoost and gentle AdaBoost and its popular application in face detection is well known to computer vision community[54]. It has been shown that boosting can be considered as a regularized maximum margin classifier [39] using which, more robust versions of boosting have been introduced[53].

Gaussian Mixture Models model the pdf of each class using GMMs. In Gaussian Mixture Regression approach, the joint probability distribution of X and Y is modeled by GMMs and in the test time, the probability distribution of Y is inferred from an observation X. In case of binary classification, the a posteriori ratio $r(X) = \frac{p(X|c_1)p(c_1)}{p(X|c_2)p(c_2)}$ is thresholded on 1 to perform the classification where each likelihood $p(X|c_i)$ is represented by a GMM. There is no optimal criterion to select the number of component in GMM but several criteria have been presented in [42][48][13].

Support Vector Machines (SVMs) [51] are well known for their capability to generalize very well(as they are maximum margin classifiers) while being very accurate. They implicitly map the data to higher dimensions with the argument that the data will be more spread and easier to separate in higher dimensions and find a maximum margin separating hyperplane. Using kernels, they avoid the actual mapping of the data to higher dimensions and the mapping is implicitly done using inner products in the same dimension as the data. The main drawback of SVMs is their computational costs(both in training time and the testing time). SVMs can be used in a classification framework(C-SVC or ν-SVC) or in a regression framework(ϵ-SVR or ν-SVR).

3.3 Localized Classifiers

Despite the fact that kernel SVMs are well known for their accuracy and generalization powers, there are practical limitations associated with them. Although using the concepts like working set and shrinking [15], there have been improvements regarding the memory requirements and computational costs of SVMs in the training time, kernel SVM training is still extremely slow for huge problems (e.g. training a kernel SVM without cross-validation with one set of parameters for a problem with 10 million 20 dimensional feature vectors takes around 1 week to be completed!). Searching for the optimal parameters(even with proper scaling of the data) and n-fold cross validations makes the training process even slower. Another limitation is the number of support vectors and the memory costs. As the complexity of the decision boundary (the manifold separating the positive and negative

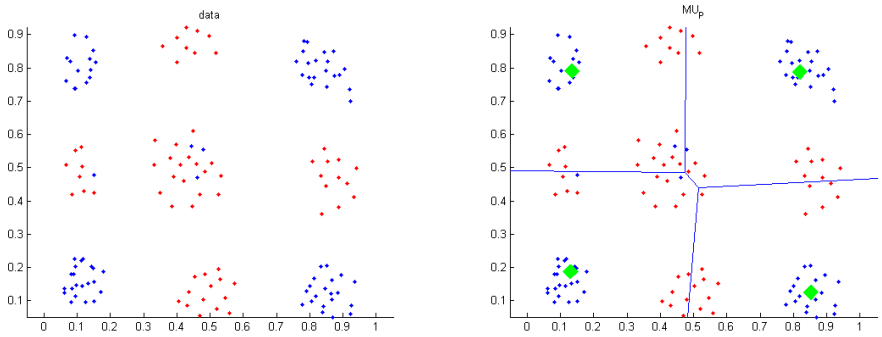


Figure 3.2. A synthetic data(left) and a partitioning of the data(right). The green dots represent cluster centers and the blue lines represent the boundaries of the partitions.

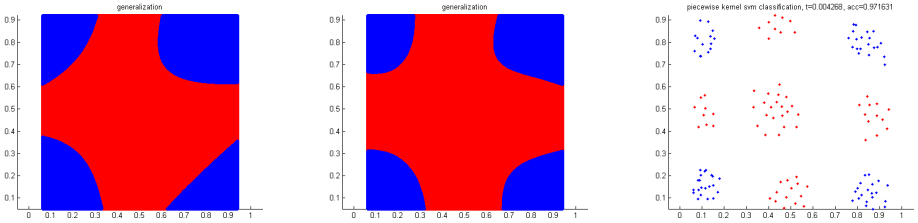


Figure 3.3. The generalization of a localized classification approach to the data and partitioning in Figure 3.2(left), the generalization of a kernel SVM(middle). Both of these generalizations lead to the same classification of the data (right)

regions in a 2-class problem) increases, the number of support vectors grow making the test phase of the SVMs computationally involved too. Alternatively, boosting based approaches combine many weak classifiers which are easy to train and evaluate to represent arbitrary complex decision boundaries. However, the boosting based methods train their weak classifiers for the entire feature space which again, leads to computational involvement in case of huge data-sets.

The idea of localized classifiers, as the name suggests, tries to train many classifiers each one being able to classify features in a specific volume of the feature space(and therefore, they are localized to some area in the feature space). This way, the spatial extent of the effect of each classifier is limited to specific regions and only with the global combination of these classifiers, the global decision boundary is defined. Figure 3.2 depicts a synthetic data and a partitioning of the data. The argument here is that a localized classifier expertized to classify one of these partitions does not need to consider all feature vectors in the data (e.g. points that drop into other partitions). Of course, the partitioning itself is a problem and also, the hard assignments of the data to partitions introduces bias towards the specific choice of the partitioning. This fact is depicted in Figure 3.3: A localized classification approach with a polynomial order 2 SVM classifying the features in each

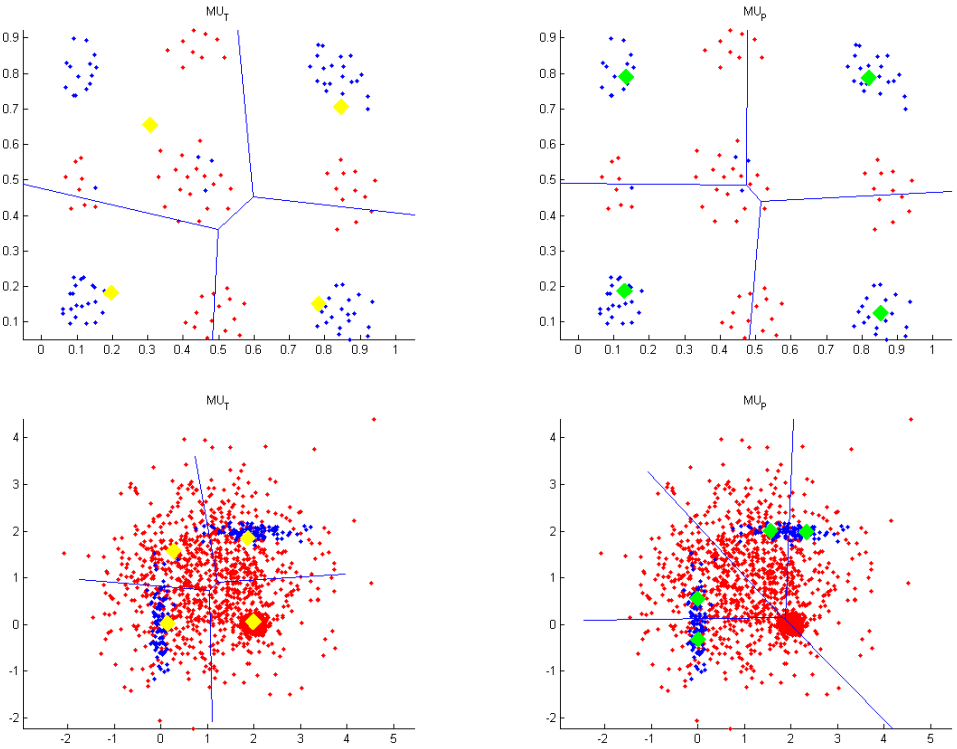


Figure 3.4. The partitioning defined by the clustering of the entire data(left) and the positive data(right) on two different problems(top/bottom)

partition and a Kernel SVM approach lead to the same classification of the data with the kernel SVM approach producing a smoother decision boundary.

It is possible to train classifiers for subsets of data(e.g. the clustered data), which by nature leads to speedups and also makes it possible to use arbitrary classifiers for arbitrary sized data-sets. As in most of the computer vision problems, the number of positives is much less than the number of negatives(e.g. number of faces vs the number of non-faces, number of occluding boundaries vs non-occluding boundaries and etc), the clustering of the data is performed on the positive set, which is much faster than the clustering of the entire data. Furthermore, in such clustering there is more chance that the positive data appear in a relatively more compact area inside the resulting partitions as the partitions are defined by the Voronoi diagrams of the resulting cluster centers. Figure 3.4 reflects the same fact.

Regarding the clustering method, a simple clustering technique such as k-means clustering can be utilized. As it was mentioned before, hard assignment of feature vectors to the nearest cluster introduces slight bias towards the resulting cluster centers in the approach. To decrease this bias, one might want to consider adding nearest data points from neighboring clusters to each cluster. There are some

possible ways to do this, but the argument here is that if the feature space is well-populated, the resulting nearest-neighbor assignments do not introduce a significant bias. As it is the case in the current problem(there are around 2 million sample feature vectors available to the method), the idea of decreasing this bias is not investigated more here. Regarding the number of clusters, it is possible to use one of the mentioned optimality measures in Section 2.4. A more efficient approach is to cross-validate the resulting localized classifiers set for different number of partitions and pick the best number of clusters.

The choice of the classifiers for this method is not crucial as long as the distribution of the sample feature vectors in resulting partitions match the VC dimension of the chosen classifier i.e. even a linear classifier can estimate arbitrary decision boundaries. However, as the clustering was chosen to be performed on the positive set and as this was expected to lead to the positive samples appearing at the center on the resulting partitions, the VC dimension of the classifier needs to be in general at least 3 i.e. a linear classifier will not be useful in general cases. A suitable classifier type for this purpose is [16], which is able to utilize degree 2 polynomial kernels in a linear SVM training framework leading to significant speedups in training/testing time. In general, it is possible to use parametric(e.g. Gaussian Mixture or Polynomial Kernel) and non-parametric(e.g. radial basis kernel) classifiers, but if the speed is of concern, parametric classifiers are preferred. Figure 3.5 compares the results of a piecewise polynomial degree 2 kernel SVM and a radial basis kernel SVM in a synthetic data with 4 partitions and a global radial basis kernel SVM. It is observable that the bias does not affect the radial basis based classifier seriously because there is enough data to infer the decision boundary with or without the clustering. A small bias is introduced when the parametric classifier is used and the generalization is slightly worse. However, as the number of training samples increases, the effect of bias gets weaker. Figure 3.6 shows the same fact for the specific data-set. As the focus here is not theoretical development of localized classifiers, but rather the development of a practical method to make efficient use of large data, the localized classifiers are not elaborated more here.

