Automatic Parameter Control for Experimental Evaluation of Vision Systems

Guido Appenzeller
IPR, University of Karlsruhe, Kaiserstr. 12,
76128 Karlsruhe, Germany
EMail: `appenz@imag.fr`

James L. Crowley
IMAG-PRIMA, 46 Ave Félix Viallet
38031 Grenoble, France
EMail: `jlc@imag.fr`

Abstract

The evaluation of vision systems is not possible without parameter control as incorrect parameter adjustment can cause performance losses that are more significant than the effect of the component being tested. In this paper we present a model for the contour extraction process and derive from it a method for automatic parameter control. The method works with arbitrary edge detectors, edge types and noise types. Experimental data on the methods performance and its accuracy is given.

1 Introduction

The science of computer vision today is characterized by a great variety of techniques for which few comparative and even less quantitative analysis exists. In recent years, a trend towards more quantitative performance analysis of vision processes is visible [1] [2]. However, much progress is to be made.

One of the main reasons for the absence of performance evaluation is that it requires a valid evaluation criteria and correct parameter tuning of the free parameters of the vision system. On both topics little literature exists.

Evaluation in computer vision literature often consists of presenting the results of the algorithm on a few example images. Quantitative evaluation criterion are seldomly used and not standardized.

Parameter tuning in computer vision is usually done by hand. Few schemes for the automatic tuning of parameters exist. This is surprising, as automatic parameter control is necessary not only for the evaluation of vision systems but also for their industrial application.

The effect of the wrong parameter tuning can be stronger than the effect of the component that we desire to test. Evaluation in computer vision therefore requires:

- The definition of an evaluation function. This is needed to give a user the possibility to compare results as well as to define a criteria for optimal parameter tuning.
- A method to automatically adjust the free parameters of the vision system. This has to include an estimation of the error caused by imperfect parameter tuning. This is necessary in order to determine whether the obtained results were valid or if they might be due only to incorrect parameter tuning.

In this paper we present a performance evluation method for contour extraction systems. Unlike previous puplications [3] our method is applicable for arbitary edge types, edge detectors and noise types. The evaluation scheme includes a model for the qualitative prediction of the performance of a vision algorithm and the automatic tuning of parameters which can be used independently.

We first present out evaluation criteria and then review the general problem of parameter tuning. In section 4 we describe the model we use, the way we determine the parameters of the model and how a qualitative performance prediction can be derived from the model. We finally present extensive experimental data verifying the validity of the model.

2 Performance Evaluation

The performance criteria of any vision system or any component thereof can only be specified in a task specific context. We assume that the contours are used for further feature detection. Performance evaluation is actually necessary for two purposes.

First, to give the vision engineer a criteria by which he is able to compare algorithms and to decide which algorithm is suitable for a particular application. In this case the evaluation should characterize all properties of the output that might be interesting for further use.

Second, to allow parameter tuning. For the parameter tuning algorithm a scalar error criteria has to be specified which the algorithm then attempts to minimize.

For further use of the extracted chains we propose the following evaluation criteria:

- The Detection rate. The number of pixels belonging to edge contours that have been correctly identified.
- Error rate. The number of false alarm (noise that has been incorrectly detected as contour points).
- Localization. The deviation of the contour from their real position. We characetrize it by the histogram of the deviation. This is especially important if segment extraction is to be performed later.
- Breaks. Independently of the number of contour points covered the contours should show few breaks. We characterize this by the length histogram of the contours.

If we assume that it can be said for each detected edge pixel whether it is correctly detected, that is whether it belongs to a real edge or is due to noise. We can then define a probability for the correct detection of contour pixels:

$$p_{Cont} = \frac{\text{Nr. of Correctly detected Contour Pixels}}{\text{Total Nr. of Contour Pixels}} \tag{1}$$

The same is possible for wrongly detected contour pixels (false alarms):

$$p_{Noise} = \frac{\text{Nr. of wrongly Detected Noise Pixels}}{\text{Total Number of Pixels in the Image}} \tag{2}$$

In the last equation we assumed that the number of real contour pixels is small compared to the total number of pixels. The second quantity has to be corrected if strong smoothing is used with a high signal to noise ratio in the image. In this case the strong response of the edge will reduce the number of noise pixels in the vicinity of the edge.

