MIR: An Approach to Robust Clustering - Application to Range Image Segmentation

Klaus Köster and Michael Spann

School of Electronic and Electrical Engineering

The University of Birmingham, UK

Edgbaston, Birmingham B15 2TT,UK

E-Mail: koesterk@eee.bham.ac.uk, M.Spann@bham.ac.uk

Telephone: ++44 121 414 4328 Fax: ++44 121 414 4291

Correspondence to:

Dr. Michael Spann

School of Electronic and Electrical Engineering

The University of Birmingham, UK

Edgbaston, Birmingham B15 2TT,UK

E-Mail: M.Spann@bham.ac.uk

Telephone: ++44 121 414 4328

Fax: ++44 121 414 4291

Abstract - This paper describes an unsupervised region growing technique based on a novel robust statistical test. The merging decision is derived from the mutual inlier ratio (MIR) of adjacent regions. This ratio is computed using robust regression techniques [1] and a novel method to estimate the robust scale of the Gaussian distribution. A discrimination value to recognize identical Gaussian distributions with the MIR is derived theoretically as a function of the sizes of the compared sets. The method to test distributions is compared with the established Kolmogorov-Smirnov test and implemented into a segmentation algorithm for planar range images. The iterative growing technique is evaluated using the detailed framework for range image segmentation comparison of Hoover *et al* [2] involving 60 real range images.

Index Terms - Segmentation, Robust Statistics, Clustering, Region Growing, Range Images, Least-Median-of-Squares

1 Introduction

The process of segmentation in computer vision groups parts of an image into segments (regions) that are homogeneous according to some criteria. Popular image processing textbooks categorise segmentation algorithms into *region growing* and *edge-based* methods [3, 4]. However, algorithms described in the literature very often break this strict classification and use region-finding techniques together with line-finding methods in order to overcome drawbacks of the separated approaches and to produce more reliable segmentations [5, 6, 7, 8]. In this paper we propose such a hybrid technique overcoming several known problems of the classical region growing approach [9].

Region growing methods start in general by segmenting the image into initial cells (*seeding*). There are several methods for this step proposed in the published literature: interactive [10] or random [8] techniques, regular [9, 11] or data-driven [12, 6] methods. The later stages of the algorithm have to be stable to cope with badly placed initial seeds if using interactive, random or regular seeding. A robust estimation of the initial model parameters is difficult as the seeds are in general small and do not provide much information. Placing the seeds on a boundary or a noise corrupted part of the image can result in a break-down of the estimation. In the worst case this creates regions which are not present in the data. Our algorithm generates the initial region map incorporating a step to avoid this bad seeding and improving the results of the initial parameter estimates especially for small regions.

The control strategies to solve image segmentation problems can be split up into model-driven methods (*top-down*), data-driven methods (*bottom-up*) or mixtures of both. Although top-down methods seem to be more appealing since there is evidence of similarities to the human cognitive process [13, 14], they have the disadvantage that they require a large amount of *a-priori* knowledge (e.g. object models, number of regions etc.) which tend to severely limit the scope of applicability. Bottom-up strategies start by combining local properties in the images into global ones. Local operators working on small data sets have to come to a robust decision in order to overcome the presence of artifacts in the data. There is a large amount of published literature using robust regression techniques, in particular, for the problem of range image segmentation [11, 15, 16, 17, 18]. This work extensively uses the *least median of squares* (LMedS) estimator, proposed by Rousseeuw [1], providing a robust model estimate in data sets with up to 50% of the data heavily corrupted by outlying data. Rousseeuw [1, p. 166] also reported good performance of the LMedS estimator in the case of particularly small data sets.

Given the initial seeding and a good estimate of the model parameters, the final region map has to be found by growing/merging the seeds. Popular approaches to this problem are based on global optimisation methods, which merge regions in order to minimise an energy function [19, 20, 21]. A more local approach is the application of a statistical test to obtain a similarity measure as a decision criteria for region merging. The advantage of these methods are decisions based on general assumptions on the data avoiding any thresholds which require special tuning for different data sets. Well-known choices for statistical tests are Fisher's *F-test* or the Kolmorogov-Smirnov *KS-test* [22]. While the first one cannot distinguish between two distributions with the same mean but different variances [8], the second needs larger data sets to come to reliable decisions. In this paper we propose a statistical similarity test based on robust outlier/inlier detection overcoming these drawbacks.

The evaluation of algorithms in computer vision is very often a neglected subject. A helpful framework for comparing range image segmenters was proposed by Hoover *et al* [2].

For this reason we decided to apply our ideas to the problem of range image segmentation for planar regions. However, the technique is not restricted to planar models, scalar data or 2D images and has already been applied to the segmentation of 3D seismic data using constant models [23, 24]. The evaluation involves 60 real range finder images measured by two types of sensors. An objective tool compares the result with manually reconstructed ground truth images[1]. The algorithm will be compared with four published range image segmentation techniques.

In Section 2 we elucidate the data model used for the case of range image segmentation. The LMedS estimation procedure of the model parameters will be described and a novel method for the estimation of the variance will be proposed. Section 2 finishes with a description of how the statistical similarity of two distributions is determined using our method. In Section 3 we illustrate our segmentation technique. Section 4 describes the evaluation of the algorithm and compares it with other methods. The section finishes with a discussion of the advantages and disadvantages of our method. Finally we give our conclusions in Section 5.

2 Theory

The segmentation process groups pixels to form higher-level regional image structures sharing the same properties. More formal definitions of the term *segmentation* can be found in [3, 25]. In this work and in particular in the later evaluation, the formal definition of Hoover *et al* [2] is applied. The criterion for assigning range image pixels to the same region is a polynomial model with additive Gaussian noise. Section 2.1 describes this data model and the inlier/outlier discrimination rule. To estimate the model parameters, the LMedS estimation technique is applied. This method is briefly explained in Section 2.2, which also discusses the estimation of the variance of the additive noise and proposes a method generating better results than the commonly used methods. Section 2.3 derives some novel aspects for statistically comparing data sets which are later applied in the segmentation algorithm.

[1] The comparison tool and the test images are publicly available via *http://marathon.csee.usf.edu/seg-comp/SegComp.html*

2.1 The Data Model

A polynomial description of the region data has been chosen to group pixels into connected regions. Given a set of pixels $V = \{v_1(x, y), v_2(x, y), \ldots v_N(x, y)\}$, the data model is specified by

$$v_i(x, y) = a_x x + a_y y + a_0 + e_i, \tag{1}$$

where the $a_{\{x,y,0\}}$ denotes the model parameters and e_i is an independent, additive, normally distributed noise term. In this case the model parameters can be estimated by minimum least squares (LS) techniques. Unfortunately, in the presence of outliers (eg. erroneous measurements, pixels belonging to another region) least-squares estimation breaks down. However, the LMedS estimator described in [1, 11] is capable of rejecting up to 50% of outliers, which is the basis of the estimation procedure in this work.

The noise term e_i in the model is taken into consideration by computing the variance s^2 of the distribution for each region. The estimation procedure is the subject of Section 2.2. To assign a sample $v(x, y)$ as an inlier within a region's model, the decision rule

$$\left|v(x, y) - \hat{v}(x, y)\right| \leq \chi_{p,97.5\%} \cdot s \tag{2}$$

has to be true. This rule has been derived using the *squared Mahalanobis* distance [1, p.223], which allows an easy extension to multi-dimensional and/or vectorial data. The value $\chi_{p,97.5\%}$ is from chi-square statistics, $\hat{v}(x, y)$ is an estimate of the model $v_i(x, y)$ using the least median of squares estimates of the model parameters, the operator $|\cdot|$ returns the absolute value and p denotes the degrees of freedom, which is equal to 1 (vectorial dimension of the data in this case).

2.2 Least Median of Squares Estimation

This section starts with a brief description of the *least median of squares* (LMedS) estimation technique. Additional information on this subject can be found in [1, 11, 15].

Using a linear or higher order polynomial model, the LMedS estimate has to be computed by an iterative projection pursuit method using sub-samples of the set. For a constant model this can be replaced by a straight forward approach involving a single sorting step of the set. For the purposes of this work the estimation procedure can be summarised as follows:

1. Randomly select a 3-tuple Π from the data. The 3 points have to be in a general position and *must not* lie on a line or a single point.

2. For this 3-tuple compute the model parameters $a_x(\Pi)$ and $a_y(\Pi)$.

3. Compute the residuals $r_i(\Pi)$ by

$$r_i(\Pi) = z_i - a_x(\Pi)x_i - a_y(\Pi)y_i, \qquad i = 1, \cdots, N. \tag{3}$$

4. Sort the sequence of residuals to produce the sorted list $\{r_{1:N}(\Pi) \leq r_{2:N}(\Pi) \cdots r_{N:N}(\Pi)\}$. Compute the absolute value of the model a_0 from

$$a_0 = \frac{r_{p:N}(\Pi) + r_{p+N/2:N}(\Pi)}{2}, \qquad \text{where} \qquad p = \arg\min_{h=1\cdots N/2} \big(r_{h:N}(\Pi) - r_{h+N/2:N}(\Pi)\big). \tag{4}$$

$N/2$ is computed by integer division and if N is odd the index h runs from $1, \cdots, (N/2+1)$. Store the corresponding shortest half-window size $\delta(\Pi)$

$$\delta(\Pi) = r_{p:N}(\Pi) - r_{p+N/2:N}(\Pi). \tag{5}$$

5. Repeat steps 1 to 4 q times. The final LMedS estimate are from the 3-tuple yielding the minimum shortest half-window size $\delta = \min_{\Pi} \delta(\Pi)$.

The estimation of the standard deviation of the data set is a crucial point of the estimation procedure as this value is used for classifying inliers in expression 2. While an over-estimated value accepts too many inliers and the final segmentation is likely to be over-merged, an under-estimated standard deviation can cause the contrary result. Figure 1 shows the true distribution of the model residuals $p\big(v_i(x,y) - \hat{v}(x,y)\big)$ and compares it with the assumed

Gaussian model with the scale σ estimated by different estimation procedures. The data was obtained from a large single region ($\approx$ 6000 elements) in a real range image (ABW test set, Section 4.1). The displayed true distribution is computed by differentiating the cumulative distribution of the residuals. The figure documents the typical error of scale estimation procedures discussed in the sequel.