3.4 Making richer cues

There are some possible ways to make the feature vectors more informative for the purpose of classification. For instance, it is possible to augment the feature vector with the integrated information from neighboring areas simply by an approach like Gaussian smoothing or more sophisticated approaches such as non-linear diffusion mentioned earlier. Also, in order to be able to achieve comparable results to radial basis kernel classifiers, a subtle manipulation of the feature space can be utilized with the following motivation: Radial basis kernel classifiers are able to implicitly map the feature space to an infinite dimensional space, where a separating hyperplane corresponds to arbitrary complex shaped decision boundaries in the original feature space. This is not the case with general parametric classifiers and therefore,

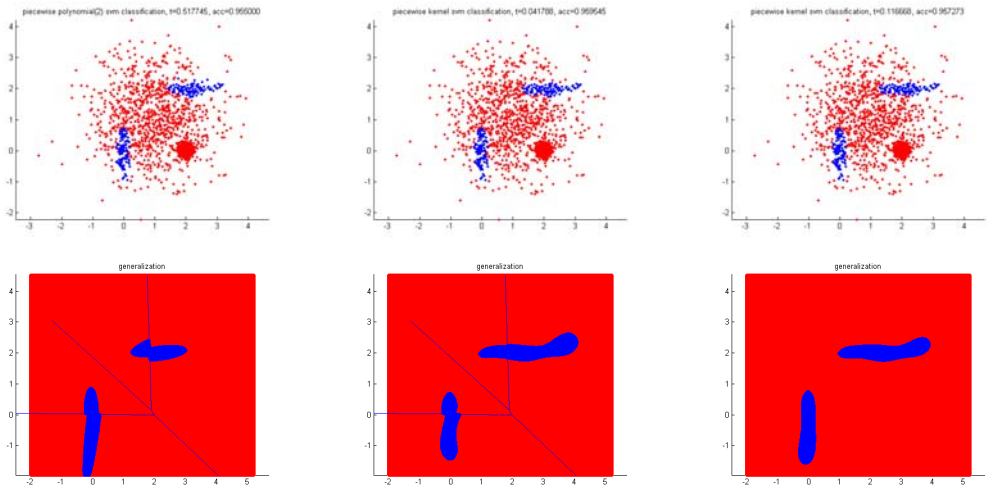


Figure 3.5. The classification results(top) and the generalization(bottom) of the different classifiers on a synthetic data. Piecewise polynomial degree 2 kernel SVM(left), Piecewise radial basis kernel SVM(middle) and global radial basis kernel SVM(right). Training at the test time of the global classifier and the piecewise radial basis classifier are 50 and 10 times more than the piecewise polynomial kernel based classifier while the accuracies does not differ significantly(less than 0.5 percent).

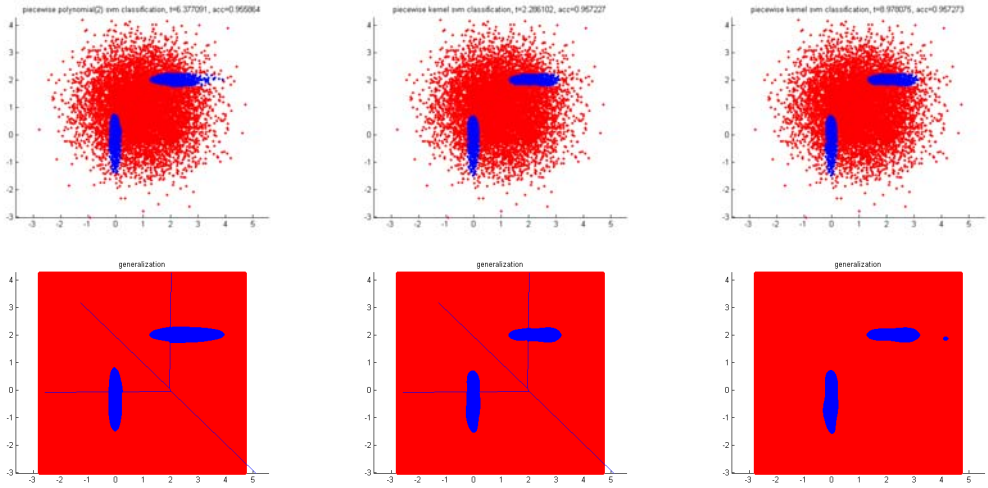


Figure 3.6. The classification results(top) and the generalization(bottom) of the different classifiers on a synthetic data with 10 times the training samples compared to Figure 3.6 and with the same parameters.

explicitly mapping the feature space to higher dimensions will help the performance of such classifiers. A suitable approach for this purpose is to augment the feature space with additional non-linear combination of its dimensions which is (implicitly) done in the popular kernel mappings.

Following a similar idea, a few additional features can be chosen greedily from a feature pool using a measure of performance such as accuracy or more sophisticated measures such as the one introduced in Section 3.5. The feature pool can be chosen to be the geometric mean of two or more features, which on feature vectors of dimension n, leads to a feature pool of size $\sum_{i=1}^{n} \binom{n}{i}$.

3.5 Quantitative assessment of the boundary detection

In order to be able to quantitatively assess the boundary detection algorithms, a criterion needs to be defined. The simplest measure for this purpose is the mean absolute error:

$$D_H = \frac{1}{N} \sum_{x} |\rho(x) - \rho^*(x)|$$

The measure performs well in case of perfect localization of the detection. However, D_H is sensitive to displacements and performs poorly if any small displacement is present in the detections. Alternatively, a criterion is proposed as:

$$D(\rho, \rho^*) = \frac{1}{N} \sum_{x} \left(\underbrace{\min_{t} \; |\rho(x) - \rho^*(t) e^{-\frac{||t-x||^2}{2\sigma^2}}|}_{P(x)} + \underbrace{\min_{t} \; |\rho^*(x) - \rho(t) e^{-\frac{||t-x||^2}{2\sigma^2}}|}_{R(x)} \right) \tag{3.5}$$

in which $0 \leq \rho \leq 1$ is the estimated boundary, $0 \leq \rho^* \leq 1$ is the ground truth and N is the number of pixels in the image. The measured criterion is useful to assess the results of any edge detection algorithm(where the ground truth information is available). Clearly, D takes into account correct/incorrect detection/rejections(and therefore, $D(0, \rho^*) > 0$ and gets a better score than a completely wrong segmentation) and penalizes imperfect detections(using σ) instead of immediately labeling the detections as false positives as soon as they are imperfectly localized(for example by 1 pixel). In other words, D is a soft approximation of D_H (it will produce a reasonable score for an exact detection of edges shifted by one pixel while the hard version penalizes such detections seriously). The proposed D has non-negativity, identity and symmetry characteristics of a metric while its triangle inequality characteristic is to be investigated. It is worth noting that D can deal with the binary/probabilistic detections and binary/probabilistic ground truth information.

$P(x)$ and $R(x)$ in (3.5) are measures of precision and recall of the detections i.e. P measures how precise the detections are while R measures the recall of the detections. With a reasonable σ, one can overcome the limitations of the motion based cues(imperfect localization). Figure 3.7 reflects the exponential term of the proposed measure. According to the figure, detection with $\rho(x_0) = 1$ having $t(x_0) =$

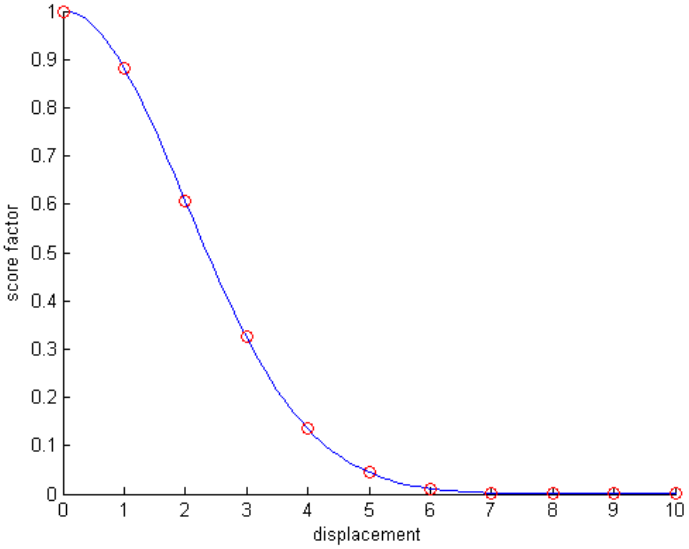


Figure 3.7. Penalization vs Displacement for $\sigma = 2$.

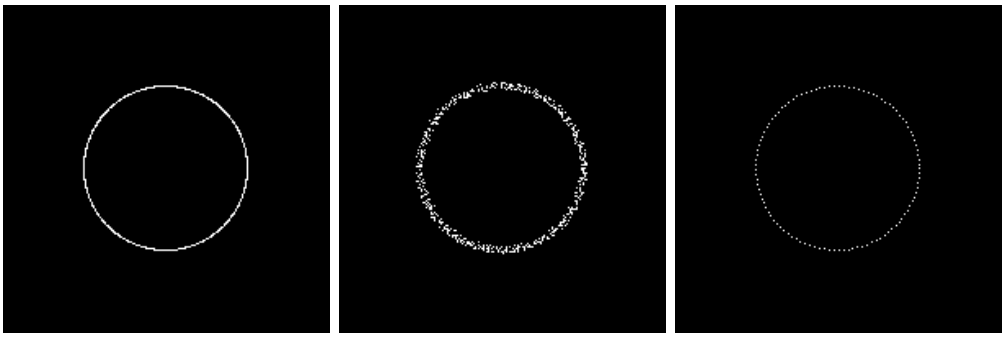


Figure 3.8. Synthetic ground truth and images and scores calculated for each case. (left) ground truth (middle) noisy detection with on average [1 1] displacement (right) partial perfect detections.

$[11]^T$ as the minimizer of (3.5)(a 1 pixel shift in x direction and 1 pixel shift in y direction) leads to 0.5 penalty(1-0.5) instead of 1. Figure 3.8 shows two test cases for which the scores using the hard accuracy(D_H) and the soft approximate are calculated using the same γ. Scores calculated using D_H are 0.25 and 0.53 respectively while using the soft approximations(D), the calculated scores are 0.82 and 0.92.

In order to scale D to [0,1], a simple scaling such as $Q(D) = e^{-\gamma D}$ can be utilized. For the results in this work, the parameters were chosen to be $\gamma = 100$ and $\sigma = 2$.

3.6 Summary

Using the concept of localized classifiers, the training and the testing phase of the occlusion boundary detection are summarized in Algorithms 2 and 3.

Algorithm 2 Training the occlusion boundary detector using localized classifiers

Inputs: N sequences of images: $I_{t,n}$, $n = 0, ..., N$ and $t = 0, 1$ with the corresponding ground truth information ρ_n^*, k the number of cluster centers
Compute backward flow u_n between $I_{0,n}$ and $I_{1,n}$
Compute the gPb detector gPb_n from $I_{1,n}$
Form the desire feature vector $F_{D\times N}$ for each pixel from gPb_n and u_n and form the target labels $L_{1\times N}$ using ρ_n^*
Cluster feature responses of positive samples to k clusters c_i, $i = 1, .., k$ using k-means clustering
for $i = 1$ to k **do**
  Find all the feature vectors F_i that associate to c_i and their corresponding labels L_i
  train a classifier $f_i(.)$ using F_i and L_i
end for
return The localized classifier set f_i and c_i

Algorithm 3 Using the occlusion boundary detector

Inputs: N sequences of images: $I_{t,n}$, $n = 0, ..., N$ and $t = 0, 1$ and the localized classifier set f_i and c_i
Compute backward flow u_n between $I_{0,n}$ and $I_{1,n}$
Compute the gPb detector gPb_n from $I_{1,n}$
Form the desire feature vector $F_{D\times N}$ for each pixel from gPb_n and u_n
for $i = 1$ to k **do**
  Find all the feature vectors F_i that associate to c_i
  Classify F_i: $Y_i = f_i(F_i)$
end for
Re-arrange Y_i in Y_n so that Y_n has the same order as the input sequences
return Y_n

Chapter 4

Evaluation

4.1 Data Set

The data set used for the intermediate boundary detection and variational segmentation consists of 25 sequences of 2 images in various conditions e.g. indoor/outdoor, textured object/background, different lighting conditions and etc. In the sequences, the camera has undergone approximately a translative motion and the motion vector associated with pixels varies from a few pixels up to 42 pixels in magnitude. A total of 8 sequences were picked from the data set presented in [44] where the F/B segmentation was applicable[1]. Figure 4.1 depicts the second frames of a few sequences of the data set and Figure 4.2 depicts the corresponding defined ground truth edges. Figure 4.3 depicts the backward motion computed between the two frames of the sequences.

