MEAN SHIFT SEGMENTATION
Evaluation of Optimization Techniques

Jens N. Kaftan, André A. Bell and Til Aach
Institute of Imaging and Computer Vision, RWTH Aachen University, Templergraben 55, 52056 Aachen, Germany

Keywords: Mean shift, segmentation, optimization, evaluation.

Abstract: The mean shift algorithm is a powerful clustering technique, which is based on an iterative scheme to detect modes in a probability density function. It has been utilized for image segmentation by seeking the modes in a feature space composed of spatial and color information.
Although the modes of the feature space can be efficiently calculated in that scheme, different optimization techniques have been investigated to further improve the calculation speed. Besides those techniques that improve the efficiency using specialized data structures, there are other ones, which take advantage of some heuristics, and therefore affect the accuracy of the algorithm output.
In this paper we discuss and evaluate different optimization strategies for mean shift based image segmentation. These optimization techniques are quantitatively evaluated based on different real world images. We compare segmentation results of heuristic-based, performance-optimized implementations with the segmentation result of the original mean shift algorithm as a gold standard. Towards this end, we utilize different partition distance measures, by identifying corresponding regions and analyzing the thus revealed differences.

1 INTRODUCTION

Image segmentation plays a crucial role in various image processing applications in several domains, including industrial as well as medical applications. It describes the task of partitioning an image into several segments or regions. Common segmentation approaches include simple thresholding techniques (Hu et al., 2006), graph-based methods (Grady, 2006), and level set techniques (Sethian, 1999) among others. They have been applied to images from different imaging modalities in typically two or three dimensions, e.g., gray/color images, high dynamic range images, CT/MR datasets, and multispectral images. In general, these methods are adapted to the specific application.

Mean shift is an unsupervised clustering algorithm (Fukunaga and Hostetler, 1975), which estimates the gradient of a probability density function to detect modes in an iterative fashion. Hence, image segmentations that take color/intensity-similarity as well as local connectivity into account, can be obtained by applying this algorithm to the combined spatial-range domain (Comaniciu and Meer, 1997). Mean shift segmentation has been successfully applied to several applications (Comaniciu and Meer, 1999; Suri et al., 2002; Bell et al., 2006).

However, for larger images or applications where processing time is very crucial, the mean shift segmentation algorithm might be still too time consuming. Based on the EDISON framework (Christoudias et al., 2002; Comaniciu and Meer, 2002) we summarize different optimization techniques for mean shift based image segmentation. Given an application for which the mean shift algorithm has proven to be effective, one might now ask how these optimization techniques influence the segmentation results. Especially heuristic-based optimization techniques alter the segmentation result and might affect the applicability of the algorithm to the specific application. Hence, we evaluate the segmentation results of performance-optimized compared to the non-performance-optimized mean shift algorithm.

The paper is organized as follows. In Section 2 we recapitulate the theoretical background of the mean shift procedure. Subsequently, we present different performance optimization techniques in Section 3. To evaluate these heuristic-based performance optimizations with respect to the non-performance-optimized mean shift segmentation we derive evaluation meth-

Kaftan J., Bell A. and Aach T. (2008).
MEAN SHIFT SEGMENTATION - Evaluation of Optimization Techniques.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 365-374
DOI: 10.5220/0001085003650374

ods in Section 4. Consequently, we apply these evaluation methods to several different image databases and give numerical results in Section 5. Finally, we conclude our results in Section 6.

2 THEORY

The mean shift algorithm (Fukunaga and Hostetler, 1975; Cheng, 1995) can be applied to a variety of applications, including clustering, segmentation, and filtering (Comaniciu and Meer, 1997; Comaniciu and Meer, 2002) and provides consistently good results (Figure 1). The mean shift technique detects modes in a probability density function based on the Parzen Density Estimate (Fukunaga and Hostetler, 1975):

$$\hat{f}_{K_S}(\mathbf{x}) = \frac{1}{Nh^d}\sum_{i=1}^{N} K_S\Big(\frac{\mathbf{x}-\mathbf{x}_i}{h}\Big) \tag{1}$$

Here, N equals the number of d-dimensional vectors $\mathbf{x}_1..\mathbf{x}_N$. The parameter h is the window radius of the used kernel K_S. In the domain of image segmentation each feature vector is composed of the spatial information of each pixel and the corresponding color/intensity information in the range domain of dimension one or more.

The multivariate mean shift vector in the point $\mathbf{x}$ is given by (Comaniciu and Meer, 2002)

$$\mathbf{m}_K(\mathbf{x}) = \frac{\sum_{i=1}^{N} \mathbf{x}_i K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}{\sum_{i=1}^{N} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)} - \mathbf{x} \tag{2}$$

For the uniform kernel K_U the calculation of the mean shift vector (2) thus becomes an average of vector differences. It can be shown that the mean shift vector then is proportional to the normalized density gradient estimate (Comaniciu and Meer, 2002)

$$\mathbf{m}_K(\mathbf{x}) = \frac{1}{2}h^2 c \frac{\nabla \hat{f}_{K_E}(\mathbf{x})}{\hat{f}_{K_U}(\mathbf{x})} \tag{3}$$

where c is the corresponding normalization constant and K_E is the radially symmetric Epanechnikov kernel given by

$$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|\mathbf{x}\|^2) & \|\mathbf{x}\| \leq 1 \\ 0 & otherwise \end{cases} \tag{4}$$

with c_d being a normalization constant.