A well known approach for estimating the standard deviation is the average absolute difference from the least mean [26, p. 611]. Unfortunately outliers in the data have a large impact on the result leading to an over-estimated standard deviation. Figure 1 shows the graph of a typical estimation result which is too flat and too wide compared to the true distribution. A more robust estimate comes with the LMedS estimator and is given by Rousseeuw [1, p.202]:

$$s = 1.4826\left(1 + \tfrac{5}{N-2}\right)\delta, \tag{6}$$

where 1.4826 is the correction factor for median based Gaussian noise standard deviation estimates, the term $1 + \frac{5}{N-2}$ corrects an under-estimation of the standard deviation occurring in small samples and δ is computed by equation 5. Applying this estimate to range data we have found that this equation still over-estimates the real value for larger set sizes (>200 elements). A re-weighting version of this estimator also proposed by Rousseeuw improves the low convergence rate of the LMedS estimator by recomputing the mean and the standard deviation from inliers only. These iterative steps result in an under-estimation of the standard deviation as only the smallest residuals contribute to the new estimate. Following the Rousseeuw [1, p.44] proposed re-weighting steps with an ideal Gaussian distribution results in a under-estimation of the standard deviation of approximately 4.5$\%$ per iterative step since the tails of the distribution cannot be fully evaluated in order to maintain robustness. The proposed second re-weighting step results already in an accumulated error of around 10$\%$. We propose multiplying the obtained standard deviation of each re-weighting step with 1.0476 to prevent introducing this error. This value can be derived by comparing the true standard deviation with the value obtained from the Rousseeuw proposed percentage of inliers of the distributions.

However, in this work we estimated the standard deviation using the cumulative distribution of the residuals. Given an ideal Gaussian distribution of the form

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \tag{7}$$

the standard deviation σ is computed by

$$\sigma = \frac{1}{\sqrt{2\pi} \cdot p(x=0)}. \tag{8}$$

The value $p(x = 0)$ can be estimated from the cumulative distribution $P(x)$ of the model residuals:

$$p(x = 0) = \frac{d}{dx} P(x)|_{x=0}. \tag{9}$$

To obtain the cumulative distribution the residuals have to be sorted into ascending order:$\{r_{1:N}(\Pi) \leq r_{2:N}(\Pi) \leq \cdots \leq r_{n:N}(\Pi) \leq \cdots \leq r_{N:N}(\Pi)\}$. This involves no additional computational work as the LMedS estimation incorporates such a step already. Now the function values of the cumulative distribution $P(x)$ can be computed as

$$P(r_{n:N}(\Pi)) = \frac{n}{N}, \tag{10}$$

where n is the position of $r_{n:N}(\Pi)$ in the sorted set. The value $p(x = 0)$ is then computed by the gradient of a line fit of a few residuals having the smallest absolute value. In our experiments involving data from 40 real range images this method returned better results than the previously discussed techniques. To get a good idea of the cumulative distribution larger sets are necessary, therefore we recommend using this method only from set sizes of around 40 elements onwards. For smaller sets the obtained cumulative distribution is no longer a good descriptor of the true distribution and the calculated gradient suffers from this (see also figure 4).

To find the optimum solution for the model parameter estimation all possible 3-tuples in the data set have to be tried. However, a more feasible number of replications q can be derived from the probability that at least one out of q sub-samples consists exclusively of "good" points, which is for this case approximately [1, p. 260]

$$1 - (1 - (1 - \varepsilon)^3)^q, \tag{11}$$

where ε is the fraction of outliers. One obtains the number q of replications by imposing that this probability exceeds a given value.

To improve the very slow convergence rate of the LMedS estimator of $n^{-\frac{1}{3}}$ [1, p. 178] we applied two iterative least mean re-weighting steps on the inlier only [1, p. 17] resulting in a better precision of the model parameters.

2.3 Statistical Distribution Comparison

In general the merging stage of a region-growing algorithm compares statistical measures of two adjacent regions and merges them or not according to the test result. There are several methods for comparing statistical distributions like Student‘s *t test* for testing the distribution's means, the *F-test* for testing the distribution's variances or *Chi-Square test* and the *Kolmorogov-Smirnov* (KS) test for comparing distributions in general. The problem for a segmentation algorithm using one of these methods is to incorporate higher level models to describe the region data. For example, in a range image segmentation algorithm using a linear or higher order model the above test cannot be applied directly to the data. A general approach to solve this problem is the application to the residuals after model fitting.

This work's approach to the comparison of two adjacent regions is based on the robust detection of inliers of the region model. The idea is to compare two distributions using the number of mutually inlying data elements. For the test we determine at first the *mutual inlier ratio* (MIR): Let R_A be the set of elements of region A. The set of inliers I_A of region A is

$$I_A = \left\{ v \in R_A : |v - \hat{v}_A| \leq \chi_{p,97.5\%} \cdot s_A \right\}, \tag{12}$$

where $\hat{v}_A$ denotes the LMedS estimate of the model of region A and s_A is the estimated standard deviation. The coordinates have been dropped for convenience. To determine the mutual inlier ratio M_{AB} of two regions A and B, those inliers of A, which are also inlying in B have to be determined:

$$I_{AB} = \left\{ v \in I_A : |v - \hat{v}_B| \leq \chi_{p,97.5\%} \cdot s_B \right\}. \tag{13}$$

Now the mutual inlier ratio M_{AB} becomes:

$$M_{AB} = \frac{\mathfrak{card}(I_{AB})}{\mathfrak{card}(I_A)}, \tag{14}$$

where the operator $\mathfrak{card}()$ returns the number of elements of a set.

Having computed the ratio, we compare the result with a discriminating value. The two distributions are found to be similar if the ratio is larger than the discriminating value. A safe test result can only be obtained if the similarly computed M_{BA} is *also* larger than the discriminating value. This avoids the situation where regions with large variances accept too many adjacent regions with smaller variances as inliers leading to an over-merged result. Certainly the discriminating value is crucial and determines the extent of under- or over-merging significantly. Meer *et al* [11] applied a value of 0.8 in their algorithm. A value of 0.5 was sufficient to find two distributions similar if an additional smoothness criterion was satisfied. In the following paragraphs an expression for the discriminating value will be determined based on statistical arguments.

The applied discriminating value has to ensure that two given data sets of the same Gaussian distribution $N(\mu, \sigma)$ with sizes n_A and n_B are tested to be of the same distribution. Only the estimated values of mean and standard deviation are available for this test. It is well-known that the precision of a statistical estimate of a data set's distribution parameters depends on the size of the set. For a region merging algorithm, where the initial region size is in general limited, the exact estimation of the model parameters suffers from the standard deviation of the model σ and from an additional finite sample size error, which decreases with the set size n. The relationship between the exact unknown mean of a distribution μ and the estimated mean $\bar{X}$ can be expressed by [27, p. 313]:

$$\mu = \bar{X} \pm d, \qquad d = t_{(n-1),\alpha/2} \frac{s}{\sqrt{n}}, \tag{15}$$

where s denotes the estimated standard deviation, n is the set size and $t_{(n-1),\alpha/2}$ is based on the Student t distribution with a $100(1-\alpha)\%$ confidence interval and $n-1$ degrees of freedom allowing us to use the estimated standard deviation instead of the true (unknown) standard deviation σ. A confidence interval for the estimated standard deviations s can be constructed using the chi-square distribution and the standard deviation σ [27, p. 325]:

$$\frac{\sigma^2 \chi^2_{(n-1),\alpha/2}}{n-1} \leq s^2 \leq \frac{\sigma^2 \chi^2_{(n-1),1-\alpha/2}}{n-1}. \tag{16}$$

Having two estimates of the same distribution, the worst case occurs if estimate $\bar{X}_A$ equals $\mu + d_A$ and estimate $\bar{X}_B$ equals $\mu - d_B$. A theoretical normalised value for the result of equation 12 can be computed by:

$$\overline{I}_A = \frac{1}{\sqrt{2\pi} s_A} \int\limits_{\bar{X}_A - s_A \chi_{p,97.5\%}}^{\bar{X}_A + s_A \chi_{p,97.5\%}} e^{-\frac{(x-\bar{X}_A)^2}{2s_A^2}} dx = 0.975. \tag{17}$$

A discrimination value finding the two estimates of the same distribution to be similar in this worst case can theoretically be determined by solving the integral over the overlapping inlier area of the two estimated distributions:

$$\begin{aligned} \overline{I}_{AB} &= \frac{1}{\sqrt{2\pi} s_B} \int\limits_{\bar{X}_A - s_A \chi_{p,97.5\%}}^{\bar{X}_B + s_B \chi_{p,97.5\%}} e^{-\frac{(x-\bar{X}_B)^2}{2s_B^2}} dx = \frac{1}{\sqrt{2\pi}} \int\limits_{b_{low}}^{\chi_{p,97.5\%}} e^{-\frac{x^2}{2}} dx, \\ \text{where} \quad b_{low} &= s_{ratio} \left(\tfrac{t_{(n_A-1),\alpha/2}}{\sqrt{n_A}} - \chi_{p,97.5\%} \right) + \tfrac{t_{(n_B-1),\alpha/2}}{\sqrt{n_B}} \\ \text{and} \quad s_{ratio} &= \tfrac{\sqrt{n_B - 1} \cdot \chi_{(n_A-1),1-\alpha/2}}{\sqrt{n_A - 1} \cdot \chi_{(n_B-1),\alpha/2}}. \end{aligned} \tag{18}$$

The steps between these two expressions for the discrimination value can be found in the appendix. Finally the worst case discrimination value $M_{\text{DISC},97.5\%}$ can be computed by

$$M_{\text{DISC},97.5\%} = \frac{\overline{I}_{AB}}{\overline{I}_{A}}. \tag{19}$$

Note that the resulting value for $M_{\text{DISC},97.5\%}(a, b)$ for set sizes a and b is in general unequal to $M_{\text{DISC},97.5\%}(b, a)$ for finite values of $a \neq b$. This is not surprising as $M_{AB} \neq M_{BA}$.

Equation 19 expresses the discrimination value as a function of the set sizes n_A and n_B which allows a simple hypothesis test and is ideal for a region merging algorithm. The test supports higher order models as well as vectorial data which only requires an adaption of the outlier/inlier rule in expression 2. The only parameter of the method is the width of the confidence intervals in equation 15 and 16. In this work a value of $100(1 - \alpha)\% = 95\%$ has been chosen from comparisons with other statistical hypothesis tests. The discrimination value decreases with wider confidence intervals. Figure 2 displays a numerical evaluation of equation 19. For computational purposes a polynomial approximation of the integral is used in the segmentation algorithm.