4.2 Intermediate Boundaries

4.2.1 Features

A few cues for intermediate boundary detection were mentioned in Section 3.1. As it was pointed out in Section 2.7, each type of regularization produces different results in optical flow each one having their own advantages and drawbacks. In order to utilize the information in both types of methods, three methods with homogeneous flow based and isotropic and anisotropic image based regularization were used to compute the mentioned cues[2]. The cues computed from the 3 flows computed

[1]Note that the ground truth information is modified in most of the picked sequences as the ground truth edges defined originally were badly localized in some cases.

[2]It is worth noting that information from the three mentioned regularizers are complementary and thus produces better results if combined together. Of course if one uses more sophisticated methods such as the ones mentioned in Sec 2.7, s/he does not need to compute 3 different flows and do this redundant step. Unfortunately, at the time of development of this work, such methods were not available at hand and as the focus was not on implementation of a sophisticated optical flow algorithm, a cheaper alternative was implemented(combining the results of multiple optical

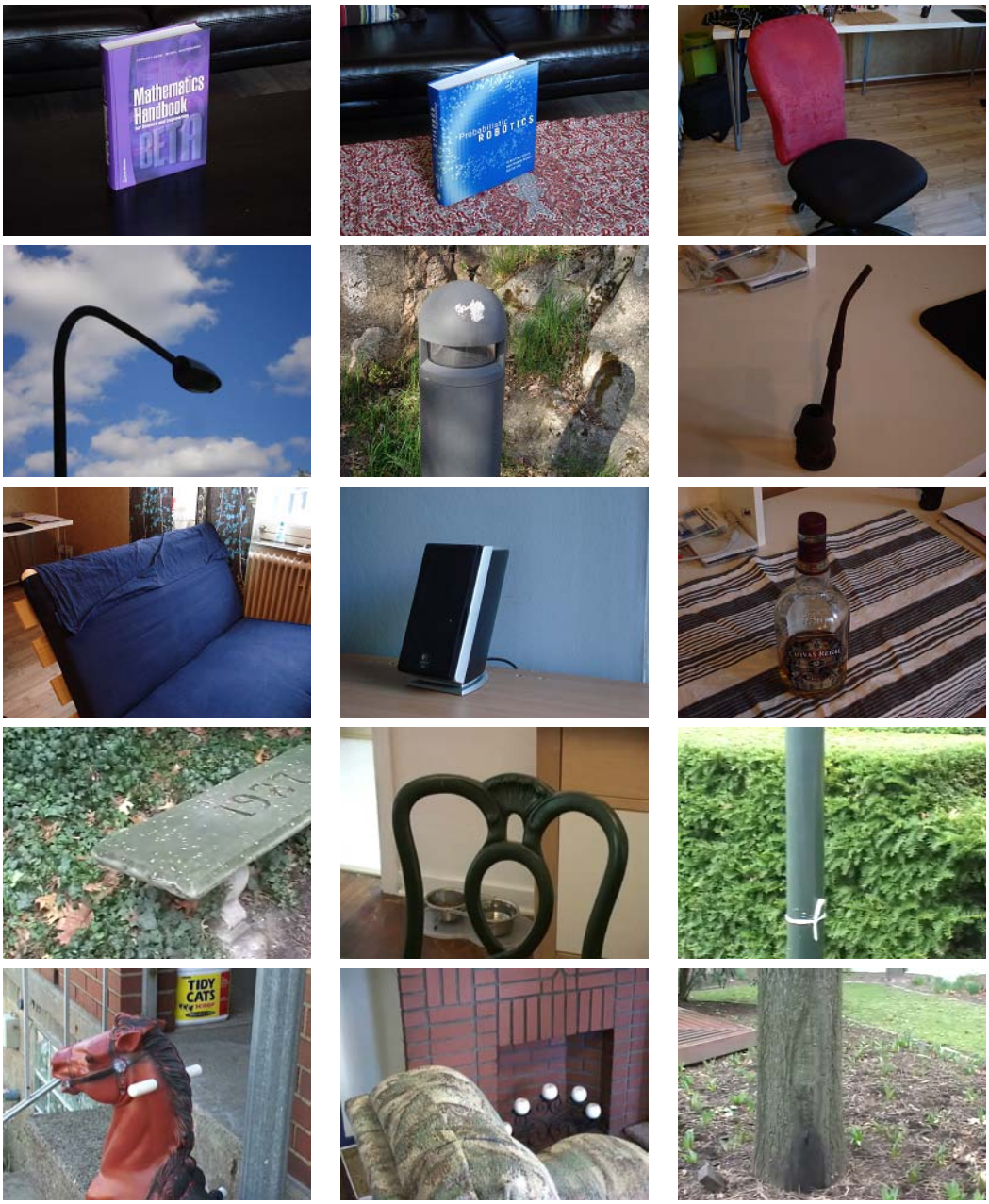


Figure 4.1. The second frame of a few sequences of the data set used for training/evaluation of the methods. The last 6 sequences were picked from the data set presented in [44]

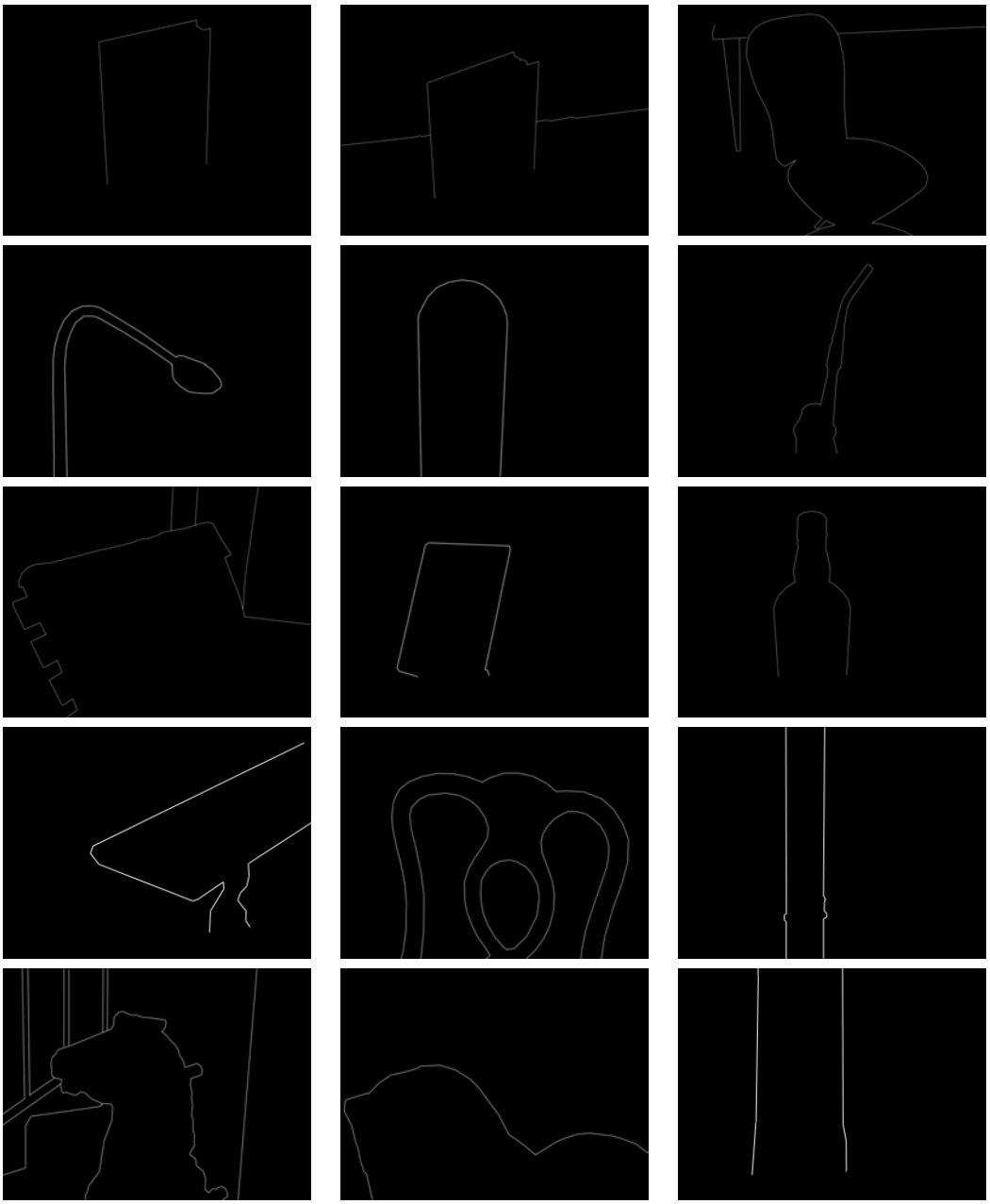


Figure 4.2. The ground truth edges of a few sequences localized on the second frame. The width of the labeled edges is 1 pixel(the images are smoothed for the demonstration purpose).

Figure 4.3. The backward motion computed for a few sequences.

by [7](homogeneous flow based regularization with color channel utilization) and [61](image based regularization on intensity channel) and two new cues were computed using the algebraic mean and the geometric mean of the 3 computed cues. The feature computed using the geometric mean is more sensitive to disagreements between the cues than the feature computed from the algebraic mean of the cues. Therefore, 10 motion based features (two per each of 5 cues mentioned in Section 3.1) in addition to the gPb cue(in its thick and thin forms) were used to form a 12 dimensional feature vector for each pixel.

Figure 4.4 depicts the cues mentioned in on a few sequences. It can be observed that the warping error cue is very noisy and contains much less information compared to the other cues. As warping in general, changes the noise distribution in the images(mainly because of the bi-linear interpolation) [21], it can be expected that the warped images have some error compared to the original image as soon as the flow vector components are not integers. For the same reason and also as the warping error is the measure which is to be minimized in the functional of the variational optic flow estimation methods, the warping error cue is not integrated in the feature vector.

Afterwards, to build richer feature vectors, 4 of the best features(as it was motivated in Section 3.4) were selected from the set of geometric means of all possible combinations of features ($2\sum_{k=2}^{10} C(n,k) = 2026$ features) and augmented them to our feature vectors. Table 4.1 shows the measured score for the original(the 8) features as well as the selected features and Figure 4.5 depicts the new features.

Surprisingly, both the thin and thick versions of the gPb detector appeared in the selected features. Furthermore, the selected features were mostly the geometric means of an appearance based features and motion based features. As the geometric mean between two(or more) values scales down very fast when the two values completely disagree and also grows faster than the algebraic mean if the two values somewhat disagree, we can interpret such features as playing the role of removing the false detections of the gPb detector and motion based cues (with respect to the defined ground truth information) while keeping most of the information each feature is not certain about. These facts can also be verified visually from Figure 4.5.