To ensure the isotropy of the feature space, a uniform color space, such as the $L^*u^*v^*$ is typically used. In the case of grayvalue images, the L^* component is used only. To account for different spatial and tonal variances it is reasonable to choose a kernel window of size $S_h = S_{h_s,h_r}$ with differing radii h_s in the spatial and h_r in the range domain.

(a) Original (b) Mean shift segmented

Figure 1: Cameraman image. (a) Original. (b) Mean shift segmented using high speedup level with $(h_s,h_r) = (32,16)$.

Since the mean shift vector is designed to be aligned with the local gradient estimate, it can be shown that by successive computation of (2) and shifting the kernel window by $\mathbf{m}_K(\mathbf{x})$, the mean shift procedure is guaranteed to converge to a point with zero gradient, i.e., to a mode corresponding to the initial position. Modes that are closer than h_s and h_r are grouped together. For segmentation purposes, to each pixel is then assigned the color/intensity value of the corresponding mode. Furthermore, regions with less than some pixelcount M might be optionally eliminated.

The mean shift procedure is hence an effective algorithm for mode seeking in a density distribution without prior calculation of the distribution itself.

3 OPTIMIZATION TECHNIQUES

Although the mean shift vector can be calculated efficiently by just averaging the differences between the current feature vector $\mathbf{x}$ and all feature vectors within a certain space $S_h(\mathbf{x})$ around $\mathbf{x}$, it is still expensive to identify the feature vectors falling into this space for each mean shift vector calculation. This problem is known as *multidimensional range searching* (Georgescu et al., 2003).

Different performance optimizations have been realized in the past. These mean shift variants reduce computational time either by accelerating the calculation of one mean shift vector (category I) or by reducing the number of necessary mean shift vector calculations (category II) or by combining both. Optimizations that speed up the mean shift vector calculation often take advantage of specialized data structures and therefore do not affect the segmentation re-

sult. Methods that reduce the number of necessary mean shift vector calculations, however, usually use heuristics, and therefore influence the quality of the segmentation outcome.

For the Gaussian mean shift four different acceleration strategies have been examined in (Carreira-Perpinan, 2006). The best speedup performance was achieved by a spatial discretization step, which is very similar to the mean shift vector reutilization technique described in Section 3.2. With an optimal parameter choice a speedup of factor $10\text{x} - 100\text{x}$ was achieved with an error $P < 3\%$ (number of misclustered pixels divided by the total number of pixels within the image). Note that the Gaussian mean shift segmentation is initially far slower to converge than the one using the Epanechnikov kernel.

Based on the EDISON framework (Christoudias et al., 2002; Comaniciu and Meer, 2002) that realizes different mean shift optimizations using the Epanechnikov kernel, we detail these optimization techniques in the following, which have also been shortly mentioned in (Christoudias et al., 2002).

While basic implementations of the mean shift procedure usually work on the regular lattice structure of the image, one can improve the performance utilizing an optimized "bucket" structure that we describe in Section 3.1. This optimization of category I significantly increases the computational performance without influencing the segmentation result. Furthermore, we present two optimizations of category II with medium (Section 3.2) and high (Section 3.3) speedup level, respectively. Using all possible combinations - two realizations of category I (lattice and bucket structure) $\times$ three of category II (non-optimized (as described in Section 2), medium speedup level, and high speedup level) - this sums up to six different realizations.

3.1 Bucket Data Structure

One bottleneck of the mean shift procedure is the cost per iteration, i.e., the cost of calculating the mean shift vector for a given position $\mathbf{x}$. The multidimensional range searching problem, i.e., the identification of feature vectors within a certain space $S_h(\mathbf{x})$ around $\mathbf{x}$, is fairly straightforward on the regular lattice structure of an image within spatial range h_s using a projection of $S_h(\mathbf{x})$ into the spatial domain $S_{h_s}(\mathbf{x})$ as a binary mask. Still, all those vectors need to be checked if they are also within dynamic range h_r, which is computationally expensive even though the Epanechnikov kernel has a finite support.

This number of comparisons can be reduced without impacting the result using a 3D bucket structure, where each feature vector is assigned to the corresponding 3D element of size $h_s \times h_s \times h_r$, a so called "bucket" (in case the feature vector is of higher dimension than three, only the first three elements are used for data organization). Now, each feature vector within $S_h(\mathbf{x})$ is guaranteed to be within the same or a neighboring bucket of $\mathbf{x}$. This discretization of the feature vectors into buckets allows for easy identification of feature vectors that are farther than $2 \cdot h_s$ or $2 \cdot h_r$ away from $\mathbf{x}$, and thus do not need to be considered any further. Hence, the number of comparisons is reduced on average.

3.2 Mean Shift Vector Reutilization

Another bottleneck of the mean shift procedure is the number of iterations to identify the mode corresponding to the initial position $\mathbf{x}$. Using the non-optimized mean shift definition, the mean shift vector is calculated and added to the current feature vector until convergence. This is repeated for each point within the image. Thus if we start the mean shift process from a data point $\mathbf{x}_n$, at some iteration τ the current feature vector $\mathbf{x}_n^{(\tau)}$ might become equal (or very similar) to another feature vector $\mathbf{x}_m$ and hence both $\mathbf{x}_n$ and $\mathbf{x}_m$ share the same fate, i.e., converge to the same mode. Such information is not exploited in the basic version, so that the mean shift vector of certain positions might be calculated several times, especially for positions close to a mode, that are typically traversed by many paths (iteration steps starting from a position $\mathbf{x}$).