For a comparison of the MIR-method with the established KS-test, pairs of random data sets have been created. The four different set-ups for mean and variance of the two examined sets are shown in table 1. Because robust decisions on small test sets are important for a merging algorithm, sets of sizes between 10 and 100 are examined. Figure 3 displays the percentage of erroneous decisions in 10k experiments for each value pair of (n_A, n_B) in each set-up. Similarly to the chosen parameters for the MIR-test, a significance level of 95% has been chosen to determine the reference D-value for disproving the hypothesis that the two tested distributions are the same in the KS-test.

In test A random sets with the same underlying distribution have to be recognised. Both the MIR- and the KS-test result in a similar sized error which does not decrease for larger sets. For the KS-test the error depends on the chosen significance level and for the MIR-test the error can be influenced by the chosen width of the confidence intervals.

Distributions with different means have to be distinguished in test B. For the MIR the test result is strongly influenced by the choice of the confidence level of the χ in equation 2. With the applied $\chi_{p,97.5\%}$ value the KS-test performs slightly better for set sizes smaller than 15 elements. For larger sets the error is close to 0 for both methods.

Test C examines distributions with different variances and same means. Both methods respond with significantly higher error rates compared to previous tests. However, the MIR method converges much faster to smaller errors and error rates below 10% are measured for sets of 20 elements. The KS-test obtains these results only from sets of 65 elements onwards.

In the final test both means and variances are different in the underlying distributions. Both methods respond with a very low error rate, while the KS-test performs slightly better for very small sets.

The data deviations from the model are assumed to be Gaussian distributed to derive the discrimination function described in equation 18 and 19. To check whether this assumption is true for real data we determined the distribution as described in Section 2.2 and found a fairly good match with the estimated distribution. For a more quantitative test we computed the *Kolmogorov-Smirnov D* (KS-D), which is the maximum difference between two cumulative distributions [26]. Here the assumed Gaussian cumulative distribution given by the estimated parameters and the step-function cumulative distribution given by sorting the residuals of the data set are compared. A low resulting value KS-D can be seen as an indicator for a high validity of the assumption. Figure 4 displays the resulting KS-D values for ideal Gaussian distributed data and for sets obtained from real range data (ABW imagery, see Section 4.1). The range graphs are averaged KS-D values computed from regions in 15 different images. The real data has been examined using the proposed standard deviation estimator described in Section 2.2 and is compared with Rousseeuws [1, p.202] scale estimator (equation 6). The graphs show smaller KS-D values for the proposed scale estimation method underlining a better fit of the Gaussian model to the data. The graphs support the assumption of a Gaussian distribution better for increasing set sizes, but the KS-D values measured in range data are clearly higher stating that the assumption of additive Gaussian distributed noise is less valid. We observed deviations from the Gaussian model in particular in the tail area of the distribution (see figure 1). Therefore the proposed function for the discrimination value can only be seen as an approximation in the case of non-Gaussian distributed data. Our function assuming ideal Gaussian residuals converges for larger set sizes towards 1. For range data we observed, that this discrimination value is too large due to the deviation from the data's real distribution, especially in the distribution's tail area. Therefore we altered the expression for the lower limit in equation 18 to

$$b_{low} = s_{ratio}\left(\frac{t_{(n_A-1),\alpha/2}}{\sqrt{n_A}} - \alpha \cdot \chi_{p,97.5\%}\right) + \frac{t_{(n_B-1),\alpha/2}}{\sqrt{n_B}}, \qquad (20)$$

where α is a value between 0 and 1. With this change the discrimination function will converge towards lower values. Later experiments will show that the choice of α is not very crucial in the context of our algorithm.

3 Segmentation Algorithm

The region-growing algorithm works in a bottom-up way generating an irregular graph pyramid to store neighbourhood relations between regions. Each graph node represents a region in the image and contains the parameters of a data model described in Section 2.1. The initial cell tessellation is the lower of two hierarchies and its generation will be described in Section 3.1. Starting from this level, the higher primitive region (PR) hierarchy is computed by iteratively merging the cells in the level below. The merging decision was described in Section 2.3. Based on this decision rule Section 3.2 describes the generation of the PR hierarchy and Section 3.3 elucidates the iterative merging strategy leading to the final region map. Section 3.4 describes the boundary refinement methods applied to the computed region map.

3.1 Tessellation and Primitive Component Shifting

For the segmentation of 2D images the original data set is tessellated with cells of 5×5 pixels, called primitive components (PC). The initial size is a trade-off between statistical robustness, which requires larger sets for estimating the model parameters (see results in figure 3), and good precision for detecting small details, which needs small basic elements for later merging. The LMedS estimator is applied for each PC to estimate the model parameters. Initially the

PCs are overlapping by 4 pixels in each dimension. A regular tessellation is likely to contain PCs placed on a boundary of two or more regions. To avoid this the initial PCs are shifted. The estimated variance s^2 will result in large values for PCs placed on a boundary. Therefore each PC is shifted by at most two pixels in the x- and y-direction to a position resulting in a local minimum of the estimated variance. Figure 5 illustrates this step. The initial PCs are overlapping to avoid pixels not being covered by any PC after the shifting procedure. This cannot be avoided for a few pixels at the image boundary. By shifting the PCs the initial tessellation can be adapted to almost any region structure and the final segmentation will not contain the "block-structure" of the initial tessellation.

In the case of range image segmentation, the cell shift works well for step edges, while flat crease (or roof) edges do not evoke a local variance maximum to remove the cells from the boundary. For this reason we have experimented with different minimisation functions for PC shifting. One approach was to use a simple edge detector (to highlight step edges) plus the response of a local curvature estimator [28, pp. 76] (to highlight crease edges). This worked well in shifting the PCs away from both types of boundaries but was also very sensitive to inner-region noise, which resulted in over-segmented final region maps. Therefore we chose the variance as the minimisation function. Section 3.4 describes a way to recover crease edges effectively.

3.2 Primitive Region Generation

This stage of the algorithm generates a first PR hierarchy by merging the initial PCs. The stage starts by sorting the PCs according to their estimated variance. A first PR is created from the PC with the smallest variance. Next all spatially adjacent neighbours of this PC are visited recursively, becoming a member of the PR if they are inlying within the model of all of the other PCs of the PR. The recursion stops if an adjacent PC does not satisfy the merging rule. The inlier decision is found by the described MIR test. If no further adjacent PCs can be added to the current PR, a new PR is created with the next available smallest-variance PC. Again the neighbours of this cell are visited recursively. The step is repeated until every PC is associated with a PR. The sorting step at the beginning of this stage avoids over-merging since cells with a large variance tend to accept other cells more easily as inlying due to a wider inlier range.

3.3 Iterative Region Growing

This final stage of the algorithm iteratively merges the PRs until no adjacent region pair no longer satisfies the merging rule. Each iterative merging step starts by constructing a link to the neighbour, which results in a mutual satisfaction of the decision rule (equation 2) with the *highest* sum of mutual inlier ratios MIR:

$$MIR = M_{AB} + M_{BA}. \tag{21}$$

The stage starts merging the PRs after the best neighbour for merging has been determined for each PR. Each PR satisfying the merge test with one of its neighbours is linked to that neighbour, which results in the maximum MIR. However, neighbouring PRs are *not* necessarily mutually optimal, e. g. in figure 6, PR ν_6 optimises the MIR for PR ν_9 even though ν_9 does not optimise the MIR for ν_6. The links build up "chains" ending in a mutually best-neighbour pair. Each of these pairs creates a new PR in the next hierarchy. Each PR with a "best-neighbour"-link to a pair member is also incorporated into the new PR to make the algorithm converge faster. Others might be connected in later iterative steps if the merging criteria are still satisfied. In figure 6 regions ν_2 and ν_3 are a mutually best-neighbour pair and qualify for merging together with region ν_6 resulting in a new region in the next hierarchy. The same rule merges region pairs ν_7 and ν_8 with region ν_4.

This iterative merging strategy has the advantage of being independent of the order of visiting the PRs. Therefore the iterative step has been implemented in parallel using multiprocessor architectures and a multithreaded environment.

3.4 Boundary Refinement

This final stage of the algorithm refines the boundaries of the computed regions to pixel level as the minimum unit up to this stage were 5×5 pixel windows.

The applied refinement method distinguishes between *step edges* and *crease edges*. For the case of a step edge equation 2 can be applied to each single boundary pixel obtaining a robust decision whether a pixel belongs to an assigned region or to an adjacent region. In the case of a crease edge this decision can become ambiguous and pixels can be inlying within multiple adjacent regions. The probability of this case becomes larger for flat crease edges making boundary refinement difficult. However, crease edges of planar surfaces have the property that these are located exactly on the intersection line of the corresponding surfaces. Using the model parameters $\{a_{xA}, a_{yA}, a_{0A}\}$ and $\{a_{xB}, a_{yB}, a_{0B}\}$ of the adjacent regions A and B, the function of the border line can easily be computed by:

$$\text{Intersection line} \triangleq \begin{cases} y(x) = \frac{a_{xB}-a_{xA}}{a_{yA}-a_{yB}}x + \frac{a_{0B}-a_{0A}}{a_{yA}-a_{yB}} & \text{for} \quad |a_{xB}-a_{xA}| \leq |a_{yA}-a_{yB}| \\ x(y) = \frac{a_{yA}-a_{yB}}{a_{xB}-a_{xA}}y + \frac{a_{0A}-a_{0B}}{a_{xB}-a_{xA}} & \text{else.} \end{cases} \tag{22}$$

The function describes the border line computed from the regression result of the region data. Using this method to find an assigning decision for boundary pixels involves information of all pixels of both regions and therefore becomes a very robust method.

For each boundary pixel the intersection line of the two adjacent regions is calculated first. In the rare event of more than two adjacent regions this calculation is performed for each region pair. Next the perpendicular distance of the current boundary pixel from the computed line is determined. If this distance is longer than a certain threshold, the algorithm assumes the presence of a step edge and assigns the pixel using expression 2. Otherwise a crease edge is assumed and the pixel is now assigned to the region which is located on its side of the intersection line. We derived the applied threshold for the distance between the current boundary pixel and the computed line from the properties of our method. The worst case at a boundary occurs when the model of a PC is mis-estimated and the PC is assigned to the wrong region. In this case the longest possible distance between a pixel and the true border line is equal to the length of the diagonal of the PC ($= 5 \times \sqrt{2}$ pixels). Note that for the special case of irregularly sampled data, this length in range units changes according to the position of the current pixel in the image.