Subsequently, in order to integrate information from neighboring areas into each feature vector, smoothed versions(gaussian smoothing with $\sigma = 2$) of all features were again augmented to the feature vector increasing its dimension to 18[3]. Figure 4.6 represents the final feature vector.

flow algorithms)

[3]For the combinatory features, which were computed from the geometric mean of multiple features, the geometric mean was not smoothed, but it was computed from the corresponding smoothed features.

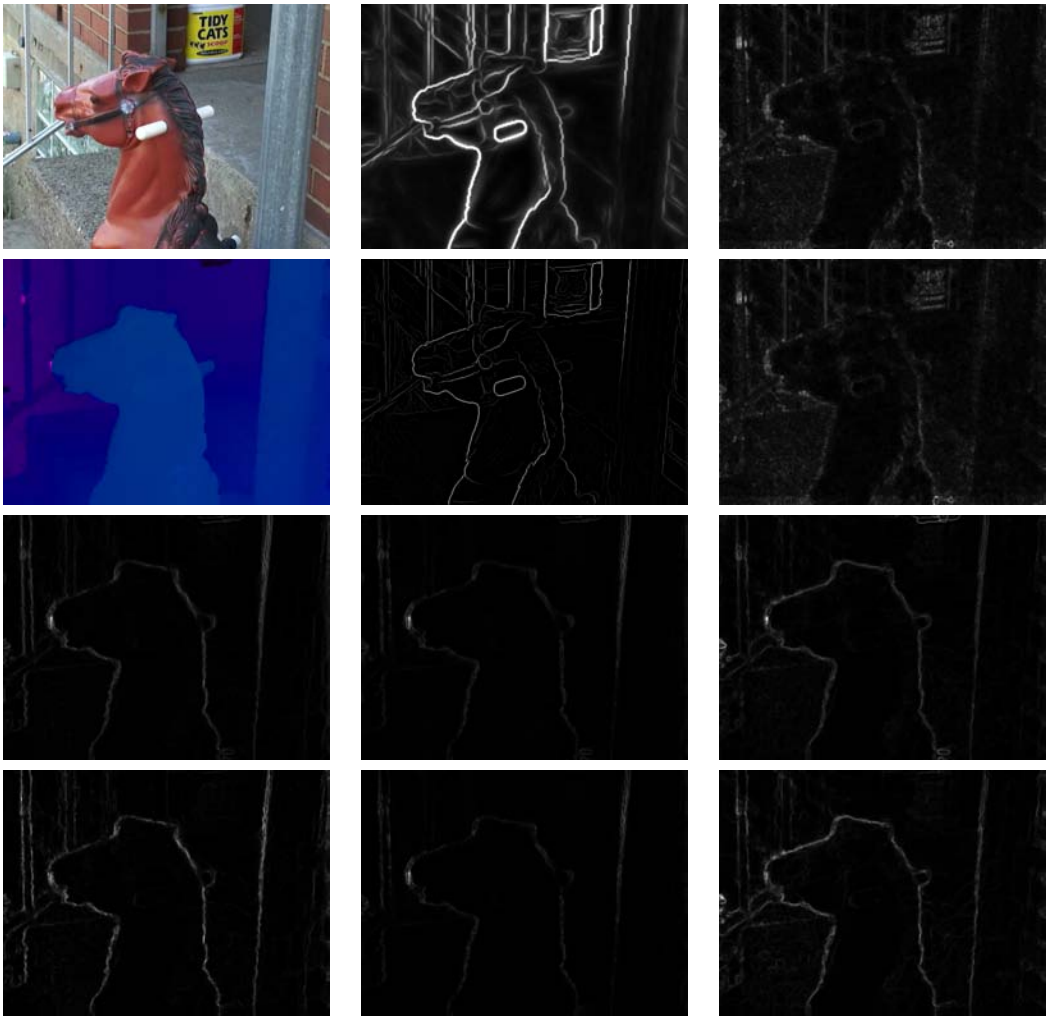


Figure 4.4. Cues for intermediate boundaries: (top) the original image, gPb response, algebraic mean of (contrast invariant)warping errors, (second top) backward flow, thinned gPb response, geometric mean of (contrast invariant)warping errors, (second bottom) and (bottom) are the algebraic means and geometric means of divergence of the flow field, incompatibility of the forward and backward flow fields and gradient magnitude of the norm of the flow field respectively.

Table 4.1. The augmented features

Feature Num	Description	100* Score
0	Empty Detection	51.54
1	gPb	17.48
2	gPb thin	49.02
3	Div aMean	48.63
4	Div gMean	55.46
5	Inc aMean	45.75
6	Inc gMean	51.21
7	GMN aMean	53.63
8	GMN gMean	**58.22**
1 2 7	comb	**60.44**
1 2 8	comb	60.37
1 8	comb	58.81
8	GMN gMean	58.22
1 7 8	comb	57.85
1 7	comb	57.79
7 8	comb	57.68
2 8	comb	57.44
1 2 3	comb	57.32

4.2.2 Classifier

The choice of radial basis kernel SVMs as the classifiers was motivated in Section 3.2 and alternatively, the concept of the localized polynomial degree 2 kernel SVM classifiers was introduced and motivated in Section 3.3. Having split the available sequences to (half) training - (half) test sets[4], the details of the train / test phase is elaborated as follows. As training a radial basis kernel SVM with cross validations on very large data-sets such as the current one (order of 2 million training / test sample feature vectors) is a very time consuming task(training each [global] SVM with cross validation takes approximately one day on a 3 GHZ Intel Quad Core with 6 GB of RAM and therefore, searching on a grid for the optimal parameters(c and γ) with a grid of size $15{=}3\times5$ takes around two weeks. The test time of such classifiers is around 1 hour on the test set.), concepts such as localized classifiers come into attention(training a localized classifier set with the mentioned parameters on 25 partitions takes around 20 minutes on the same computer and the test time is around 4 minutes). Furthermore, exploiting the distribution of the negative samples(as in

[4]Note that the proposed measure in Section 3.5 utilizes the spatial information of detections / ground truth and therefore, randomly splitting the feature vectors to two sequences leads to two drawbacks: 1- In the testing phase, the correlation and dependency of the neighboring feature vectors prevents a fair evaluation of the method and 2- To have a consistent spatial order, training feature vectors need to be integrated in the test set making the comparisons inaccurate. Therefore, the sequences instead of feature vectors are split to train/test sets.

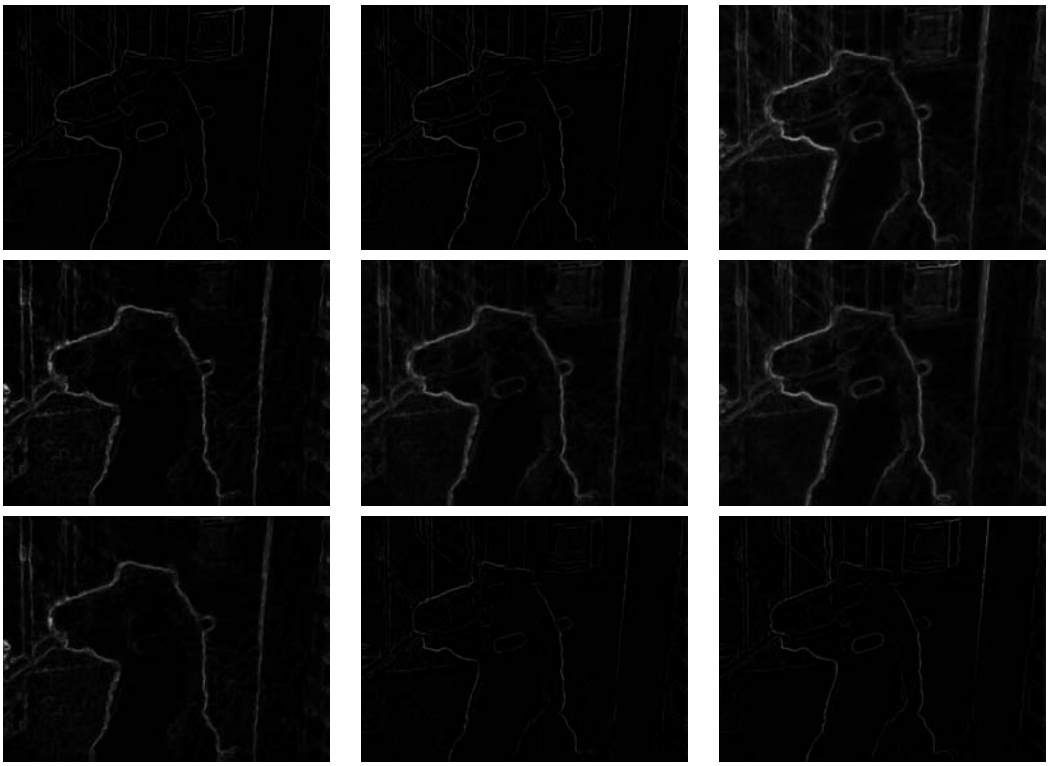


Figure 4.5. Selected features from the feature pool according to Table 4.1 on one of the sequences.

this problem, most of the negative samples consist of feature responses with very low magnitude, more than 90 percent of the negative data ends up in one cluster), it is possible to train even simpler and faster classifiers such as linear SVMs on the partitions with very imbalanced distribution of negative and positive samples. Utilizing a similar approach, the training time can be reduced to 4-5 minutes and the testing time to 2-3 minutes.

Figures 4.7 and 4.8 show the results of the best performing radial basis kernel SVM and localized classifiers set on the test data. It can be observed that the results of the two methods are comparable, while the localized classifier set results to noisier detections. However, because of the thicker detections of the radial basis kernel SVM(the ground truth information has a width of one pixel and thicker detections by the definition contain false positives), the overall score of the localized classifier set's results is superior. Considering the huge differences in the test time(45 minutes vs 22 seconds), one might be interested in such tradeoffs between precision and the computational requirements.