The impact of falsely detected contours against missed real contours depends on the application. With λ specifying a weight between these two error types we define as an error criterion:

$$E = \lambda \cdot p_{Cont} - (1 - \lambda) \cdot p_{Noise} \tag{3}$$

This scalar evaluation criterion is minimized for the automatic tuning of parameters. We usually set λ to 0.5 weighting false alerts and undetected contours euqally damaging.

3 Automatic Parameter adjustment

Vision systems typically have a large number of free parameters. So far, parameter adjustment in computer vision has been mainly done manually. For the quantitative evaluation of vision systems hand tuning is not applicable. To guarantee reproducable results automatic parameter selection schemes are necessary.

Using the same parameters for a comparative analysis of a class of vision system components is not the solution. When testing several edge detectors the values of the hysteresis thresholds will have to be tuned individually. Edge detectors with stronger smoothing will produce weaker responses and require lower thresholds.

The first step for parameter tuning is to specify an evaluation criterion as we have done in section 2. The minimum of this evaluation function in the parameter space has to be found. Due to the high computational cost associated with vision systems, it takes too long to find this minimum by classical methods such as gradient descent or simmulated annealing.

The only solution to this problem seems to be to model the vision system and to use this model to predict the best settings of the parameters. Ramesh et. al. in [4] develop a model for image features, their pertubations and the image extraction process and use it for automatic parameter tuning [3]. It is therefore only applicable for this particular system. We follow a more general approach by using a model that does not make assumptions about the features, the edge detector or the noise. The model we present does not have to be precise enough to allow us to predict the results quantitatively but it must be sufficiently precise to predict the optimal parameter setting.

4 Contour Extraction Model

In this paper we are interested in contour based vision systems. These systems use an edge detection operator and a thinning process to transfor the original image into an edge image. From this edge image contour points are extracted and gatherd into contour chains. The vision process we want to model is depicted in figure 1.

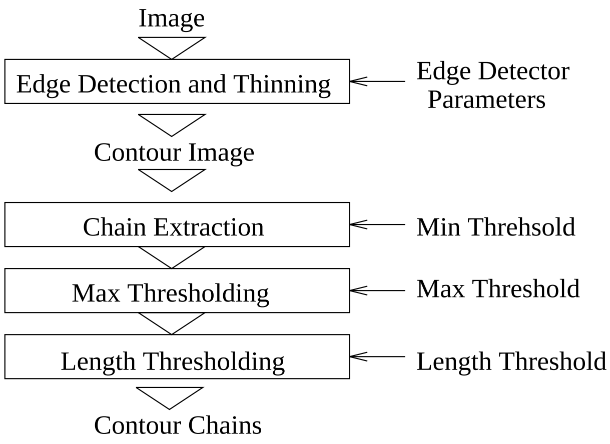

Figure 1: The Vision System

The edge detection process is fairly well understood today and models for their behavior in the presence of noise exist (e.g. [5]). This is not the case for the contour chaining process wich involves thresholding.

The model which we will use to predict the results of the contour extraction process will not include a model for the edges we want to detect and the noise in the image. Instead the parameters of our model that describe the input image are directly calculated from the contour image produced by the edge detector. This way we avoid restrictions on the type of noise and the edge detector used.

The Model for our vision process contains two things:

- A model for describing the input and output of each step of the vision system.
- A function that models the transformation from the input to the output for each step in the vision system.

Each of these should be as simple as possible while still being able to predict the final result with sufficient precision. The natural approach in this case is to start with an easy model and refine it until prediction is precise enough.