Using this optimization, the closest point (i.e., the spatially nearest neighbor) to the current feature vector $\mathbf{x}_n^{(\tau)}$ at each iteration step τ, which we refer to as $\mathbf{x}_{\text{candidate}}$ in the following, is considered and if $\mathbf{x}_{\text{candidate}}$ is within a distance of $\varepsilon_1 \cdot h_r$ to the current feature vector $\mathbf{x}_n^{(\tau)}$, both points are assigned to the same mode. Because of the regular grid of the image and the uniform kernel chosen to calculate the mean shift vector, it is obvious that a feature vector $\mathbf{x}_n^{(\tau)}$ converges to the same mode as its nearest neighbor $\mathbf{x}_{\text{candidate}}$ on the grid if both feature vectors are identical in their range domain (after quantization of the range part of $\mathbf{x}_n^{(\tau)}$ into the image range domain). If both vectors are not equal but very similar in their range domain, however, this is not always true, but a reasonable assumption. Now, in case the corresponding mode to $\mathbf{x}_{\text{candidate}}$ has already been calculated, the iteration can be stopped early, while otherwise $\mathbf{x}_{\text{candidate}}$ can be assigned to the same mode as the initial feature vector $\mathbf{x}_n$.

Compared to this optimization technique, the spatial discretization step (Carreira-Perpinan, 2006) sub-

divides each pixel into $l \times l$ cells. Each data point $\mathbf{x}_n$ that projects onto a cell at some iteration τ_n does converge to the same mode as any other point $\mathbf{x}_m$ that projects onto the same cell at some iteration τ_m, independent of the range domain.

This mean shift vector reutilization technique is called "medium speedup" in the EDISON framework. The parameter ε_1 can be freely chosen and is preset to $\varepsilon_1 = 0.5$. A low factor will result in a small speedup and a high segmentation accuracy compared to the non-optimized case and vice versa.

3.3 Local Neighborhood Inclusion

Additionally to the heuristic described in Section 3.2 one can extend the idea of forcing feature vectors $\mathbf{x}_n^{(\tau)}$ and $\mathbf{x}_{\text{candidate}}$ that are identical in their spatial domain (after quantization into the image grid) and similar in their range domain to converge to the same mode into forcing feature vectors that are similar in both domains. That means that additionally *all* feature vectors within $S_h(\mathbf{x}_n^{(\tau)})$ are checked if their distances are smaller than $\varepsilon_2 \cdot h_r$ to $\mathbf{x}_n^{(\tau)}$. Then, in analogy to the above described method, feature vectors $\mathbf{x}_{\text{candidate}}$ within $\varepsilon_2 \cdot h_r$ distance that are not yet assigned to another mode get assigned to the same mode as the current feature vector $\mathbf{x}_n^{(\tau)}$. In case one vector candidate has already been assigned to a mode, the iteration process for $\mathbf{x}_n$ can be again stopped early.

As this optimization strategy, called "high speedup" in the EDISON framework, is more error-prone than the mean shift vector reutilization it is reasonable to choose the parameter $\varepsilon_2 < \varepsilon_1$ (ε_2 is preset to 0.1).

4 EVALUATION METHOD

Assuming the mean shift segmentation results are satisfying for a given application, we are interested in how much the heuristics of the performance-optimized mean shift segmentation will alter the segmentation result. The performance optimizations covered in Section 3.2 and 3.3 will in this respect introduce "errors" compared to the non-optimized mean shift segmentation result. Hence, to examine the influence of these optimization techniques on the mean shift segmentation, we use the original, non-optimized mean shift segmentation as reference, which we will call the "original" result in the following. To simplify matters, we limit the evaluation to single channel 2D images, but one can easily transfer the following methods and results to multichannel and multidimensional images.

As the mean shift algorithm partitions the image into several segments rather than separating an object (foreground) from the background, typical evaluation features based on the well-known true/false positive/negative notation (Udupa et al., 2002) cannot be used for validation purposes. It is also not sufficient to analyze the difference of both resulting images to compare different mean shift methods in the domain of image segmentation. In this particular case the object boundaries of all resulting regions are relevant. To identify erroneous segmented areas one needs to find corresponding regions in both resulting images, i.e., the original and comparison segmentation (see Section 4.1). Then, those pixels at identical image position that belong to non corresponding regions are erroneous segmented and can be further analyzed (see Section 4.2).

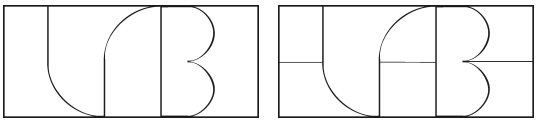
4.1 Corresponding Regions

In the output images of the segmentation step, each region is colored by the gray value of the corresponding mode. As multiple modes may have the same gray value we firstly label each connected region of pixels of identical gray value with an unique ID resulting in two partitions: O for the original and C for the comparison result. To identify corresponding regions, the number of pixels at identical image positions that belong to region $c_i \in C$, $i \in [0, n-1]$ and $o_j \in O$, $o \in [0, m-1]$ are counted with n, m being the total number of regions in C and O, respectively. These pixel-counts are then stored within a joint-histogram or correspondence matrix $A(O,C) := (a_{i,j})_{n \times m}$ as elements at position (i, j). In the following refers one row i to the pixelcounts of one region c_i in C and one column j to the pixelcounts of one region o_j in O. The identification of corresponding regions equals an optimal assignment problem that can be defined in different manners (Cardoso and Corte-Real, 2005):