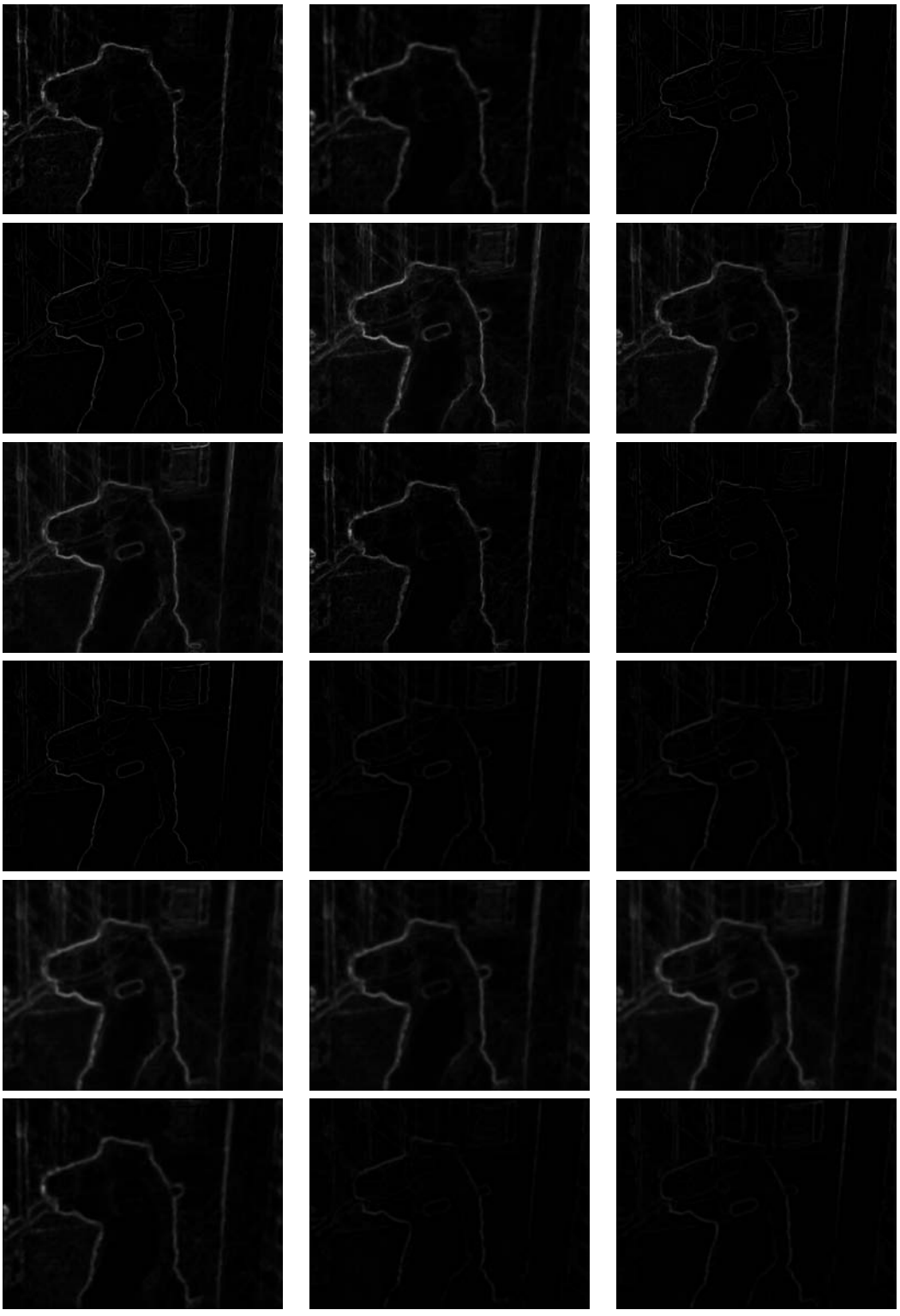


Figure 4.6. The final feature vector on one of the sequences.

Figure 4.7. The results of the best radial basis kernel SVM(chosen using the cross validation score from 9 trained classifiers) on the test set. The average score on all sequences is 71.15. The test time is 45 minutes.

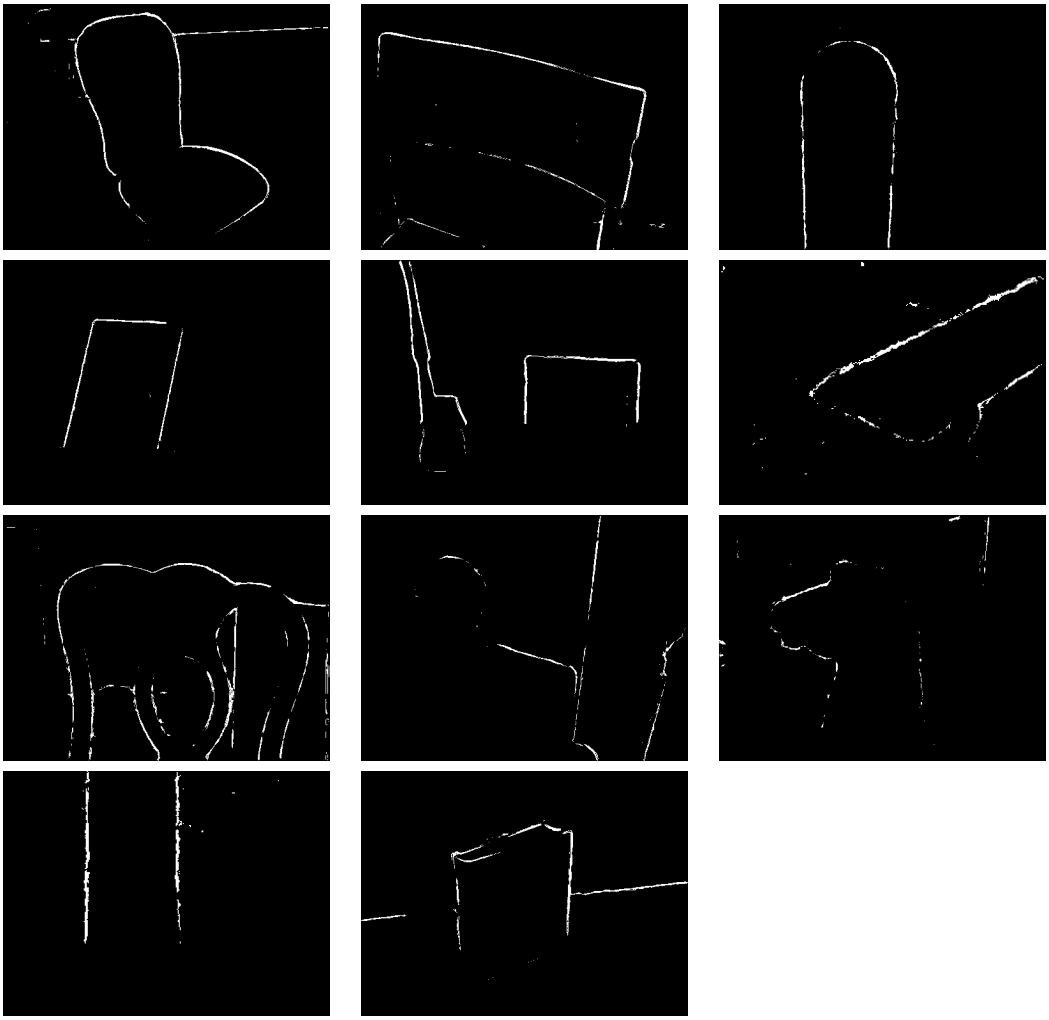
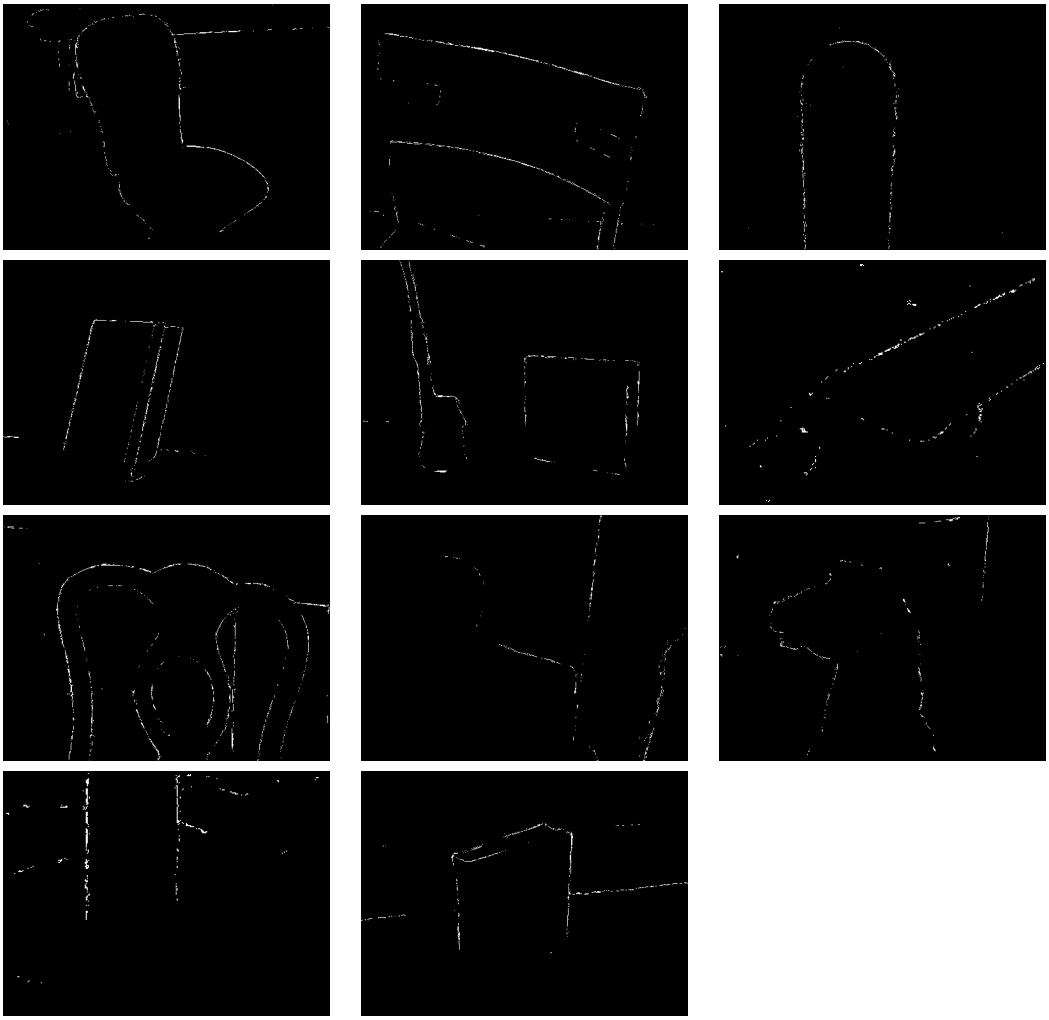


Figure 4.8. The results of the best localized classifier set(chosen using the cross validation score from 18 trained classifiers) on the test set. The average score on all sequences is 78.18. The test time is 22 seconds.



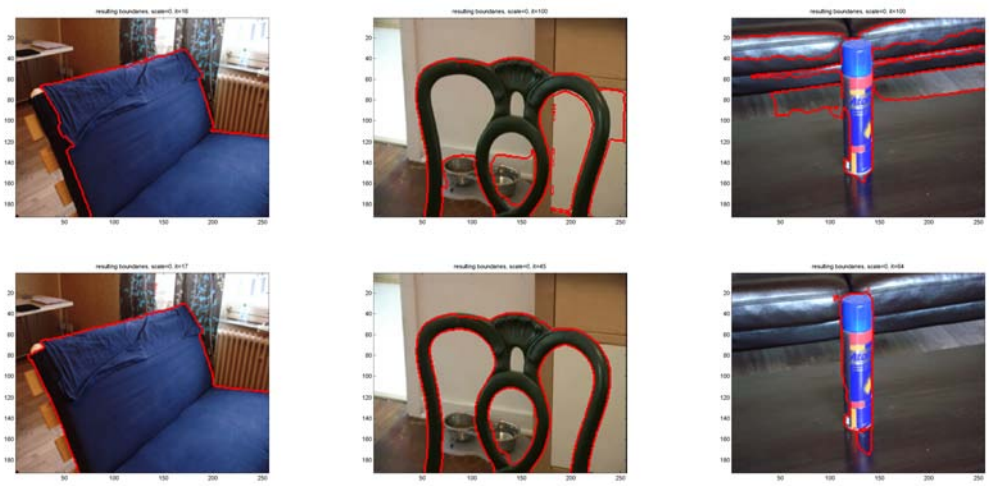


Figure 4.9. The resulting segmentations using (top) $\sigma = 8$ and (bottom) $\sigma = 16$.