Finally we apply a simple 3×3 median filter to remove single outlying pixels from the region map. These outliers can be created by pixel assignments using equation 2 when the pixel value is a gross outlier. Note that the refinement is the only step working with pixel values directly. All other stages create decisions based on groups of pixels.

4 Experimental Results

The evaluation of the proposed method for range image segmentation is mainly performed following the methodology of Hoover *et al* [2]. Section 4.1 elucidates this framework briefly and Section 4.2 describes some tests which are specific to this method. Section 4.3 compares the results with other range image segmenters and Section 4.4 discusses them.

4.1 Evaluation Methods

The evaluation method involves 30 time-of-flight laser range finder images (Perceptron set) and 30 structured light scanner images (ABW set) which have been manually segmented using CAD models of the objects and several enhancement methods in order to obtain *ground truth* (GT) images. A tool objectively compares the *machine segmentation* (MS) against the corresponding GT image using a set of defined performance metrics for instances of correctly segmented, missed, and noise regions, over- and under-segmentation, and accuracy of the recovered geometry. The definitions for deciding classifications are based upon a tolerance T having a range of $0.5 < T \leq 1.0$ and reflecting the strictness of the definitions. Informally, Hoover *et al* [2] defined the metrics as follows:

correct detection: A pair of regions R_n in the GT image and R_m in the MS image are classified as an instance of correct detection if

1. at least T percent of the pixels in R_m are marked as pixels of R_n *and*
2. at least T percent of the pixels in R_n are marked as pixels of R_m.

over-segmentation: A region R_n in the GT image and a set of regions $R_{m1}, \ldots, R_{mx}$ in the MS image are classified as an instance of over-segmentation if

1. at least T percent of the pixels in each region R_{mi} are marked as pixels of R_n *and*

2. at least T percent of the pixels in R_n are marked as pixels of the union of $R_{m1}, \ldots, R_{mx}$.

under-segmentation: A set of regions $R_{n1}, \ldots, R_{nx}$ in the GT image and a region in the MS image R_m are classified as an instance of under-segmentation if

1. at least T percent of the pixels in region R_m are marked as pixels of the union of $R_{n1}, \ldots, R_{nx}$ *and*
2. at least T percent of the pixels in R_{ni} are marked as pixels of R_m.

missed: A region *in the GT image* not being found as an instance of correct detection, over-segmentation or under-segmentation is classified as missed.

noise: A region *in the MS image* not being found as an instance of correct detection, over-segmentation or under-segmentation is classified as noise.

To measure the accuracy of the recovered geometry the segmentation algorithm has to supply the surface normals of the detected regions. The accuracy is determined by measuring the absolute difference of each surface normal and its corresponding normal in the ground truth image.

4.2 Experimental Results

To compute the discrimination value we applied values of 0.63 for the ABW imagery and 0.5 for the Perceptron data for the constant α in equation 15. Using these values we obtained the maximum performance according to the described error metrics.

In Section 2.2 we described the LMedS estimation method having a high break-down point in the presence of noise. However, the major drawback of this method is its high computational demand. So a key question might be whether this method is really necessary and whether a simple estimation method can obtain the same result. This question arises especially because the proposed strategy grows regions very carefully merging only the most qualified regions in each iterative step. To test this we have replaced the LMedS estimator by a *least mean* (LM) estimator while the rest of the algorithm was left untouched. The first thing we noticed applying both segmenters to the ABW and Perceptron test sets was a significant drop of the computing time: the LMedS based method had an average run time of around 12 minutes on a Sun Ultra-2, while the LM based method finished after just 6 minutes.

Looking at the obtained segmentations we observed two categories of results. For images with a very low level of noise the LM based method worked very well and there was no significant difference compared to the LMedS based results. Figure 7 displays an example. We found 5 of the 30 ABW images and 8 of the 30 Perceptron images, where the LM based method worked very well. The remaining images suffered seriously from under-segmentation due to over-estimation of the variance or break-downs of the LM-estimation method in noisy image regions. Figure 8 displays an example, where the LM based method failed.

Section 2.3 derived a theoretical discrimination *function* for the mutual inlier ratio. Instead of a function, other researchers have used a constant value (for example [11]). To examine which solution performs better we replaced the function by a constant value between 0.2 and 0.95 and measured the average number of correctly detected instances in the 30 ABW test images. Figure 9 shows also the results for over- and under-segmented instances for a compare tool tolerance of 80%. As expected the over-segmentation graph rises with the discrimination value while the under-segmentation decreases with rising discrimination value. The correctly detected regions graph shows surprisingly high results even for very low discrimination values around 0.4, where we expected lower values due to over-segmentation. These high values are a result of our merging strategy. In the early stages, regions are small invoking a high uncertainty on the merge decision. Our strategy merges only the most qualified region pairs in each iterative step and their inlier ratio is in general very close to 0.9. In the later stages the regions have grown and the inlier ratios for "bad" merges decrease significantly. For this reason the discrimination value becomes less crucial and even badly chosen constant values achieve reasonable results. However, using our derived function we detect more regions correctly. The difference is not too significant, which justifies constant values in applications where time restrictions do not allow the computation of a polynomial approximation for the discrimination function.

4.3 Comparison with Existing Methods

Following the comparison framework according to Hoover *et.al.* we compared our algorithm with approaches of the Universities of South Florida (USF), Bern (UB), Edinburgh (UE) and Washington State University (WSF). The comparison tool computed average values of the described performance metrics (Section 4.1) in 30 ABW range images and 30 Perceptron range images. The results for different compare tool tolerances are displayed in figures 10 to 15. Tables 2 and 3 show the results on all metrics at 80% compare tolerance.

Figure 10 shows the average numbers of correctly detected regions for both sets. On the ABW imagery our segmenter scores 89% at the weakest tolerance 51% and 88% at a tolerance of 80%. Similar to the other segmenters the score drops significantly for tolerances above 90%. Overall our score is among the top two in this set. In the Perceptron test set our algorithm returns significantly higher values than the other segmenters. At a compare tool tolerance of 80% we obtained a score of 77% while the best reference algorithm (UE) scored 68% at this tolerance level. The performance difference between the ABW set and the Perceptron images at this compare tool tolerance is only 11% while the best reference algorithm's performance drops 20%.

The scores for the error metrics are displayed in figures 11 to 15. Figure 11 displays the average number of over-segmented regions in both test sets. The ABW graph shows values around 0.4 up to compare tool tolerances of 90% and is well within the performances of the reference algorithms. The Perceptron graph displays high over-segmentation values for low compare tool tolerances, but obtains the best results for tolerances of 70% and higher. The graphs for under-segmentation display a performance within the range of the reference algorithms. Although the proposed method obtains the best result for missed regions in figure 14 there is still a need for improvement. This error type and the noise error are, especially for higher tolerances, considerably higher than the errors due to over- and under-segmentation.

One of the conclusions of [2] is the need for improvement of the boundary refinement of segmented regions. We compared edge pixels in the machine segmentations and the ground truth images to measure the precision of the detected boundaries. A pixel is assumed to be an edge pixel whenever one of its neighbours has a different label. We classified edge pixels into correct, missed and noise similar to the described metrics for regions. Table 4 displays the average results for the five algorithms in 30 ABW and 30 Perceptron test images. The values are given as percentages relative to the total number of edge pixels in the ground truth image. Both tables show disappointing results for all five methods and offer scope for improvement. For our method this result and the obtained accuracy of the recovered geometry parameters are closely connected due to the the applied boundary refinement method. An improved parameter estimation will result in more precisely detected boundaries as the border lines of crease edges are computed from the region model parameters.

The visual comparison in figures 16 and 17 show color coded evaluation results of two range images at a compare tool tolerance of 80%. We chose to show these images to allow a comparison with the results shown in [2]. Note that multiple classifications are possible for each region [2] which cannot be displayed in this way. However, the color display supports the detection of possible errors of the segmenters better than for example the commonly used edge image of the segmentation.

4.4 Discussion

A well performing algorithm obtains a high value of correctly detected regions and low scores in the error metrics for under-segmentation, over-segmentation, missed and noise instances. Our algorithm returns a high score for correctly detected regions in the ABW set and the highest score in the “noisier” Perceptron set. The segmentation performance does not drop as significantly as the reference algorithms’ due to the application of robust regression techniques.

Regarding the error metrics for over- and under-segmentation our algorithm performed reasonably well. The over-segmentation error in the ABW set was mainly caused by a detection of small regions close to or within shaded regions (eg. figure 16). The application of a minimum region threshold can help to tackle this problem, but unfortunately this will result in an increased number of missed small regions. Such a threshold needs adaption to different data sets and extends the tuning process.

The score for the precision of the estimated region parameters is within the range of the reference algorithms. The application of the re-weighted LMedS estimator does not improve the results significantly. For our method this score is particularly important since we use the estimated geometry parameters in the refinement step to improve the boundaries. Mis-estimated parameters can shift the computed intersection lines. Hoover *et al* [2] reported, and our observations supported, that the majority of mis-classified regions have a size of below 1000 pixels. The compare tool counts correctly classified pixels and very often, in particular for small regions, a mis-classification is only a matter of a few pixels, where the human subjective observer would expect a classification as a correctly detected region. Therefore mis-estimated parameters affect the classification of small regions. The trade-off that has to be solved when computing the parameters is to avoid involving precision harming outliers into the computation while the number of involved "good" samples needs to be maximised. To improve the precision of the estimated parameters we apply two iterative reweighting steps to the obtained LMedS estimate. An alternative estimator is the *least trimmed squares* estimator [1] which uses 50% of the samples resulting in the smallest squared residuals to compute the model. This estimator has been improved by Lee *et al* [18] introducing a criterion to adapt the quantile of "good" samples. However, both estimators are computational more expensive than the LMedS estimator. Since the graphs for the obtained average precision of the geometry parameters are very close for all five methods the question of a natural limit of the accuracy of this measure arises and whether these computational efforts will pay off.