Each region in C corresponds to a maximum of one region in O and vice versa. Then, O and C are *identical* if and only if each corresponding pair of regions is identical and when there are no regions without correspondence left. Otherwise, the partition distance $d_{sym}(O,C)$ equals the sum of pixels that belong to non corresponding regions in O and C and pixels of regions without correspondence. In other words, the symmetric partition distance equals

$$d_{sym}(O,C) = t - v(O,C) \tag{5}$$

with t being the total number of pixels and $v(O,C)$ denoting the value of the assignment (Cardoso and Corte-Real, 2005), which is the sum of a selection of



(a) Partition O (b) Partition C

Figure 2: The right partition is a refinement of the left partition.

elements $a_{i,j}$ of $A(O,C)$ such that no row or column contains more than one selected element. The optimal assignment is that one of all possible ones that minimizes the symmetric partition distance $d_{sym}(O,C)$ and consequently maximizes the number of pixels $v(O,C)$ at identical image positions that belong to corresponding regions c_i, o_j. This assignment problem is solved based on the Hungarian method (Kuhn, 1955) and results in a *one-to-one* matching.

Alternatively the maximum element in each row $\max_j A(O,C)\ \forall\ i$ can be located and used to assign corresponding regions c_i, o_j. Such strategy, however, can result in a *many-to-one* matching (Kaftan et al., 2006). In other words, if a region o_j within the original partition falls apart into two or more different regions, many regions c_i within the comparison partition might be assigned to o_j, allowing an oversegmentation of C compared to O. Then, C is a *refinement* (Cardoso and Corte-Real, 2005) of O (or C is *finer* than O) if and only if each region c_i in C is identical to or is contained in one region o_j in O (see Figure 2).

The resulting asymmetric partition distance $d_{asy}(O,C)$ can be obtained straightforward from matrix $A(O,C)$ as

$$d_{asy}(O,C) = t - \sum_i \left(\max_j A(O,C) \right) \qquad (6)$$

Under this asymmetric distance, any partition finer than the partition O will be at zero distance from it. Notice that, in general, $d_{asy}(O,C) \neq d_{asy}(C,O)$.

The other asymmetric distance $d_{asy}(C,O)$ can be calculated accordingly by locating the maximum element in each column of $A(O,C)$

$$d_{asy}(C,O) = t - \sum_j \left(\max_i A(O,C) \right) \qquad (7)$$

Under this asymmetric distance, any partition C for which O is an refinement will be at zero distance from it and hence allows an undersegmentation of C compared to O.

The optimal strategy, however, depends on the intended application, because different error types might have unequal strong effects on the overall performance.

4.2 Evaluation Features

Once corresponding regions have been identified, the revealed differences can be analyzed. For qualitative evaluation we show binary error images wherein each missegmented pixel is colored in black. For quantitative evaluation we have furthermore calculated different features:

For segmentation purposes it is important how the errors are distributed over the image. If many isolated pixels are evenly spread it might not be as severe as large connected regions being missegmented. Hence, we applied a connected component analysis to the resulting error images and calculated the average region size $\bar{A}_{region}$ and its standard deviation.

Using the mean shift procedure as edge preserving filtering technique, however, does not necessarily need a large overlap of corresponding regions. For such an application the grayvalue differences might be more meaningful and hence we calculate the mean grayvalue difference of missegmented pixels between the original and comparison filter output.

5 RESULTS

We have evaluated optimization techniques based on several real world images from different databases (see Figure 3). In the domain of object segmentation we have used the publicly available databases COIL-20 and COIL-100 (Nene et al., 1996a; Nene

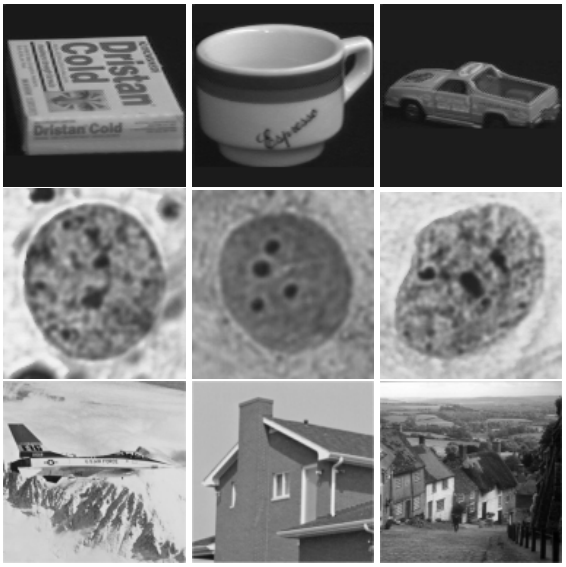


Figure 3: Example images from the COIL, CELL, and MISC databases, respectively.

Table 1: Processing times for the Cameraman image (256x256 Pixel, 256 grayvalues, Figure 1) using the lattice data structure ($\varepsilon_1 = 0.5$, $\varepsilon_2 = 0.1$) as absolute time (in seconds) and relative to the non-optimized version using the lattice structure.