4.3 Variational Segmentation

In this section, the results of segmentation using the functional (2.18) are presented. The segmentations are performed using a coarse to fine processing strategy starting with images resized to have maximum of width and height of 128 and then a finer image with double the size of the starting image(a 2-scale strategy). On the coarser level, the level set is evolved until the resulting segments get fixed or a maximum of 250 iterations is reached and on the next level, the maximum iteration number is fixed to 100.

4.3.1 Likelihood Estimation

Assuming independent feature channels(Section 2.4), the 1-D KDE is performed using grids with 256 bins covering [0,255]. Although it is possible(and preferable) to tune the kernel bandwidth(σ) for each channel separately, in this work the same σ is considered for all channels. In general, in situations where the (un-smoothed) pdfs have peaks in quite different areas, it is preferable to use larger bandwidths. On the other hand, in situations where the feature distribution of the foreground and background regions are somewhat similar, smaller σs will be able to discriminate between the segments better. Perhaps an adaptive kernel bandwidth can be defined using a measure of similarity of the evolving segments e.g. the χ^2 distance between the pdfs using a pre-defined kernel bandwidth. However, such an approach is not implemented in this work and instead a few kernel bandwidths were evaluated visually and $\sigma = 16$ was found to perform well on most of the sequences in overall. Figure 4.9 depicts the resulting segments using different kernel bandwidths.

4.3.2 Features

The three described features: color, texture and motion are integrated in the energy functional (2.18). The threshold on the smoothness measure of the texture introduced in Section 2.5 was chosen to be $\theta_t = 20$ for all scales. This leads to the texture feature not getting smoothed considerably and instead, the smoothing is postponed to the coupled diffusion of the entire feature vector. Prior to the isotropic coupled non-linear diffusion, all feature channels are re-scaled so that they have the dynamic range in [0,255]. Afterwards, the feature vector is smoothed until the smoothness measure reaches the threshold θ_f. θ_f is varied linearly with the scale of the image. The value of θ_f and the kernel bandwidth of KDE play similar roles in the outcome of the segmentation process: with smaller θ_fs, the feature vector gets smoother and with the same bandwidth, leads to less local minima in the pdfs. With larger bandwidths, the same effect is achieved provided that θ_f is small enough. Figure 4.10 depicts the resulting segmentations in case of two different thresholds of the mentioned measure. It is observable that with smaller θs, the resulting solution is smoother. Therefore, θ is chosen as 2.5 and 3 for the two mentioned scales in the segmentation process.

The motion feature is computed using a variational method with anisotropic image based regularizer [61] without any post processing. In case of independent pdf estimation, the dependency of the x and y components of the optical flow can be reflected in the pdf using an extra channel reflecting the correlation between channels. The norm of the optical flow vector can reflect such dependency and also, can be used as a measure of inverse depth of the 3D points associated with each pixel in the image. Figure 4.12 depicts the resulting segmentations using different uses of the motion information. It can be observed that in general, utilizing the motion information helps finding the correct objects while adding the extra channel(the norm of the motion vector) to the motion vector improves the results. In order to be able to estimate informative motion, as it was mentioned in Section 2.7, the sequences in general need to contain enough gradient so that the regularization does not produce discontinuities in the flow field, which do not correspond to depth discontinuities. Furthermore, if the main assumptions of optical flow (e.g. Lambertian objects) are violated by e.g. strong shadows or non-Lambertian surfaces, regularization will prevent the violation of piecewise smoothness assumption of the resulting flow to necessarily correspond to depth discontinuities in the images yet again. Figure 4.13 depicts some examples of such situations. It is observable from the figure that the method is not too sensitive to bad estimates of flow if the appearance is discriminative between foreground and background situations, if neither the flow is estimated correctly and nor the appearance is discriminative enough, the method will fail to produce reasonable results. Notice that the black chair can be distinguished from the background, but in the pipe sequence, neither the object nor the background represent distinguishable textures or color distribution to compensate for the incorrectly estimated flow.

In order to evaluate the effect of each feature in the segmentation process, the

Figure 4.10. The resulting segmentations using different thresholds on the measure of smoothness of the feature space for two sample sequences: (top) the thresholds are {2.5,3} on the two scales and (bottom) the thresholds are {3,4}. The rest of the parameters are exactly the same.

resulting segmentations using each feature separately on 3 sequences are depicted in Figure 4.11. It can be observed that, as it should be expected, none of the features are able to produce reasonable results in all cases and it is also observable that each feature provides unique information. However, it will be shown later that integrating the features in a proper way, can lead to superior results(compare with Figure 4.12). The situations in which each type of feature provides useful information and the limitations of usage of each feature is described in the following.

In order for the motion information to be useful for the segmentation process, the object needs to show enough parallax in the sequences. However, at the contact points(or contact surfaces) of objects, this requirement is not satisfied and neither there is a depth discontinuity at the contact points. Therefore, on such regions some other cue needs to help distinguish the correct boundaries. Hence, if the appearance based information is not discriminative enough on the contact points of objects, larger bandwidths weaken the correct boundaries leading to different segments getting merged to each other. Such a situation is depicted in Figure 4.14. It can be observed that with decreasing the kernel bandwidth and emphasizing

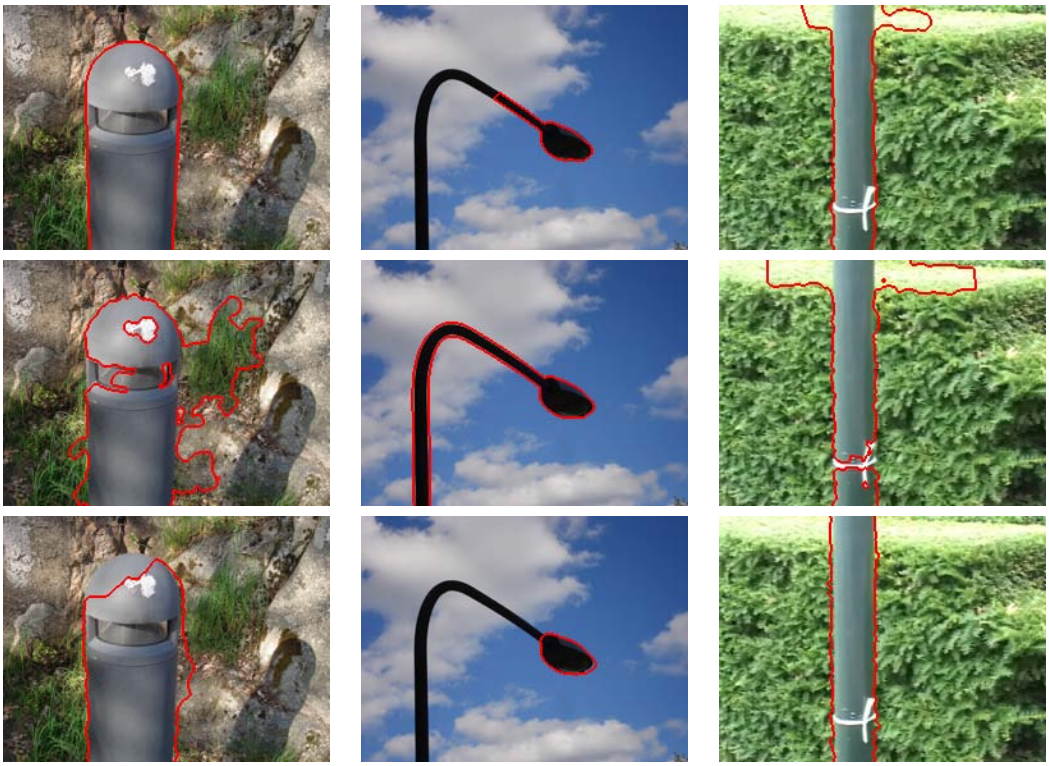


Figure 4.11. The resulting segmentations using: (top) color feature, (second top) texture feature and (bottom) motion feature with the same parameters on 3 sequences.

less on the motion cue(using the 2-channel motion instead of the 3 channels), it is possible to make the algorithm more sensitive to appearance changes and thus finding the correct boundaries for the tree and the bench sequences(second bottom row). Although for the juice sequence, the mentioned strategy adaptation improves the segmentation slightly, the lack of gradient and discriminative color and texture features leads to the failure of the algorithm to segment out the object. In such sequences, different types of knowledge / prior needs to be integrated in the algorithm.

4.3.3 Initialization and Sensitivity to assumptions

As it was mentioned earlier, the final results do not depend on the initialization of the level set if the energy functional has only one local minimum. Although that requirement is not satisfied in practice, using feature smoothing and kernel bandwidth adaptation, the number of minima are significantly reduced. Figures 4.15 and 4.16 depict the resulting segmentation using the default initialization and a manual initialization in cases that the choice of initialization led to different results(for the

Figure 4.12. The resulting segmentations using: (top) no motion information: color and texture features only(8 dimensions), (second top) motion norm (8+1 dimension), (second bottom) motion vector(8+2 dimensions) and (bottom) motion vector + motion norm (8+3 dimensions) on 3 sequences.

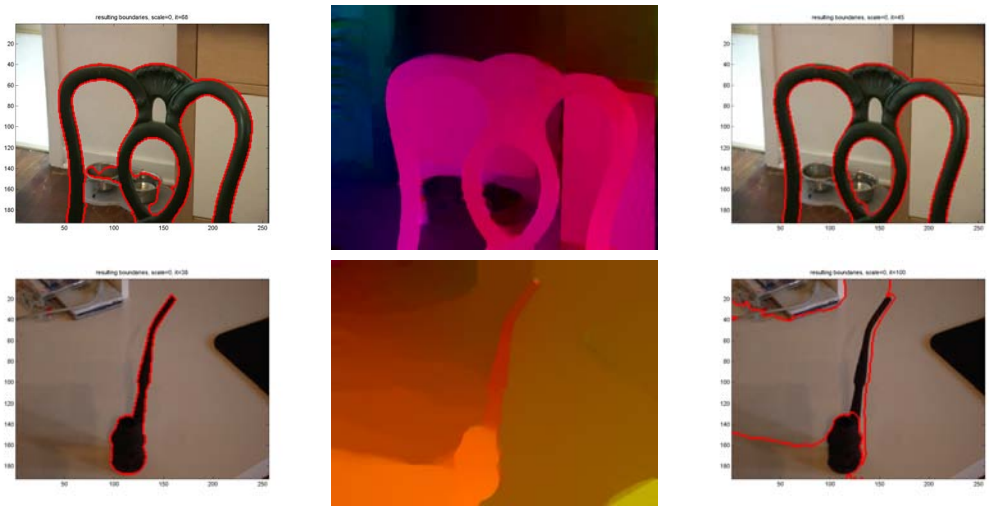


Figure 4.13. The resulting segmentations using: (left) no motion information, (middle) calculated flow, (right) motion vector + motion norm on 2 sequences. The method can use the appearance cue to find the correct boundaries where motion information is not accurate(top), but if the appearance cues are not discriminative and the motion is not estimated correctly, the method will fail to segment out the object(bottom)

rest of the sequences, the choice of the initialization did not matter). It can be observed that with proper initialization, the algorithm can compensate for violation of some of the assumptions of the method. For the chair sequence, the chair does not show enough parallax (it is comparatively close to the wall behind it), the motion of the stiles is merged with the motion of the background(because the blue stile connecting the chair back and the seat does not have enough gradient), the appearance of its connecting parts is very similar to the wall on the left(they are both blue and do not show different textures) and therefore, the method can not find the desired segmentation[5]. For the street light sequence, because of the lack of strong gradient in the image, the motion of the upper part of the light is merged with the motion of the background and also the motion of the blue sky is estimated differently than the motion of the white clouds(see Figure 4.3). Also, the default initialization(containing mainly the blue sky and the street light) leads to some parts of the blue sky to be merged in the foreground segmentation. For the pipe-box sequence, there are two objects in the image violating the one object assumption of the method and leading to only the box getting segmented out. However, with a manual initialization containing both objects, both objects can get segmented out. Similar facts can be found out in Figure 4.16: the red background of the couch sequence can be segmented out while the white part of the background getting merged to the foreground region. The presence of the shadows disturbing the appearance(and

[5]However, with manual initialization, the seat of the chair can be segmented out.