Our model for the result of each vision step is the length distribution of the edges. The steps of the contour extraction process we model are:

- Extracting and linking chains from the image from all points above a certain threshold T_{min}.
- Removal of all the chains that do not have a pixel above a certain threshold T_{max}.
- Removal of all chains with a length below a certain threshold T_{len}.

The model is depicted in figure 2. First statistics about the edge image is collected from a number of images for wich groundtruth is known. This is done separately for real contours and regionsof non-interest. From the statistics an initial length distribution is calucalted. On this initial distribution two thresholding steps are performed.

To describe the contour characteristics after the first extraction and linking step we will use their length histogram. To calculate this length histogram we present two different approches.

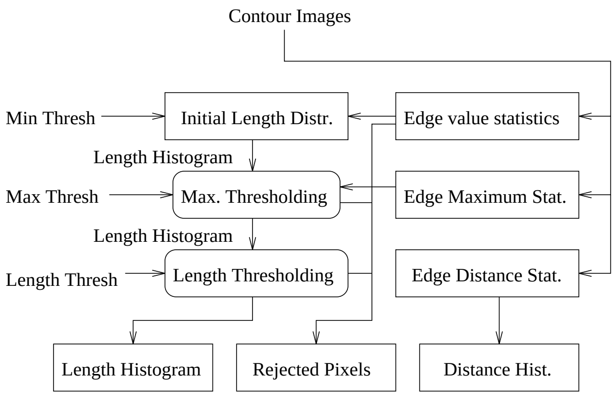

Figure 2: The Contour Extraction Model

4.1 Statistical prediction of the initial length histogram

For the distribution of edge values a simple way of predicting the length distribution of the chains is to measure the histogram of the edge pixel $H(pix)$ values along an edge. From this we can calculate the probability that an edge pixel is below our Threshold T_{min} as:

$$p_{br} = \sum_{n=0}^{T_{min}-1} H(n) \tag{4}$$

Assuming that neighboring pixel values are independent we can then calculate the length distribution as:

$$p(l) = (1 - p_{br})^l \cdot p_{br} \tag{5}$$

The problem of this model is that the pixel values of neighboring pixels on a contour are not independent. This is especially the case if a strong smoothing edge detector is used. The next best approximation is to believe that values are dependent on each other but only on neighboring values. In this case we measure a Histogram of the pixel and the neighbor Pixel $H(pix, neigbour)$. We normalize it to $H_N(pix, neighbor)$ so that the sum of each column is one. We first calculate the intial distribution.

$$p_{start}(x) = p(0, x) = \sum_n H(n, x) \quad (6)$$

From this we are able to calculate the edge value distribution at a given distance from our starting point d. We only have to sum up from T_{min} as otherwise the chain has ended.

$$p(d+1, x) = \sum_{n=T_{min}}^{\inf} H_N(x, n) \cdot p(d, n) \quad (7)$$

From this we get a length distribution:

$$p(l) = \sum_{n=0}^{T_{min}-1} p(l, n) \quad (8)$$

4.2 Direct prediction of the initial length histogram

This model needs more data then the previous ones but uses a direct method to calculate the initial length distribution as well as the maxima distribution. We first obtain the contour chain geometry as described in section 4.4. Now we follow the chains and measure for all possible thresholds the length histogram giving a joint length T_{min} histogram $H(T_{min}, l)$. The algorithm works as follows:

- Set actual length $L(T_{min})$ to zero for all T_{min}.
- Follow the chain and get actual edge pixel value v.
- For all $T_{min} < v$ add one to $H(T_{min}, L(T_{min}))$ and set $L(T_{min})$ to zero.
- For all $T_{min} \geq v$ add one to $L(T_{min})$.
- Go to step 2.

From this histogram the initial length distribution can be calculated by a simple extraction operation. As an additional benefit the maximum distribution can be calculated the same way if an additional array of $Max(T_{Min})$ is introduced. Each time a new contour is added to the histogram $H(T_{min}, l)$ the maximum value of the current contour $Max(T_{Min})$ is registerd in the maximum histogram $H_{Max}(Max(T_{Max}), L(T_{min}))$.