Parameters	No optimization	Medium speedup	High speedup
$h_s = 8$, $h_r = 4$	6.575 (1.0)	1.704 (0.26)	0.239 (0.04)
$h_s = 16$, $h_r = 8$	40.574 (1.0)	5.997 (0.15)	0.333 (0.008)
$h_s = 32$, $h_r = 16$	279.530 (1.0)	17.343 (0.06)	0.281 (0.001)

Table 2: Processing times using the bucket data structure ($\varepsilon_1 = 0.5$, $\varepsilon_2 = 0.1$) as absolute time (in seconds) and relative to the non-optimized version using the lattice structure as listed in Table 1.

Parameters	No optimization	Medium speedup	High speedup
$h_s = 8$, $h_r = 4$	5.471 (0.83)	0.933 (0.14)	0.214 (0.03)
$h_s = 16$, $h_r = 8$	27.075 (0.67)	2.949 (0.07)	0.229 (0.006)
$h_s = 32$, $h_r = 16$	180.909 (0.64)	8.471 (0.03)	0.219 (0.0008)

et al., 1996b). Each available object was segmented at grayscale level using three different rotations, summarizing to a total of 360 images (in the following referred to as COIL-database). In the medical domain we have examined 200 brightfield light microscopy cell images (Bell et al., 2006) from human oral mucosae in Feulgen and Silver staining each (CELL-database). Additionally we have used a set of 20 different images, cropped and scaled to a common image size to allow a better comparison, that have been used in earlier publications (see especially (Comaniciu and Meer, 1999)) to show the effectiveness of the original mean shift procedure (MISC-database). Each performance-optimized segmentation has been validated (see Section 5.1 and Section 5.2) in comparison to the original result as reference. Since the used data structure does not influence the segmentation result, we have used the bucket structure (see Section 3.1) for all experiments analyzing the segmentation accuracy. The computational efficiency has been examined using all possible permutations of optimization techniques. Considering both aspects, segmentation quality and computation time, allows to rate the applicability of each technique to the application of interest, based on its requirements.

To measure the processing time, the Cameraman image (256x256 Pixel, 256 grayvalues, Figure 1) has been processed using various spatial and range kernel radii on a Pentium 4, 3.4GHz. The results averaged over three runs are shown in Table 1 using the lattice and in Table 2 using the bucket structure. Additionally to the absolute times (in seconds) the relative time according to the non-optimized version using the lattice structure are listed.

5.1 Segmentation Quality

To evaluate the segmentation quality, each image has been segmented without using optional post-processing steps to avoid their influence on the segmentation output. Using the mean shift procedure alone, however, will still result in an image containing several very small regions depending on the kernel size. To circumvent the image falling apart into too many regions we have chosen relatively large kernel sizes $(h_s, h_r) = (15, 25)$. The parameters of the optimization techniques have been set to $\varepsilon_1 = 0.5$ and $\varepsilon_2 = 0.1$.

To quantify the differences between the non-optimized and optimized versions of the mean shift procedure, the resulting images have been compared using the symmetric partition distance $d_{sym}(O,C)$ and both asymetric partition distances $d_{asy}(O,C)$ and $d_{asy}(C,O)$, which allow the optimized result to be oversegmented, resp. undersegmented compared to the original result. The resulting difference images are exemplarily shown in Figure 4. Averaged over all test images the relative number on missegmented pixels ranges between 4.6% and 18.2% for the mean shift vector reutilization and between 6.2% and 23.6% for the local neighborhood inclusion technique depending on the chosen partition distance and database (see Table 3).

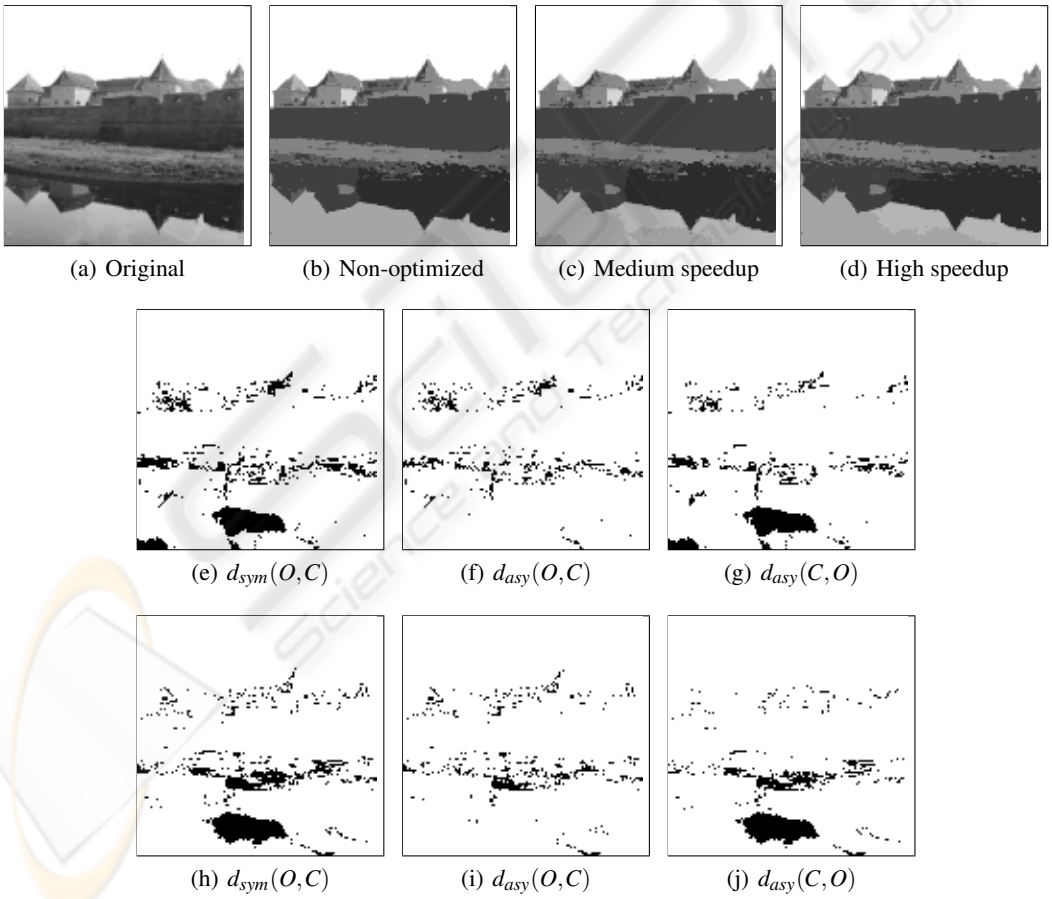
However, as the test set includes different images, it is difficult to compare the absolute number of missegmented pixels for both optimization techniques. Looking at each individual image, we have therefore calculated the relative increase of missegmented pixels of the local neighborhood inclusion technique in relation to the number of missegmented pixels using the mean shift vector reutilization technique and hence obtain a measure that considers the complexity of the segmentation task. Determining the median