The average processing time of our algorithm was 12 min 18 sec (with a standard deviation of 105 sec) on a Sun Ultra, which is faster than the USF algorithm, but slower than the UE, WSU and UB algorithm. Our algorithm is computational demanding due to multiple sorting steps in the estimation procedure. To improve the processing time we implemented the algorithm in a multi-threaded environment. This parallel implementation of the algorithm running on two CPUs needs 35% less time than the serial version. Rousseeuw reports a parallel implementation of the LMedS on multiple CPUs making the estimation procedure almost as quick as least squares estimation [1, p. 201]. Another possible improvement is the application of special hardware optimized for sorting and median filtering [29] and the processing of the data in integer format instead of the applied float description. The code is currently not very portable due to the parallel implementation but we plan to make it available within the Image Understanding Environment (IUE) project [2].

5 Conclusions

In this work we described a statistical method to compare polynomial models with additive Gaussian distributed noise. The method is based on the robust *mutual inlier ratio* and takes into account the fact that the parameter estimation in small data sets results in a higher uncertainty. We compared the MIR method with established methods using ideal Gaussian data and observed a superior performance in our tests. The LMedS estimator and a novel method to estimate the model variance have been applied to estimate the model parameters robustly in real data containing outliers. This variance estimation results in better estimates as it uses only the best inliers of the estimated model to determine its result. The method therefore avoids the detrimental influence of outliers or asymmetricly distributed data.

We proposed a region growing technique which uses the robust test to compare and merge adjacent regions. A shifting technique avoids placing the initial seeds on multiple regions and adapts the initial tessellation to the image content. The result of an iterative merging stage is finally processed by a boundary refinement stage. Apart from the refinement stage the algorithm is not limited to the planar model of the range image segmenter in the evaluation. It can also be applied to segment 3D voxel data using a constant model [23, 24] or to higher order vectorial data (for example optical flow problems) with minor changes in the mathematical theory (scalar values have to be replaced by vectors and the scalar variance becomes a covariance matrix). An example for the application of the LMeds to vectorial data is the *Minimum-Volume-Estimator* (MVE) [30, 1]. The projection pursuit based estimation becomes computationally more expensive for higher order models. However, using an increasing number of degrees of freedom can introduce under-segmentation artifacts. Lee *et.al.* [18] proposed a higher order approximation based on an initial planar segmentation to overcome this problem.

[2]More information about the IUE project can be found via *http://www.aai.com/AAI/IUE/IUE.html*

The algorithm avoids information obtained from single pixels and derives decisions only from groups of pixels. Therefore decisions become very robust which is one reason for the significantly better performance of our technique in the noisier Perceptron set compared to the reference methods.

The very small number of thresholds or other tuning parameters makes the tuning process very fast and simple. We started the evaluation with the ABW set. After several debugging-testing-improving cycles we found the final version for the segmentation technique. For the evaluation on the Perceptron imagery we only had to adapt the α of the discrimination function.

A Expression for Discrimination Value

This section describes the necessary steps to derive equation 18. By substituting $y = \frac{x-\bar{X}_B}{s_B}$ and replacing $\left(\bar{X}_A - \bar{X}_B\right)$ with $\left(t_{(n_A-1),\alpha/2}\frac{s_A}{\sqrt{n_A}} + t_{(n_B-1),\alpha/2}\frac{s_B}{\sqrt{n_B}}\right)$ (derived from equation 15) in the first term of equation 18, we obtain:

$$\overline{I}_{AB}(s_A, s_B, n_A, n_B) = \frac{1}{\sqrt{2\pi}} \int\limits_{\frac{s_A}{s_B}\left(\frac{t_{(n_A-1),\alpha/2}}{\sqrt{n_A}} - \chi_{p,97.5\%}\right) + \frac{t_{(n_B-1),\alpha/2}}{\sqrt{n_B}}}^{\chi_{p,97.5\%}} e^{-\frac{y^2}{2}}\, dy. \tag{23}$$

Expression 23 includes the yet unknown term $\frac{s_A}{s_B}$. This ratio is well known from *F-statistics* where the squared value has the F-distribution. So far, expression 23 includes the worst-case estimate for the means $\bar{X}_i$. The discrimination value which discriminates even the worst-case variance estimates in the range of expression 16 is $\min\limits_{s_A,s_B} M_{\text{DISC}}(s_A, s_B, n_A, n_B)$. The result is obvious in equation 23 and 16 as s_A has to be small and s_B has to be large. Using the minimum and maximum values of the confidence interval in expression 16 we can replace $\frac{s_A}{s_B}$ in equation 23 by

$$\frac{s_A}{s_B} = \frac{\chi_{(n_A-1),1-\alpha/2} \cdot \sqrt{n_B-1}}{\chi_{(n_B-1),\alpha/2} \cdot \sqrt{n_A-1}}. \tag{24}$$

Equation 18 assumes the existence of an *overlapping* inlier area of the two compared distributions. Under the given assumptions ($\bar{X}_A = \mu + d_A$ and $\bar{X}_B = \mu - d_B$) this constraint can be expressed as

$$\bar{X}_B - s_B\chi_{p,97.5\%} \quad \leq \quad \bar{X}_A + s_A\chi_{p,97.5\%} \quad \leq \quad \bar{X}_B + s_B\chi_{p,97.5\%}. \tag{25}$$

By inserting $\bar{X}_A = \mu + d_A$ and $\bar{X}_B = \mu - d_B$ and equation 15 this constraint can be rewritten as

$$\frac{\chi_{p,97.5\%} - \sqrt{n_B^{-1}} \cdot t_{(n_B-1),\alpha/2}}{\sqrt{n_A^{-1}} \cdot t_{(n_A-1),\alpha/2} - \chi_{p,97.5\%}} \quad \leq \quad \frac{s_A}{s_B} \quad \leq \quad \frac{\chi_{p,97.5\%} + \sqrt{n_B^{-1}} \cdot t_{(n_B-1),\alpha/2}}{\chi_{p,97.5\%} - \sqrt{n_A^{-1}} \cdot t_{(n_A-1),\alpha/2}}. \tag{26}$$

The lower limit in expression 26 converges for $\lim n_1, n_2 \to +\infty$ towards -1 and is always smaller than 0. The upper limit converges from the *right* hand side towards 1 while the computed ratio in equation 24 converges from the *left* hand side towards 1. Therefore the constraint assuming an overlapping inlier area for the compared distributions in expression 26 and 25 is always satisfied and does not disagree with the assumption in equation 16. We finally obtain

$$\begin{aligned} I_{AB}(n_A, n_B) &= \frac{1}{\sqrt{2\pi}} \int\limits_{b_{low}}^{\chi_{p,97.5\%}} e^{-\frac{x^2}{2}} dx, \\ \text{where} \quad b_{low} &= s_{ratio}\left(\tfrac{t_{(n_A-1),\alpha/2}}{\sqrt{n_A}} - \chi_{p,97.5\%}\right) + \tfrac{t_{(n_B-1),\alpha/2}}{\sqrt{n_B}} \\ \text{and} \quad s_{ratio} &= \tfrac{\sqrt{n_B-1} \cdot \chi_{(n_A-1),1-\alpha/2}}{\sqrt{n_A-1} \cdot \chi_{(n_B-1),\alpha/2}}. \end{aligned} \tag{27}$$

References

[1] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.

[2] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher, “An experimental comparison of range image segmentation algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 673–689, July 1996.

[3] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, New York: Prentice Hall, 1982.

[4] R. J. Schalkoff, *Digital Image Processing and Computer Vision*. Singapore: Wiley, 1989.

[5] M. A. Wani and B. G. Batchelor, "Edge-region-based segmentation of range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 314–319, 1994.

[6] E. Trucco and R. B. Fisher, "Experiments in curvature-based segmentation of range data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 177–182, 1995.

[7] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 1. Reading, Mass.: Addison-Wesley, 1992.

[8] S. C. Zhu and A. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 884–900, Sept. 1996.

[9] S. W. Zucker, "Region growing: Childhood and adolescence," *Computer Graphics and Image Processing*, vol. 5, pp. 382–399, 1976.

[10] R. Adams and L. Bishof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641–647, June 1994.

[11] P. Meer, D. Mintz, and A. Rosenfeld, "Least median of squares based robust analysis of image structure," (Pittsburgh, P. A.), pp. 231–254, DARPA, 1990.

[12] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 167–192, 1988.

[13] U. Neisser, *Cognitive Psychology*. New York: Appleton-Century-Crofts, 1967.

[14] R. L. Gregory, *The Intelligent Eye*. New York: McGraw-Hill, 1970.

[15] P. Meer, D. Mintz, and A. Rosenfeld, "Robust regression methods for computer vision: A review," *International Journal of Computer Vision*, vol. 6, no. 1, pp. 59–70, 1991.

[16] C. V. Stewart, "MINPRAN: A new robust estimator for computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 925–938, October 1995.

[17] C. V. Stewart, "Bias in robust regression caused by discontinuities and multiple structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 816–833, August 1997.

[18] K.-M. Lee, P. Meer, and R.-H. Park, "Robust adaptive segmentation of range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 200–205, 1998.

[19] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, Mass.: MIT Press, 1987.

[20] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 712–741, 1984.

[21] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision*, vol. 19, no. 1, 1996.

[22] F. Moscheni and F. Dufaux, "Region merging based on robust statistical testing," in *SPIE Proc. Visual Communications and Image Processing*, (Orlando, Florida), SPIE, Mar. 1996.

[23] K. Köster and M. Spann, "A robust approach to unsupervised segmentation of seismic data sets," in *Vision Interface*, (Vancouver), IAPR, June 1998.

[24] K. Köster and M. Spann, "A system for seismic data processing," in *European Signal Processing Conference*, (Rhodes), EURASIP, September 1998.

[25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, Mass.: Addison-Wesley, 1992.

[26] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, 2nd ed., 1995.

[27] L. L. Lapin, *Probability and statistics for modern engineering*. Boston, MA: PWS-Kent, 2nd ed., 1990.

[28] P. J. Besl, *Surfaces in Range Image Understanding*. New York: Springer Verlag, 1988.

[29] D. S. Richards, "VLSI median filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 1, pp. 145–153, 1990.

[30] J. M. Jolion, P. Meer, and S. Bataouche, "Robust clustering with application in computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, August 1991.