Figure 3 shows the predicted initial length histogram for contours fromr eal edges for both methods

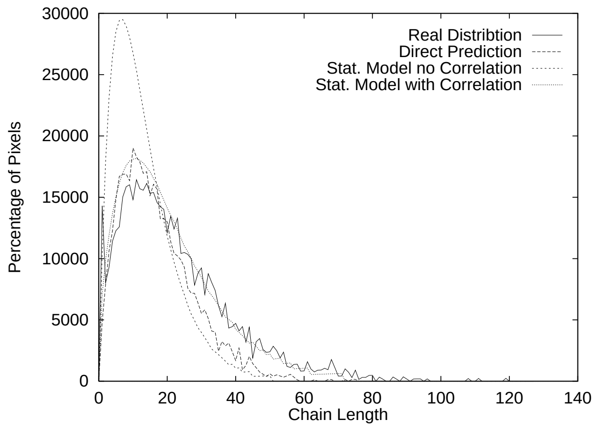

Figure 3: Comparison of Statistical vs. Direct Prediction

and the real, empirical distribution. Each bin tells the probability that a pixel is on a chain of this length (which is equal of the probability of a chain of this length multiplied with its length). Edge detection was done using the Deriche version of the Canny edge detector ($\alpha = 2.0$, $S/N = 2.0$). It can be seen that the statistical model without correlation is inferior to the statistical model wiht correlation and the direct method. The same experiment with noise gave similar performance with direct method giving slightly better results than than the statistical model wiht correlation. For further experimants we will use the direct method.

4.3 Calculating the Model Results

The maximum and length thresholding are now straightforward. It is the same for edge contours and noise contours. The probability for each length is multiplied with the probability of a segment of this length having a maximum of at least T_{Max}.

$$p(l) = p(l) \cdot \frac{\sum_{n=T_{Max}}^{\inf} H_{Max}(n, l)}{\sum_{n=0}^{\inf} H_{Max}(n, l)} \quad (9)$$

For the length thresholding the probability of all lengths below the length threshold is set to zero.

$$p(l) = \begin{cases} 0 & \text{if} \quad l < T_{Len} \\ p(l) & \text{if} \quad l \geq T_{Len} \end{cases} \quad (10)$$

To calucalte the detection rates for the noise and the contours as defined in equation 2 and 1 the final length histograms are divided by the size of the image and the number of contour pixels respectively. Summing up over the histogram bins multiplied by their corresponding length gives us the detection rates. If we denote the final contour length histograms p_N for noise and p_C for the real contours, N_I the number of total image pixels and N_C the number of real contour pixels in the image we get:

$$p_{Cont} = \frac{\sum_{l=T_{Len}}^{\text{inf}} p_C(l) \cdot l}{N_C} \quad (11)$$

$$p_{Noise} = \frac{\sum_{l=T_{Len}}^{\text{inf}} p_N(l) \cdot l}{N_I} \quad (12)$$

With this and equation 3 the final evaluation result is calculated.

4.4 Measuring Model Parameters

The parameters of our contour extraction model are several histograms. For these histograms we need the pixel values along the contours in the contour image and for lines their displacement from their real location. Additionally, the effects of breaking lines have to be taken into account. We now describe how to measure these values. The methods we use are different for noise and real contours.

For real contours we first obtain ground truth on their positions. If the noise can be reproduced this is best done by usings artifical images. Next we follow the contours. For each step along the contour the following steps are executed:

- We search the closest nonzero pixel to the actual contour point on a line orthogonal to the current contour direction. We will call this the edge pixel value from now on.
- If this pixel is not connected to the pixel found for the last contour point we assume the contour is broken.
- If their are more than one neigbouring pixel to the actual contour pixel (i.e. branches) we assume a 50 percent probability that the contour breaks.