Table 3: Averaged segmentation error. For each optimization technique, the partition distances are listed as absolute pixel count and relative to the image size (mean value ± standard deviation). Note that the image size within one database is constant while varying between different databases.

	MISC-database		**COIL-database**	
	Medium speedup	High speedup	Medium speedup	High speedup
$d_{sym}(O,C)$	2807 (18.0±11.2%)	3163 (20.2±10.2%)	2217 (13.5±11.7%)	2802 (17.1±12.7%)
$d_{asy}(O,C)$	1209 (7.7±4.2%)	1433 (9.2±5.2%)	841 (5.1±5.5%)	1167 (7.1±7.1%)
$d_{asy}(C,O)$	2120 (13.6±10.8%)	2333 (14.9±8.3%)	1738 (10.6±10.6%)	2124 (13.0±11.0%)

Averaged segmentation error relative to the image size (image size varies within CELL-database).

	CELL-database	
	Medium speedup	High speedup
$d_{sym}(O,C)$	18.2±14.2%	23.6±15.1%
$d_{asy}(O,C)$	4.6±4.0%	6.2±5.6%
$d_{asy}(C,O)$	15.8±13.5%	20.3±14.1%



(a) Original (b) Non-optimized (c) Medium speedup (d) High speedup

(e) $d_{sym}(O,C)$ (f) $d_{asy}(O,C)$ (g) $d_{asy}(C,O)$

(h) $d_{sym}(O,C)$ (i) $d_{asy}(O,C)$ (j) $d_{asy}(C,O)$

Figure 4: (a) Fagaras image. (b-d) Mean shift segmented using different speedup levels with $(h_s,h_r) = (15,25)$. (e-g) Partition distance of (c) and (b). (h-j) Partition distance of (d) and (b) (missegmented pixels are shown in black color).

Table 4: Averaged size of missegmented regions in pixel. The common image sizes of the MISC- and COIL-database are 125x125, and 128x128 respectively. The averaged standard deviation of the region sizes within each image is listed in brackets.

	MISC-database		**COIL-database**	
	Medium speedup	High speedup	Medium speedup	High speedup
$d_{sym}(O,C)$	19.2 (137.3)	17.7 (143.7)	67.6 (233.7)	66.0 (299.3)
$d_{asy}(O,C)$	6.1 (26.0)	6.8 (37.4)	14.5 (48.9)	12.2 (69.0)
$d_{asy}(C,O)$	16.6 (119.3)	14.4 (117.2)	83.3 (209.1)	66.9 (252.9)

Averaged size of missegmented regions relative to the image size on the CELL database. The averaged standard deviation of the relative region sizes within each image is listed in brackets.

	CELL-database	
	Medium speedup	High speedup
$d_{sym}(O,C)$	0.26 (3.3)	0.33 (4.0)
$d_{asy}(O,C)$	0.04 (0.2)	0.06 (0.4)
$d_{asy}(C,O)$	0.32 (4.9)	0.35 (4.9)

Table 5: Averaged mean difference between the grayvalues of missegmented pixels compared to the non-optimized segmentation result in absolute values and relative to the dynamic range (mean value $\pm$ standard deviation).

	MISC-database		**COIL-database**	
	Medium speedup	High speedup	Medium speedup	High speedup
$d_{sym}(O,C)$	14.5 (5.7$\pm$2.0%)	14.0 (5.5$\pm$2.1%)	19.5 (7.6$\pm$2.9%)	19.7 (7.7$\pm$3.0%)
$d_{asy}(O,C)$	17.4 (6.8$\pm$2.4%)	16.5 (6.5$\pm$2.4%)	22.6 (8.8$\pm$3.0%)	22.0 (8.6$\pm$3.6%)
$d_{asy}(C,O)$	14.9 (5.8$\pm$2.1%)	14.7 (5.7$\pm$2.5%)	20.5 (8.0$\pm$3.2%)	21.6 (8.4$\pm$3.3%)

Averaged mean grayvalues difference on the CELL database.