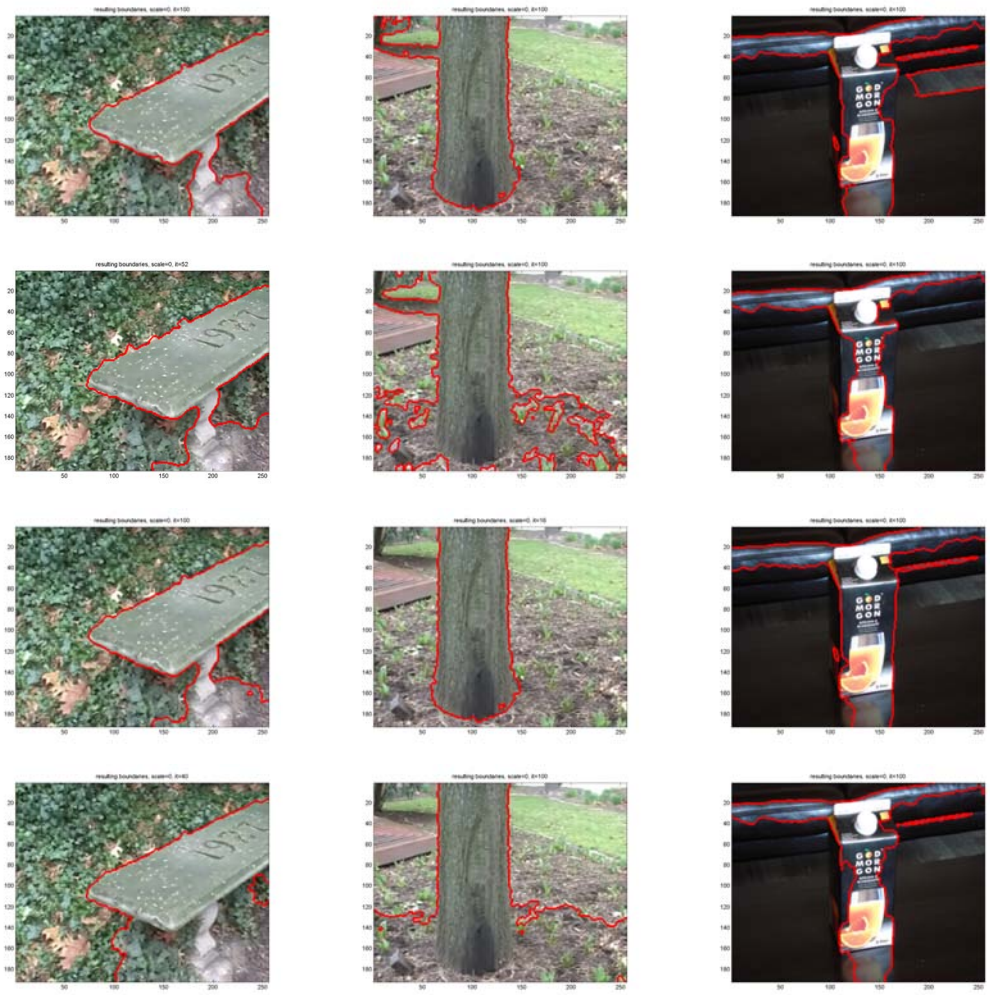


Figure 4.14. The resulting segmentations using: (top) kernel bandwidth $\sigma = 8$ and no motion, (second top) $\sigma = 16$ and no motion, (second bottom) $\sigma = 8$ and motion vector, (bottom) $\sigma = 16$ and motion vector + motion norm on 3 sequences.

Figure 4.15. The resulting segmentations using $\sigma = 16$ and different initializations: (top) the default initialization, (second top) the result of the default initialization, (second bottom) manual initialization with the initialization covering a part of the object only and (bottom) the result of the manual initialization.

the motion of) the background/foreground regions in the middle sequence and in addition to the lack of parallax and presence of the 3 foreground objects, prevents the segmentation algorithm to converge to the desired segmentation. Again, with proper initialization, 2 out of 3 objects can be segmented out. Similarly, for the last sequence in the same figure, a proper initialization compensates for violations of the assumptions and leads to the frontal brick wall getting segmented out.

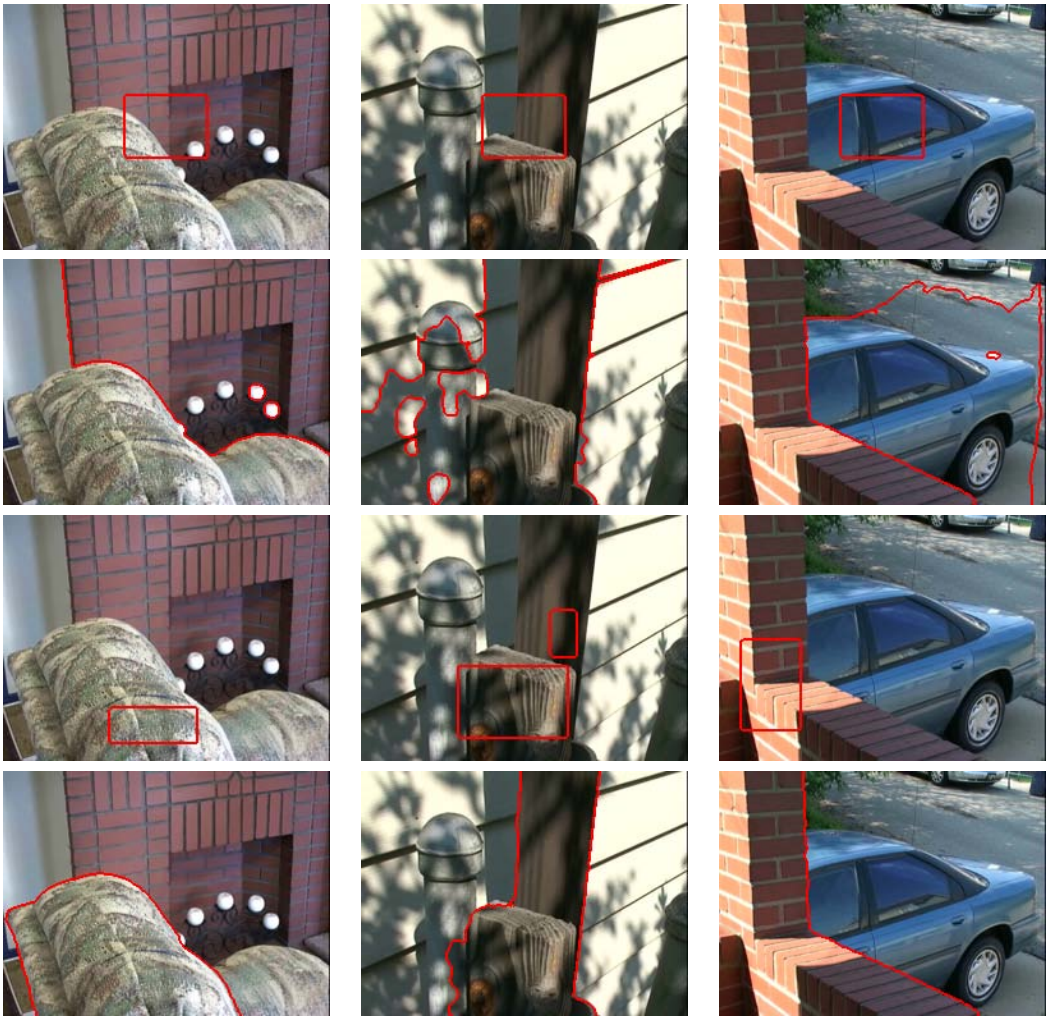


Figure 4.16. The resulting segmentations using $\sigma = 16$ and different initializations: (top) the default initialization, (second top) the result of the default initialization, (second bottom) manual initialization with the initialization covering a part of the object only and (bottom) the result of the manual initialization.

4.3.4 Curvature Motion

MCM is performed using ν parameter adapted to the size of the image as it was suggested in [6]: $\nu = 10^3 N^{0.7}$, where N is the number of pixels in the image and the parameter τ is fixed to 0.25 for stability purposes. The mentioned ν was found to perform well in most of the cases and therefore it was not fine-tuned furthermore. It is worth noting that the value of the ν parameter depends on the dynamic range of the likelihoods being estimated and the re-scaling and the regularizing value being used.

Similarly, the magnitude of the GAC energy depends on the dynamic range of the likelihoods of the features. Poorly regularized likelihoods(likelihoods with very large dynamic range) will cause the effect of the MCM and GAC terms to be negligible and likelihoods with very narrow dynamic range lead to the domination of the MCM and/or GAC terms in the evolution equation. On the other hand, scaling down the likelihoods of the features, which are the main force evolving the level set leads to very slow convergence of the evolution equation. Therefore, in order to be able to use GAC successfully with the mentioned functional, another tuning step and perhaps an adaptive weight for the GAC energy needs to be considered. This was not done in this thesis and therefore, it was found out empirically that for small(and stable) weights of the GAC energy, MCM and GAC produced very similar results in case of reasonably large magnitude of the likelihoods[6]. Therefore, the GAC energy in form of the functional (2.10) was not found to lead to a considerable improvement over the MCM energy (2.8) using the mentioned features and/or parameters.

[6] A reasonable magnitude of the likelihood function was empirically set to be bounded in [-100,100] so that the evolving set converges to a minimum in less than 250 iterations on the first scale.

Chapter 5

Summary and Conclusion

5.1 Summary

In this thesis, the problem of object segmentation using spatial(e.g. color and texture) and spatio-temporal cues(e.g. apparent motion) was investigated. The foreground / background object segmentation problem was introduced in Chapter 1 and some possible approaches to the mentioned problem and their advantages and shortcomings were discussed in the same chapter.