For the edge value statististics a broken contour will always count as an edge value of zero. For Noise the main problem is getting the geometry of the resulting contours. We do this by using the contour extraction and chaining process with all thresholds set to zero. Then for each point in the contour chains the following is done.

- The value at the contour point is the edge pixel value.
- If the contour ends we assign a 50 percent chance that it is connected with the next extracted contour.

The last step is obviously wrong for very small thresholds. For them the main reason for edges ending is running into T-intersections with themselves or other already extracted contours. When the threshold is raised these intersections get fewer and we would predict to short contours. As the purpose of the thresholds is too eliminate noise, best precision is needed for higher thresholds. Therefore this precision tradeoff of high threshold against low threshold seems justified.

4.5 Independance of T_{min}, T_{max} and Displacement

So far in the model so far we have assumed independance of the minimum threshold T_{min} and the maximum threshold T_{max} as well as the distance.

The maximum distribution will initially be randomly distributed. The smoothing in the edge detector will act as a low pass filter introducing a dependency between edge pixels.

The minimum thresholding will cause all edge pixel values to be above the minimum threshold. This does not interfere with the maximum thresholding as the maximum threshold is always above the minimum threshold. As connected pixels are correlated the pixels near the ends of a contour will however often be close to the minimum threshold. For very short chains a correlation between the minimum threshold and the maximum of the chain can be expected. As very short chains will be eliminated by the length thresholding the error from this should be small.

For the distance histogram it seems possible that it is correlated with the minimum threshold for edge pixels. As weak edges have a weaker detector response, noise might be more likely to displace them. To test this we measured the correlation between edge pixel values and edge displacement with different degrees of smoothing. The result for a strong smoothing Deriche edge detector ($\alpha = 0.5$) is shown in figure 4. The true position of the edge is between 7 and 8.

As it can be seen in the image the distance and the canny value are hardly correlated. A mathematical evaluation by calculating the covariance matrix of the absolute distance vs. the edge pixel values suggest a change of about 0.2 pixels of the expectation value of the location over the full range of minimum threshold values. The error from the assumption that location and T_{min} are independent can therefore be expected to be small.

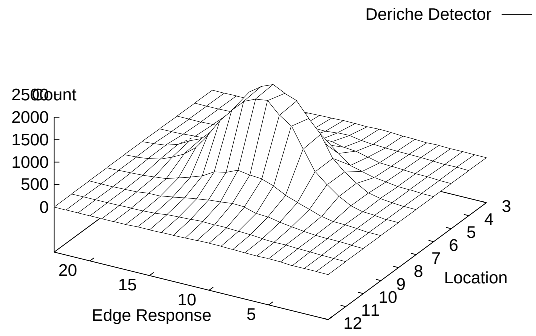

Figure 4: Correlation between Edge Value and Localization

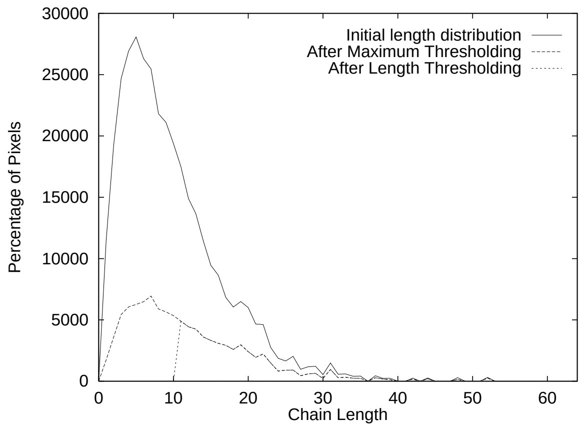

Figure 6: Extraction Stages - Model Data

5 Experimental Evaluation of the Model

In this section we want to show experimental data to confirm the models validity and to determine how precise the model is. The artifical images contained straight edges corrupted by additive guassian noise. Edge detection was done with the Deriche [6] version of the Canny edge detector ($\alpha = 2.0$).