	CELL-database	
	Medium speedup	High speedup
$d_{sym}(O,C)$	15.3 (6.0$\pm$2.3%)	14.3 (5.6$\pm$2.0%)
$d_{asy}(O,C)$	21.0 (8.2$\pm$2.0%)	19.9 (7.8$\pm$2.3%)
$d_{asy}(C,O)$	16.2 (6.3$\pm$3.3%)	14.8 (5.8$\pm$3.0%)

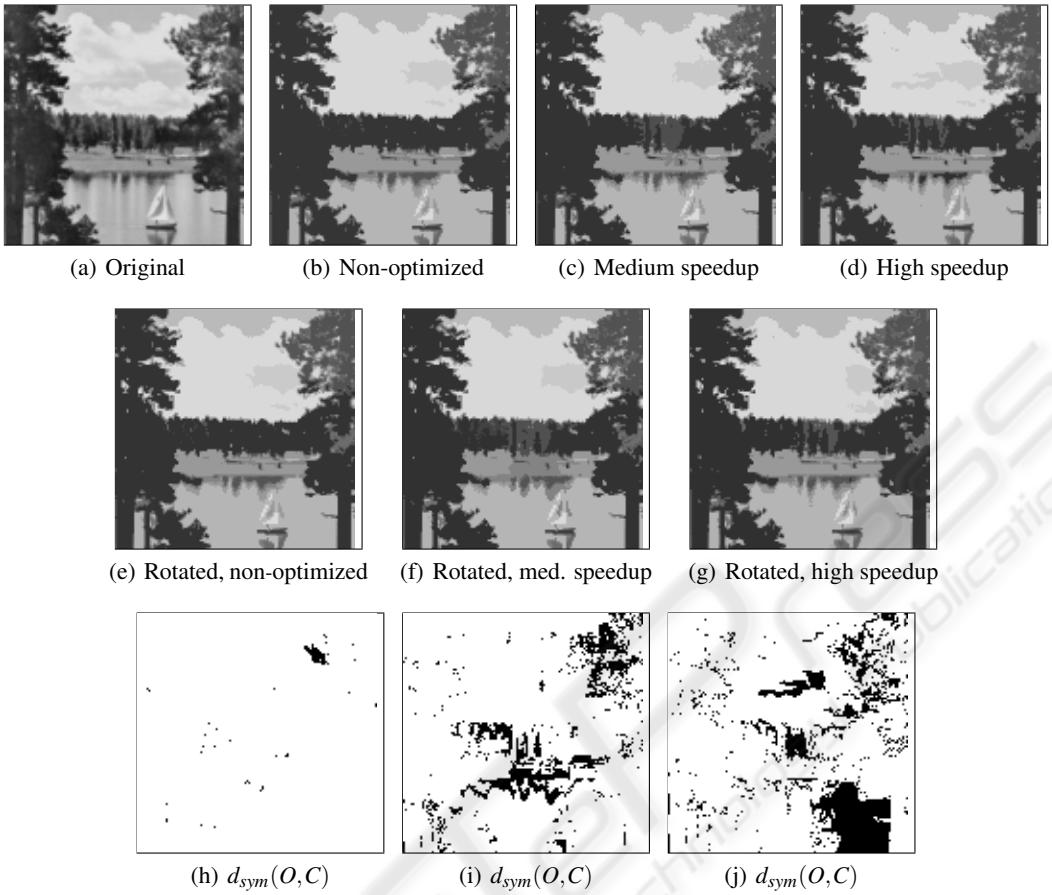
value of this measure shows that the number of missegmented pixels grows by 24.7% using the symmetric distance measure and by 19.7% and 24.3% using $d_{asy}(O,C)$ and $d_{asy}(C,O)$, respectively.

Next, the average missegmented region size did not significantly change between the optimization techniques. However the increased averaged standard deviation of the region sizes within each image indicates that the local neighborhood inclusion techniques produces a larger variance ($d_{sym}(O,C)$ and $d_{asy}(O,C)$) and hence also larger connected regions of missegmented pixel than the mean shift vector reutilization technique (see Table 4).

Finally, the average mean difference of the missegmented pixel gray values compared to the non-optimized segmentation result did not change significantly (see Table 5).

5.2 Rotational Invariance

The heuristic-based optimization techniques both access earlier calculated mean shift vectors to estimate the correct mean shift vector at some positions and hence result in a mean shift procedure, which is not rotationally invariant anymore. In fact, the ordering of the pixels being processed may have a significant impact on the segmentation result. To verify this observation, we have exemplarily rotated one image by 90° and compared its segmentation using $(h_s,h_r) = (15,25)$ for each optimization technique to the segmentation result of the non-rotated image. Figure 5 shows on the basis of the error images using the symmetric partition distance that the local neighborhood inclusion technique is stronger corrupted (as expected) than the mean shift vector reutilization technique, while the non-optimized version can be con-



(a) Original (b) Non-optimized (c) Medium speedup (d) High speedup

(e) Rotated, non-optimized (f) Rotated, med. speedup (g) Rotated, high speedup

(h) $d_{sym}(O,C)$ (i) $d_{sym}(O,C)$ (j) $d_{sym}(O,C)$

Figure 5: (a) Lake image. (b-d) Mean shift segmented using different speedup levels with $(h_s,h_r) = (15,25)$. (d-f) Mean shift segmented using identical parameters, but a 90° rotated input image. (h-j) Partition distance of (b-d) and (e-g) using $d_{sym}(O,C)$ comparing corresponding optimization techniques.

sidered to be practically rotationally invariant. In fact, even for the non-optimized version, a few error pixels can be observed due to numerical reasons. Analyzing the error images shows (see Table 6) that 16.1% of the pixels are affected using the local neighborhood inclusion speedup technique and 11.1% using the mean shift vector reutilization technique (compared to 0.6% for the non-optimized version) with an average region size of 10.56 and 9.75 pixels, respectively.