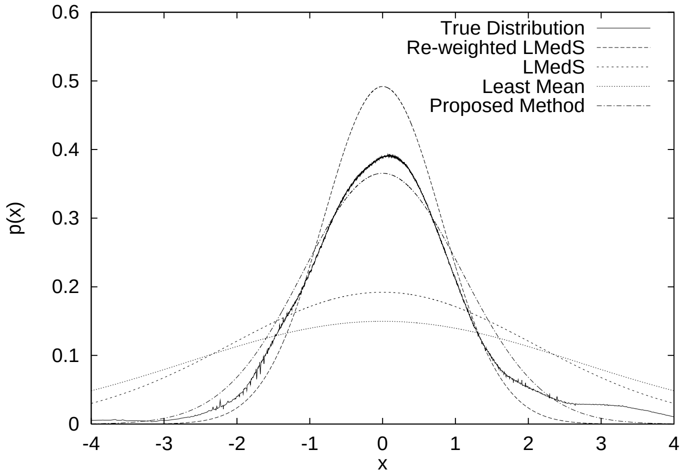


Figure 1: Comparison of the true distribution of the model residuals with the assumed Gaussian distribution in real range data. The scale parameter σ of the Gaussian model has been estimated by different methods.

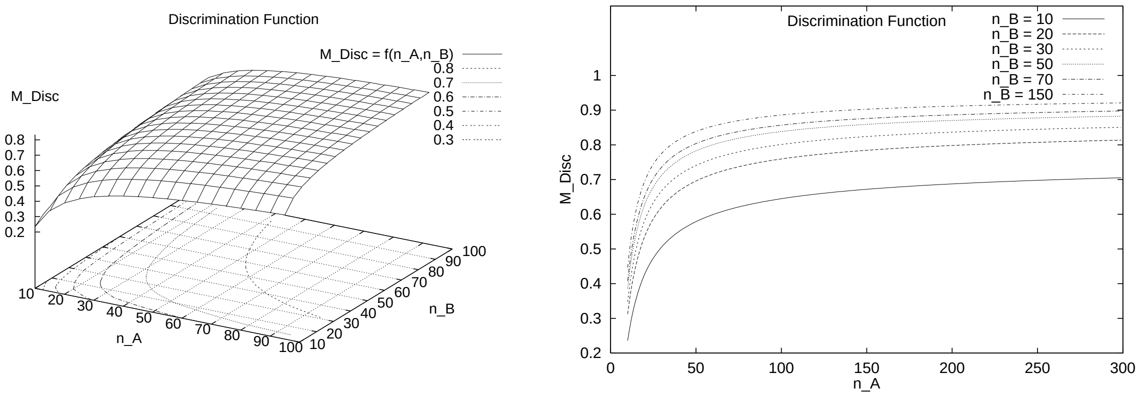
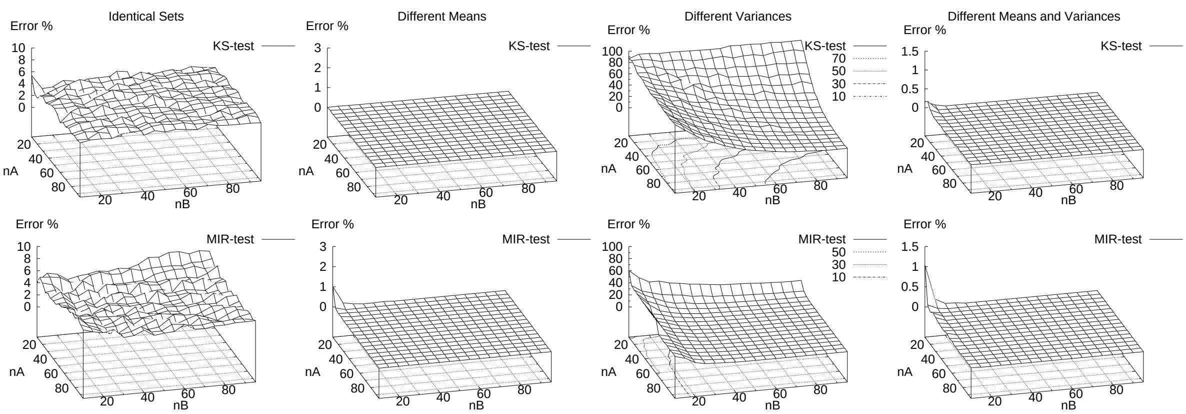


Figure 2: Numerical evaluation of discrimination function $M_{\text{DISC},97.5\%}(n_A, n_B)$.



Test A Test B Test C Test D

Figure 3: Comparison of KS-test and MIR-test. Error of classification when examining sets with identical and different underlying distribution.

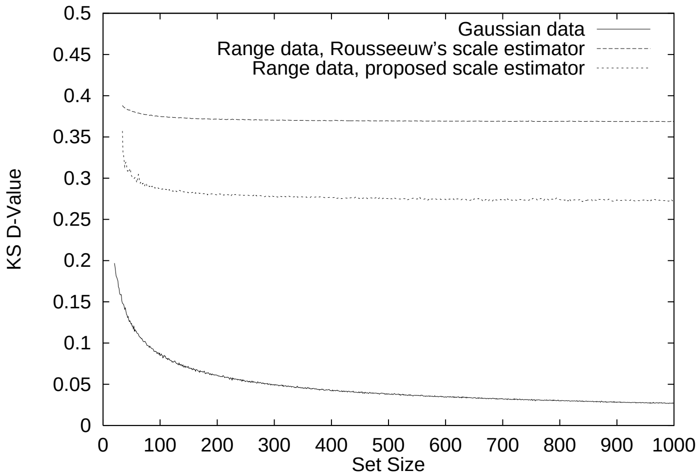


Figure 4: KS-D value measured for different set sizes in ideal Gaussian distributed data sets and real range data sets.

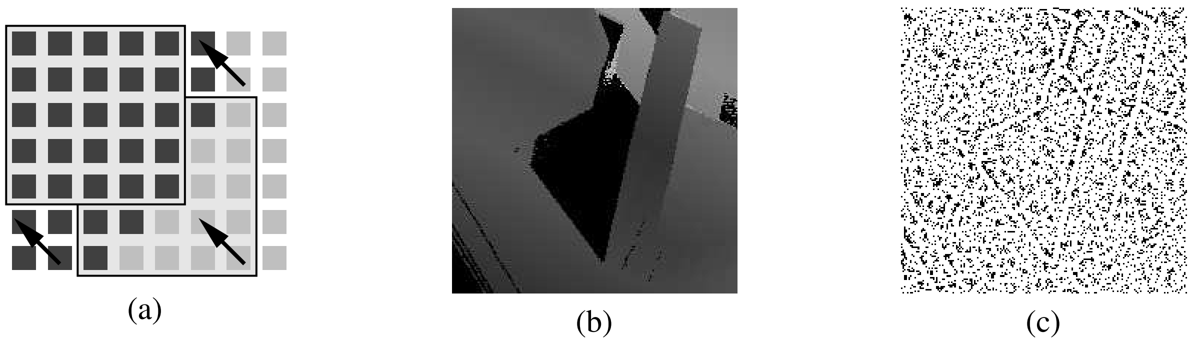


Figure 5: Shifting of the initial cells. The cells, marked by a frame in figure (a), are shifted to a position resulting in a local minimum of the estimated variance. Figure (b) shows a part of a range image and figure (c) the shift result, where the centre pixel of each 5x5 cell has been marked by a black point, clearly the cells shift away from boundaries.

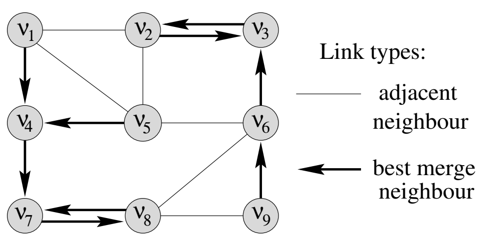
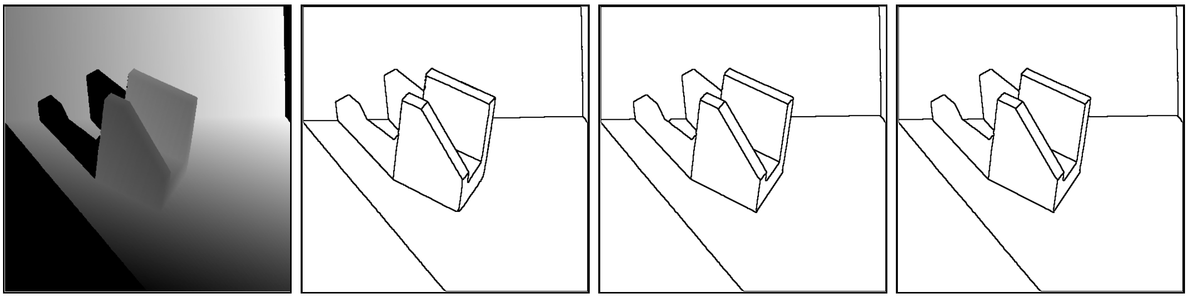
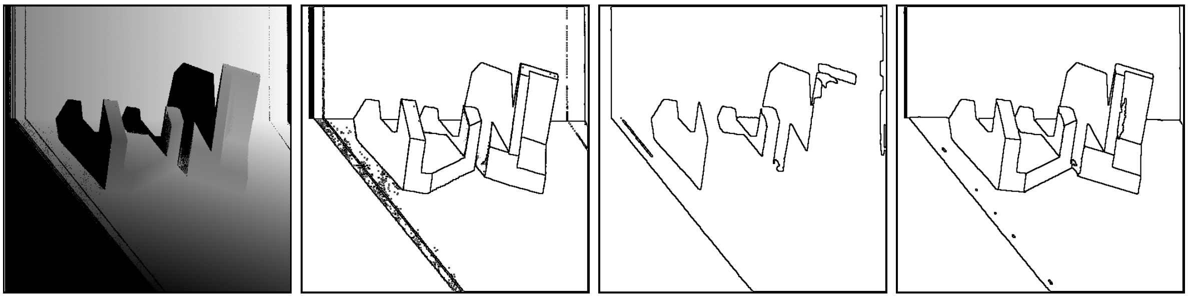


Figure 6: The regions ν_i having a link to their neighbour being most qualified for merging.



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

Figure 7: ABW test image 21 (a) with ground truth image (b) and the segmentation results of a *least squares*- (c) and a *least median of squares*- based method (d).



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

Figure 8: ABW test image 14 (a) with ground truth image (b) and the segmentation results of a *least squares*- (c) and a *least median of squares*- based method (d).