In Chapter 2, the variational segmentation framework was discussed and elaborated and the multi-cue version of the cartoon-limit of the Mumford-Shah functional using the level sets were introduced and it was shown to be equivalent to maximizing the a-posteriori of the image partitioning, provided that the underlying assumptions of the method(e.g. the random process giving rise to the feature vectors being i.i.d and the prior term for the partitioning being a member of the exponential family) hold. MCM and GAC were introduced in the same chapter and the use of GAC for the purpose of encouraging the evolving contour of the level set to some pre-defined areas(e.g. image edges or the intermediate boundaries which were introduced later on) was elaborated. Also, possible ways to estimate the pdfs required for estimation of the likelihoods of features belonging to each region were discussed and the chosen approach was the 1-D KDE and the independency assumption of the feature channels for the sake of lowering the computational costs of the segmentation algorithm. The concept of weighting as well as the functional reflecting the assumptions were discussed at the end of the same chapter.

In Chapter 3, a method was introduced to integrate the appearance based and motion based cues in a classification / regression framework to detect occlusion boundaries in image sequences. Some possible motion based features and the use of relevant classification / regression methods were discussed in the same chapter. A measure for assessing the quality of the detections was proposed and motivated, which can cope with imperfect localization of the detected edges in sequences. A possible way to make richer features according to the same measure based on the geometric means of multiple cues were introduced and finally, the concept of local-

ized classifiers were motivated and elaborated in the same chapter. It was shown that the localized classifiers can be used for huge speed-ups in the training and the test time with the cost of introducing a slight bias towards the training data.

In Chapter 4, the data set for the evaluations was proposed and introduced, which consists of a few sequences from well known data-sets as well as some new sequences for the purpose of F/B object segmentation or occlusion boundary detection. The proposed methods were evaluated and the qualitative and quantitative results of the methods were presented in the same chapter as well as a thorough discussion of advantages and limitations of the proposed methods.

5.2 Future Works

At the end of this work, some issues still remained to be investigated in future works. A few interesting areas for future works are as the following:

- At the beginning of this work, the occlusion boundary detector(the intermediate boundary detector) was developed for the purpose of integration in the variational segmentation framework. The idea was to utilize the GAC energy to encourage the level sets to converge to the detected areas. However, as it was pointed out in Section 4.3.4, that approach did not work out as expected. Perhaps an adaptive weighting of the second term of the GAC energy as it was pointed out in Sections 2.3 and 4.3.4 can be useful for that purpose.

- While the concept of localized classifiers offer huge speedups, as it was pointed out in Section 3.3, they introduce additional bias towards the training set. Investigating systematic approaches to decrease this bias can be interesting and if the bias can be eliminated by approaches such as the ones mentioned in the same Section, the concept can find its use in many other similar problems.

- Developing systematic approaches to solve many issues of the variational segmentation method such as parameter tuning, feature weighting, feature smoothing, kernel bandwidth tuning and likelihood regularization can be very beneficial for the robustness of the algorithm. A few approaches were suggested in Chapter 2, but were not evaluated. Evaluation of such ideas on larger data-sets is worth considering for future works.

- Generalizing the F/B object segmentation to arbitrary number of objects can be very interesting(and more realistic).

5.3 Conclusions

In this work, the use of the apparent motion feature in addition to the appearance based features in a variational segmentation and in an occlusion boundary detection framework were investigated in case of translative camera(observer) motion. Using

the state of the art motion estimation algorithms, it was shown that the apparent motion can be successfully integrated in such frameworks to get better results, provided that the sequence does not violate the assumptions of the optical flow methods(e.g. Lambertian object) and also, the sequences contain enough gradient so that the optical flow can be computed correctly and the regularization does not introduce artificial discontinuities in the flow field. The contributions of this work were as the following:

- Investigation of integrating the [apparent] motion in the variational segmentation problem.
- Proposing a classification approach to occlusion boundary detection.
- Proposing a reliable measure of goodness of the edge detection algorithms.
- Preparing a small data-set of 2-image sequences for the purpose of F/B object segmentation or occlusion boundary detection.

Bibliography

[1] R. Adams and L. Bischof. Seeded region growing. *Pattern Analysis and Machine Intelligence*, 16(6):641–647, June 1994.

[2] J. H. Ahn, K. Kim, and H. Byun. Robust object segmentation using graph cut with object and background seed estimation. In *International Conference on Pattern Recognition*, pages 361–364, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2521-0.

[3] M. Björkman and D. Kragic. Active 3d scene segmentation and detection of unknown objects. In *Internationlal Conference on Robotics and Automation*, 2010.

[4] A. Blake. *Introduction to Active Contours and Visual Dynamics*. Department of Engineering Science, University of Oxford, June 1999.

[5] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006. ISSN 0920-5691.

[6] T. Brox. *From pixels to regions: partial differential equations in image analysis*. PhD thesis, Faculty of Mathematics and Computer Science, Saarland University, Germany, April 2005.

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *European Conference on Computer Vision*, volume 3024 of *LNCS*, pages 25–36, Prague, Czech Republic, May 2004. Springer.

[8] T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In A. Leonardis, H. Bischof, and A. Pinz, editors, *European Conference on Computer Vision*, volume 3951 of *LNCS*, pages 471–483, Graz, Austria, May 2006. Springer.

[9] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence*, 2010.

[10] T. Brox, M. Rousson, R. Deriche, and J. Weickert. Technical report 147. Technical report, 2005.

[11] T. Brox and J. Weickert. A TV flow based local scale measure for texture discrimination. In T. Pajdla and J. Matas, editors, *European Conference on Computer Vision*, volume 3022 of *LNCS*, pages 578–590, Prague, Czech Republic, May 2004. Springer.

[12] T. Brox and J. Weickert. Level set segmentation with multiple regions. *Image Processing*, 15(10):3213–3218, October 2006.

[13] S. Calinon, F. Guenter, and A. Billard. On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 37(2):286–298, 2007.

[14] T. F. Chan and L. A. Vese. Active contours without edges. *Image Processing*, 10(2):266–277, 2001.

[15] C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[16] Y. W. Chang, C. J. Hsieh, K. W. Chang, M. Ringgaard, and C. J. Lin. Training and testing low-degree polynomial data mappings via linear svm. *Journal of Machine Learning Research*.

[17] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition*, pages 1124–1131, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2.

[18] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007. ISSN 0920-5691.

[19] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *Computer Vision and Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

[20] C. Elkan. Using the triangle inequality to accelerate k-means. In *International Conference on Machine Learning*, 2003.

[21] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[22] B. K.P. Horn and B. G. Schunck. Determining optical flow. Technical report, Cambridge, MA, USA, 1980.

[23] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *European Conference on Computer Vision*, pages 65–81, London, UK, 2002. Springer-Verlag. ISBN 3-540-43746-0.

[24] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001. ISSN 0920-5691.

[25] C. Liu, W. T. Freeman, and E. H. Adelson. Analysis of contour motions. In *Neural Information Processing Systems*, 2006.

[26] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *European Conference on Computer Vision*, pages 28–42, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88689-1.

[27] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0164-8.

[28] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, April 1981.

[29] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[30] P. McCullagh and J. A. Nelder. *Generalized linear models (Second edition)*. London: Chapman & Hall, 1989.

[31] M. Mellor, B. W. Hong, and M. Brady. Locally rotation, contrast, and scale invariant descriptors for texture analysis. *Pattern Analysis and Machine Intelligence*, 30(1):52–61, 2008. ISSN 0162-8828.

[32] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Computer Vision and Pattern Recognition*, pages 22–26, 1985.

[33] M. E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Computer Vision and Pattern Recognition*, pages 1447–1454, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0.

[34] A. S. Ogale and Y. Aloimonos. A roadmap to the integration of early visual modules. *International Journal of Computer Vision*, 72(1):9–25, 2007. ISSN 0920-5691.

[35] A. S. Ogale, C. Fermuller, and Y. Aloimonos. Motion segmentation using occlusions. *Pattern Analysis and Machine Intelligence*, 27(6):988–992, 2005. ISSN 0162-8828.

[36] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding 97*, 2005.

[37] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990. ISSN 0162-8828.

[38] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *European Conference on Computer Vision*, pages 792–805, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88689-1.

[39] S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research*, 5:941–973, 2004. ISSN 1532-4435.

[40] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Computer Vision and Pattern Recognition*, pages 699–704, 2003.

[41] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2): 197–227, 1990. ISSN 0885-6125.

[42] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6 (2):461–464, 1978.

[43] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence*, 2000.

[44] A. N. Stein and M. Hebert. Combining local appearance and motion cues for occlusion boundary detection. In *British Machine Vision Conference*, 2007.

[45] A. N. Stein and M. Hebert. Local detection of occlusion boundaries in video. *Image and Vision Computing*, 27(5):514–522, 2009. ISSN 0262-8856.

[46] G. C. Stockman and L. G. Shapiro. *Computer Vision*. Prentice-Hall, 2001.

[47] C. Strecha, R. Fransens, and L. Van Gool. A probabilistic approach to large displacement optical flow and occlusion detection. In *ECCV Workshop SMVP*, pages 71–82, 2004.

[48] H. G. Sung. *Gaussian Mixture regression and classification*. PhD thesis, Rice University, 2004.

[49] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof. Tvseg - interactive total variation based image segmentation. In *British Machine Vision Conference*, Leeds, UK, September 2008.

[50] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *Pattern Analysis and Machine Intelligence*, 2007.

[51] V. Vapnik and S. Kotz. *Estimation of Dependences Based on Empirical Data: Empirical Inference Science (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387308652.

[52] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *Computer Vision and Pattern Recognition*, volume 2, page 691, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[53] A. Vezhnevets and V. Vezhnevets. Modest adaboost - teaching adaboost to generalize better, 2005.

[54] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, volume 1, pages I–511–I–518 vol.1, 2001.

[55] P. D. Wasserman. *Neural computing: theory and practice*. Van Nostrand Reinhold Co., New York, NY, USA, 1989. ISBN 0-442-20743-3.

[56] A. Wedel, D. Cremers, T. Pock, and H. Bischof. Structure- and motion-adaptive regularization for high accuracy optic flow. In *International Conference on Computer Vision*, Kyoto, Japan, 2009.

[57] A. Wedel, A. Meißner, C. Rabe, U. Franke, and D. Cremers. Detection and segmentation of independently moving objects from dense scene flow. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 14–27, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-03640-8.

[58] A. Wedel, T. Schoenemann, T. Brox, and D. Cremers. Warpcut - fast obstacle segmentation in monocular video. In *DAGM*, Heidelberg, Germany, September 2007. Springer.

[59] J. Weickert. *Anisotropic Diffusion in Image Processing*. ECMI Series, Teubner-Verlag, Stuttgart, Germany, 1998.

[60] J. Weickert and C. Schnörr. A theoretical framework for convex regularizers in pde-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264, 2001. ISSN 0920-5691.

[61] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l1 optical flow. In *British Machine Vision Conference*, London, UK, September 2009.

[62] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *European Conference on Computer Vision*, pages 211–224, 2006.

[63] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *Pattern Analysis and Machine Intelligence*, 27(10):1644–1659, 2005. ISSN 0162-8828.

[64] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *Computer Vision and Pattern Recognition*, volume 2, pages 437–444, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[65] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, and H. P. Seidel. Complementary optic flow. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 207–220, 2009.