Table 6: Segmentation error caused by rotational variance. For each optimization technique, the symmetric partition distance $d_{sym}(O,C)$ and the average error region size $\overline{A}_{region}$ are listed from the images shown in Figure 5.

Optimization	$d_{sym}(O,C)$	$\overline{A}_{region}$
Non-optimized	96 (0.6%)	3.84
Medium speedup	1742 (11.1%)	10.56
High speedup	2516 (16.1%)	9.75

6 CONCLUSIONS

We have quantitatively evaluated different optimization techniques for the Epanechnikov based mean shift segmentation algorithm regarding their accuracy and computational performance. As expected, each optimization technique that reduces the number of necessary mean shift vector calculations introduces some errors compared to the non-optimized mean shift procedure. On the one hand, the quantity of errors has been measured to be on average 18.0% for the mean shift vector reutilization and 20.2% for the local neighborhood inclusion technique using a symmetric partition distance measure. However, the impact of such optimizations on the segmentation accuracy depends on the image material. Depending on the application the fact that using the described heuristics results in a rotationally variant segmentation (up to 16.1% of the pixels have been affected in our example) might be even more severe than the previously

mentioned errors. On the other hand, the improvement of the computational efficiency is remarkable. Especially for large kernel radii, the processing time could be reduced from 280s to 0.219s on a standard PC in our example. Finally, it depends on the requirements of the intended application if the described optimization techniques are applicable but analyzing our results one can rate the pros and cons of each technique.

ACKNOWLEDGEMENTS

We would like to thank the authors of (Christoudias et al., 2002) for providing the EDISON software at their laboratory website (http://www.caip.rutgers.edu/riul/) and the authors of (Cardoso and Corte-Real, 2005) for providing their evaluation application upon request.

REFERENCES

Bell, A. A., Kaftan, J. N., Aach, T., Meyer-Ebrecht, D., and Böcking, A. (2006). High Dynamic Range Images as a Basis for Detection of Argyrophilic Nucleolar Organizer Regions Under Varying Stain Intensities. In *IEEE International Conference on Image Processing. ICIP 2006*, pages 2541–2544.

Cardoso, J. and Corte-Real, L. (2005). Toward a generic evaluation of image segmentation. *IEEE Transactions on Image Processing*, 14(11):1773–1782.

Carreira-Perpinan, M. A. (2006). Acceleration strategies for gaussian mean-shift image segmentation. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2006*, pages 1160–1167.

Cheng, Y. (1995). Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799.

Christoudias, C. M., Georgescu, B., and Meer, P. (2002). Synergism in Low Level Vision. In *IEEE International Conference on Pattern Recognition. ICPR 2002*, volume 4, pages 150–155.

Comaniciu, D. and Meer, P. (1997). Robust Analysis of Feature Spaces: Color Image Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition. CVPR 1997*, pages 750–755.

Comaniciu, D. and Meer, P. (1999). Mean Shift Analysis an Applications. In *International Conference on Computer Vision. ICCV 1999*, volume 2, pages 1197–1203.

Comaniciu, D. and Meer, P. (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

Fukunaga, K. and Hostetler, L. D. (1975). The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Transactions on Information Theory*, 21(1):32–40.

Georgescu, B., Shimshoni, I., and Meer, P. (2003). Mean Shift Based Clustering in High Dimensions: A Texture Classification Example. In *International Conference on Computer Vision. ICCV 2003*, volume 1, pages 456–463.

Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783.

Hu, Q., Hou, Z., and Nowinski, W. L. (2006). Supervised range-constrained thresholding. *IEEE Transactions on Image Processing*, 15(1):228–240.

Kaftan, J. N., Kiraly, A. P., Naidich, D. P., and Novak, C. L. (2006). A Novel Multi-Purpose Tree and Path Matching Algorithm with Application to Airway Trees. In *SPIE Medical Imaging 2006: Physiology, Function, and Structure from Medical Images*, volume 6143, pages 215–224.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97.

Nene, S., Nayar, S., and Murase, H. (1996a). Columbia Object Image Library (COIL-100). Technical report, Computer Vision Laboratory, Columbia University.

Nene, S., Nayar, S., and Murase, H. (1996b). Columbia Object Library (COIL-20). Technical report, Computer Vision Laboratory, Columbia University.

Sethian, J. A. (1999). *Level Set Methods and Fast Marching Methods*. Cambridge University Press.

Suri, J. S., Setarehdan, S. K., and Singh (Eds), S. (2002). *Advanced Algorithmic Approaches to Medical Image Segmentation*. Springer.

Udupa, J. K., LaBlanc, V. R., Schmidt, H., Imielinska, C., Saha, P. K., Grevera, G. J., Zhuge, Y., Currie, L. M., Molholt, P., and Jin, Y. (2002). Methodology for evaluating image-segmentation algorithms. In *SPIE Medical Imaging: Image Processing*, volume 4684, pages 266–277.