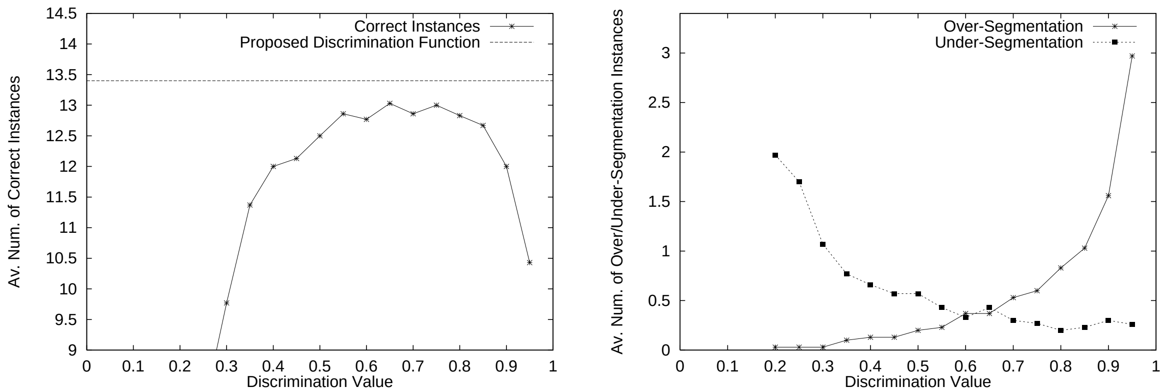


Figure 9: Average results of correctly detected instances, over- and under-segmentation in 30 ABW test images for constant discrimination values between 0.2 and 0.95.

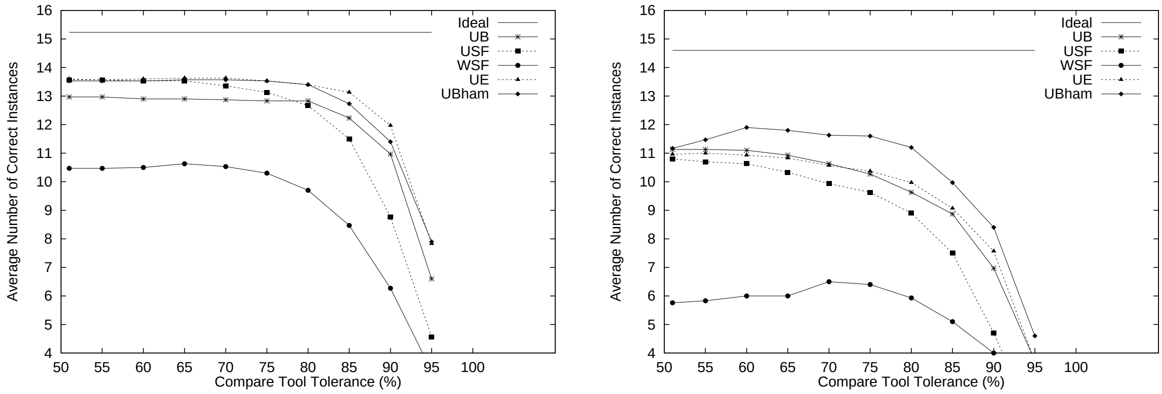


Figure 10: Average number of correctly detected regions in 30 ABW (left) and 30 Perceptron test images (right).

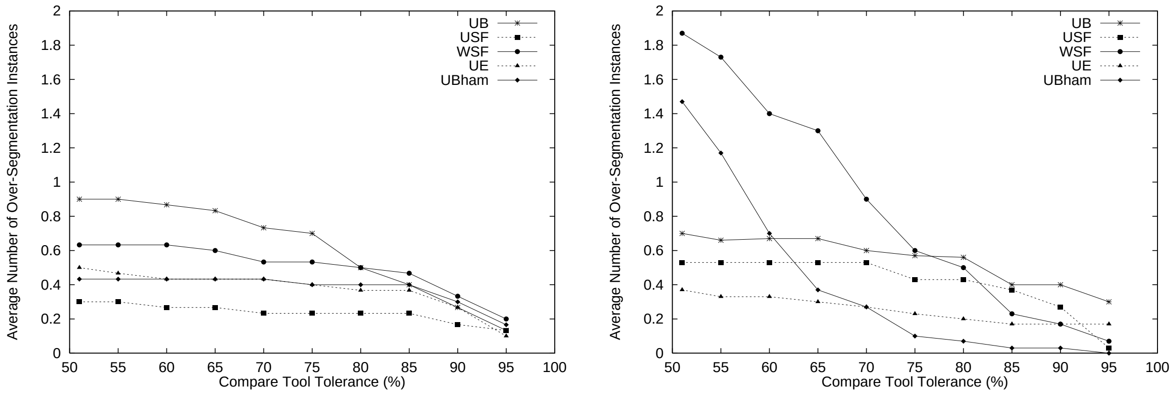


Figure 11: Average number of over-segmented regions in 30 ABW (left) and 30 Perceptron test images (right).

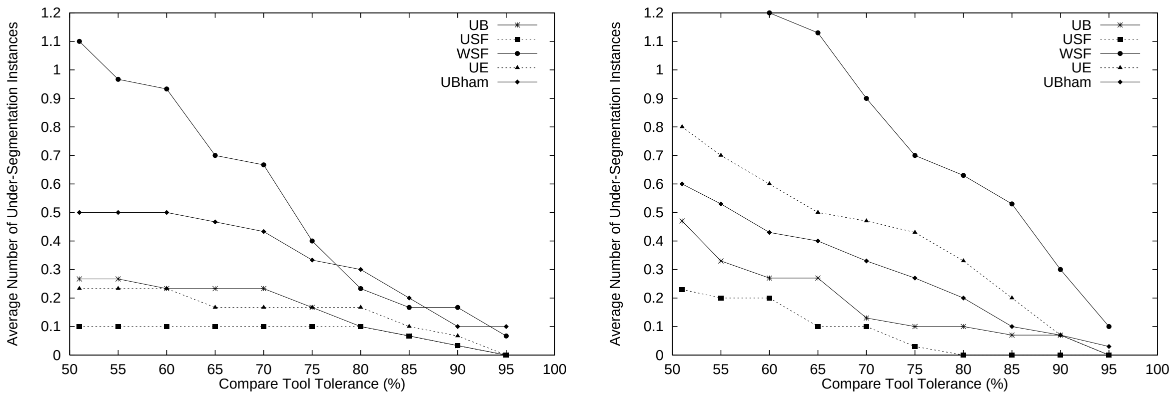


Figure 12: Average number of under-segmented regions in 30 ABW (left) and 30 Perceptron test images (right).

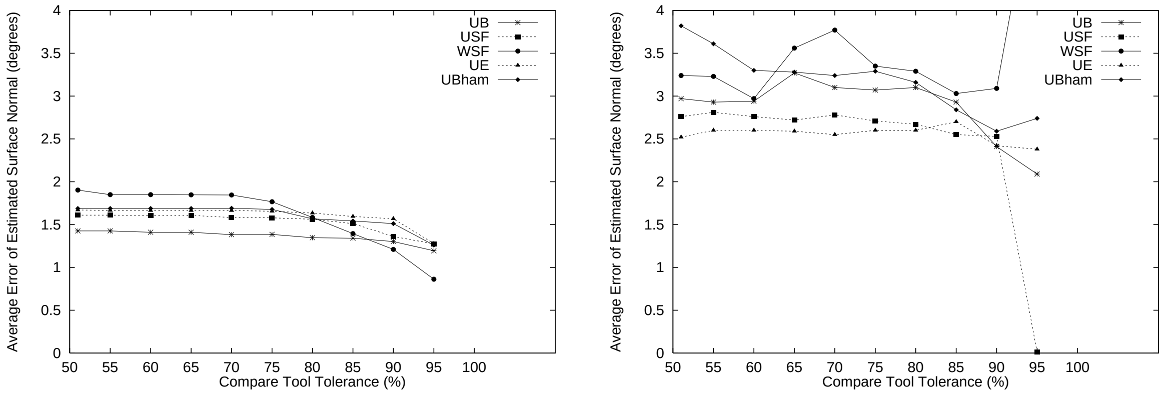


Figure 13: Average error of estimated region normal in degrees in 30 ABW (left) and 30 Perceptron test images (right).

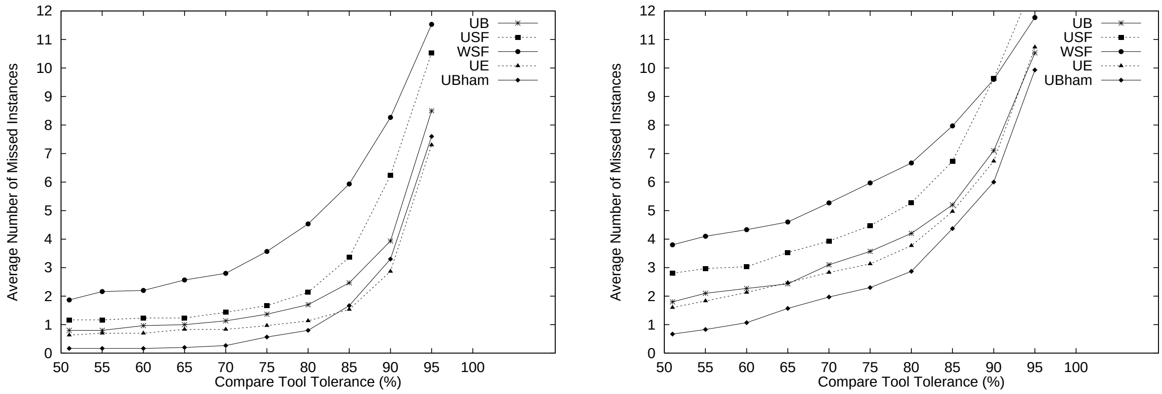


Figure 14: Average number of missed regions in 30 ABW (left) and 30 Perceptron test images (right).

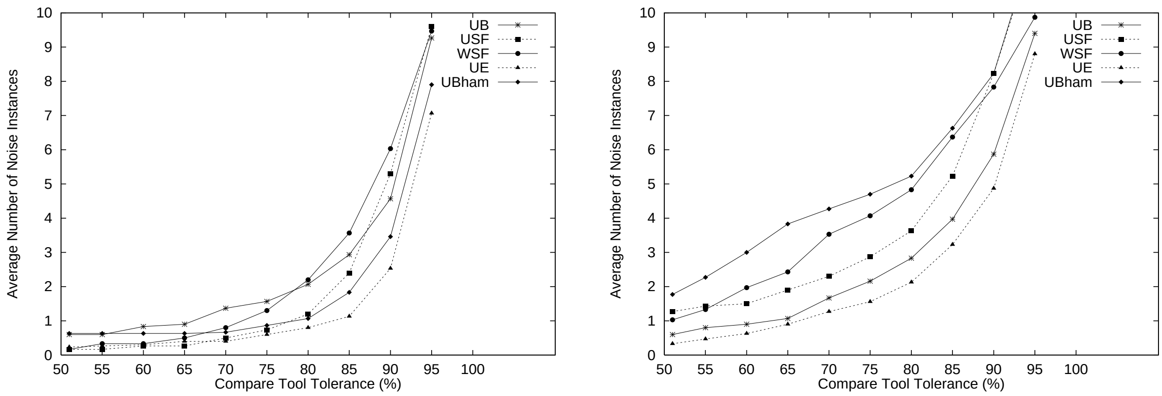


Figure 15: Average number of noise regions in 30 ABW (left) and 30 Perceptron test images (right).

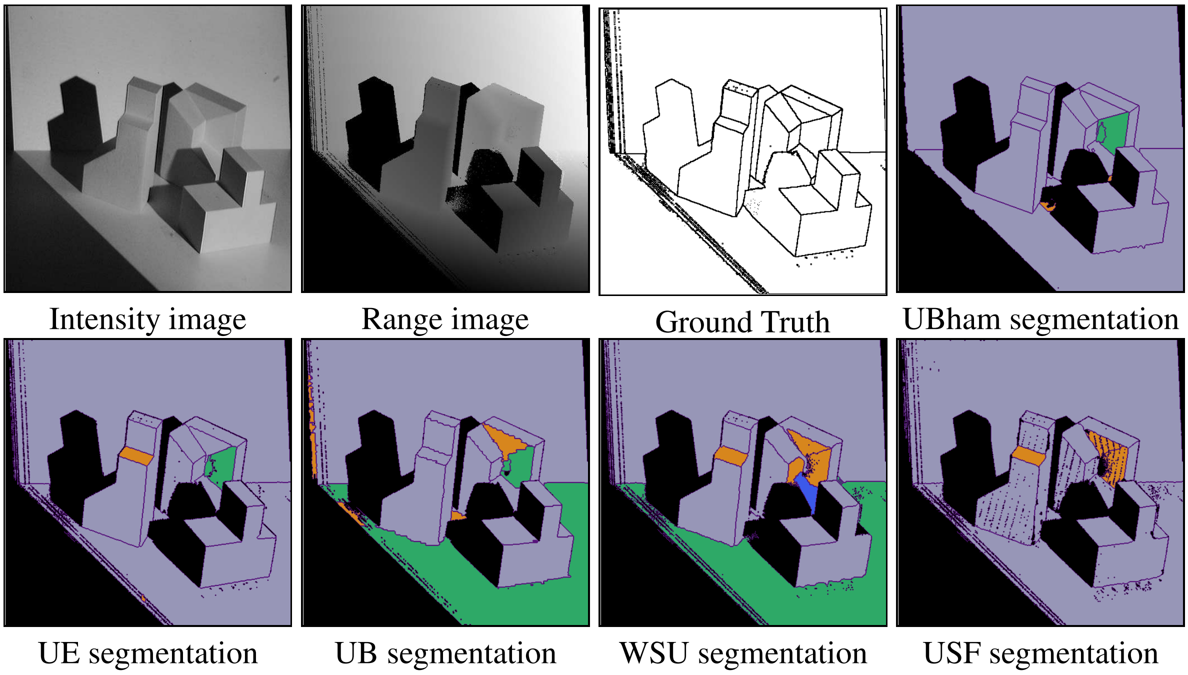


Figure 16: Comparison of the segmentation results of ABW test image 8 at 80% compare tool tolerance. Displayed are non-surface regions (shadow, noise etc.) (black colored), over-merged (blue), over-segmented (green), noise (yellow) and correctly detected regions (grey).

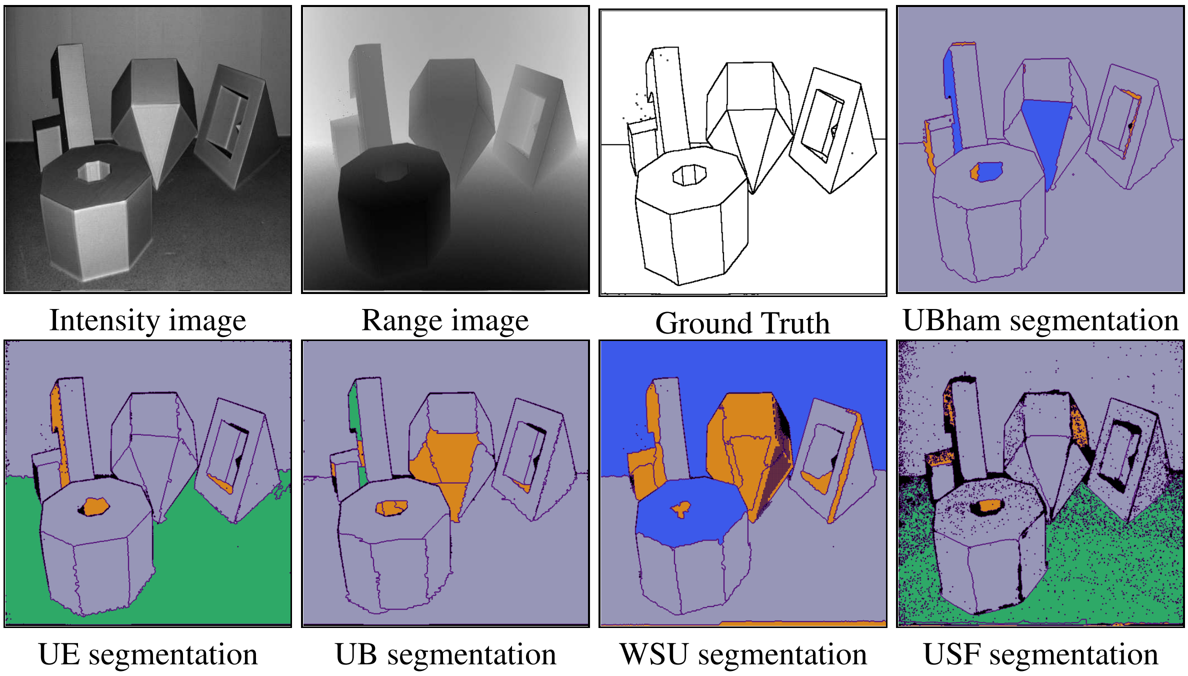


Figure 17: Comparison of the segmentation results of the Perceptron test image 26 at 80% compare tool tolerance. Displayed are non-surface regions (shadow, noise etc.) (black colored), over-merged (blue), over-segmented (green), noise (yellow) and correctly detected regions (grey).

Test	Set A	Set B	Error counted if
A	$N(10.0, 10.0)$	$N(10.0, 10.0)$	distribution A $\neq$ distribution B
B	$N(5.0, 10.0)$	$N(15.0, 10.0)$	distribution A $=$ distribution B
C	$N(10.0, 5.0)$	$N(10.0, 30.0)$	distribution A $=$ distribution B
D	$N(5.0, 30.0)$	$N(15.0, 5.0)$	distribution A $=$ distribution B

Table 1: Four tests with different underlying distributions $N(\mu, \sigma^2)$ for the examined sets.

research group	GT regions	correct detection	angle diff. (std. dev.)	over-segmentation	under-segmentation	missed	noise
USF	15.2	12.7	1.6° (0.8)	0.2	0.1	2.1	1.2
WSU	15.2	9.7	1.6° (0.7)	0.5	0.2	4.5	2.2
UB	15.2	12.8	1.3° (0.8)	0.5	0.1	1.7	2.1
UE	15.2	13.4	1.6° (0.8)	0.4	0.2	1.1	0.8
UBham	15.2	13.4	1.6° (0.9)	0.4	0.3	0.8	1.1

Table 2: Average results of segmenters on ABW test set at 80% compare tolerance. Units are instances of region-mappings between ground truth and machine-produced segmentations.

research group	GT regions	correct detection	angle diff. (std. dev.)	over-segmentation	under-segmentation	missed	noise
USF	14.6	8.9	2.7° (1.8)	0.4	0.0	5.3	3.6
WSU	14.6	5.9	3.3° (1.6)	0.5	0.6	6.7	4.8
UB	14.6	9.6	3.1° (1.7)	0.6	0.1	4.2	2.8
UE	14.6	9.7	2.6° (1.5)	0.2	0.3	3.8	2.1
UBham	14.6	11.2	3.2° (2.5)	0.1	0.2	2.9	5.2

Table 3: Average results of segmenters on Perceptron test set at 80% compare tolerance. Units are instances of region-mappings between ground truth and machine-produced segmentations.

research group	correct detection	missed	noise
USF	39.69%	60.32%	35.49%
WSU	40.90%	59.10%	45.74%
UB	58.80%	41.20%	33.97%
UE	55.72%	44.28%	27.74%
UBham	65.03%	34.97%	31.76%

research group	correct detection	missed	noise
USF	12.68%	87.32%	27.44%
WSU	33.37%	66.63%	66.74%
UB	30.57%	69.43%	47.80%
UE	27.68%	72.32%	48.31%
UBham	53.30%	46.70%	57.00%

Table 4: Percentages of correctly detected, missed and noise boundary pixels as a measure of precision at boundaries in the ABW test set (left) and the Perceptron test set (right).

Authors' Biographies

Klaus Köster received his master's degree in electrical engineering/communications from the University of Hannover, Germany, in 1995. He is currently a Ph.D student in the Department of Electronic and Electrical Engineering at the University of Birmingham, UK. His main research interests are image segmentation, robust statistics, pattern recognition and signal processing.

Michael Spann obtained a First Class Honours B.Sc in Physics and M.Sc in Computer Science from the University of Manchester in 1979 and 1981, respectively. From 1982 to 1985 he was engaged in research in image processing at Aston University and obtained a Ph.D in 1985 for his work on image segmentation. From 1985 to 1990 he was a lecturer in the Electronic and Systems Engineering School at Coventry University. In 1988 he was a visiting research fellow at the Signal Processing Laboratory, EPFL, Lausanne, Switzerland. Finally, in 1990 he was appointed as lecturer in the Department of Electronic and Electrical Engineering at the University of Birmingham where his research interests are in image processing and computer vision. Dr Spann has authored or co-authored around 30 papers in these subjects.