To do this we first have a look on the models ability to predict the results of the intermediate steps of the extraction process for noise and contours separately. We then compare the error values provided by the model to the real error determined emiprically. Finally we use the model for its actual purpose and compare its ability to tune parameters to static parameter settings.

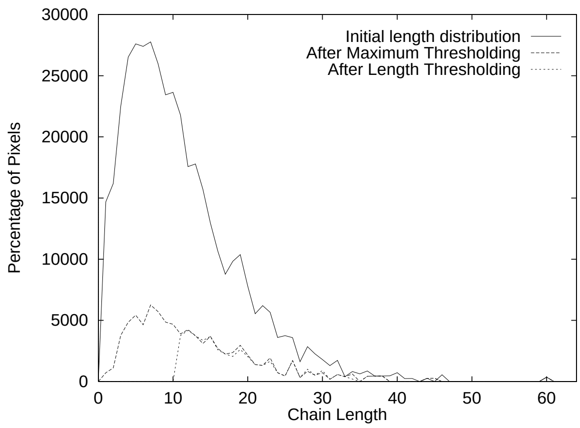

Figure 5: Extraction Stages - Real Data

Figures 5 and 6 shows the length histogram after the different extraction stages. Figure 5 shows real data while figure 6 is synthetic data generated by the model. It can be seen that qualitatively all steps of the extraction process are modeled correctly. Quantitative analysis shows that the largest error is introduced by the initial histogram generation.

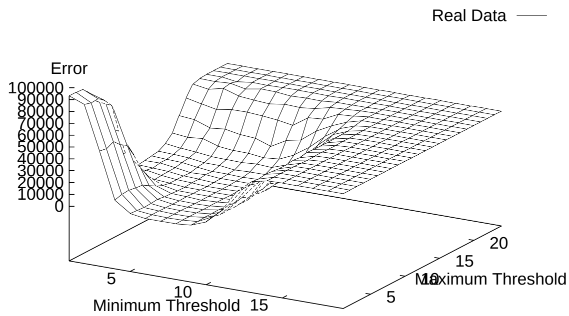

Figure 7: Real measured Detection Error

Figure 7 shows the Error surface that is obtained when the parameters T_{Min} and T_{Max} are varied with the length threshold T_{Len} set to 10. Figure 8 shows the same surface predicted by our model. Two things are apparent. First the absolute precision of the model for small threshold values is low. In this case a great number of noise pixels is detected as contours. We believe this is mainly due to the tradeoff described in section 4.4. Second the location of the (for this S/N ratio) relatively large region where the error is minimal is correctly predicted by the model.

To obtain a quantitative result we compare the error obtained with the predicted best parameters to the

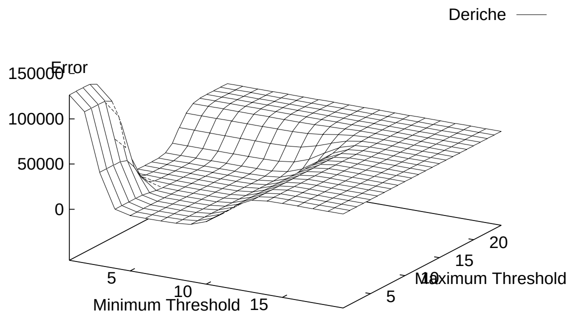

Figure 8: Predicted Detection Error

error obtained with the optimal parameters. The optimal parameters where found by a exhaustive search over the parameter space. 40 measurements have been made with a signal to noise ratio varying from infinity to 0.8. The result is shown in figure 9. 1 on the y-scale denotes that all image pixels are classified incorrectly. Calculation of the optimal parameters for all S/N ratios of the plot by exhaustive search took 16 hours on a Sparc10 while the model predicted parameters were calculated in about 10 minutes. The length threshold was set to 10 to avoid increasing the computational time of the exhaustive search by a factor of 10. Both times can be optimized considerably.

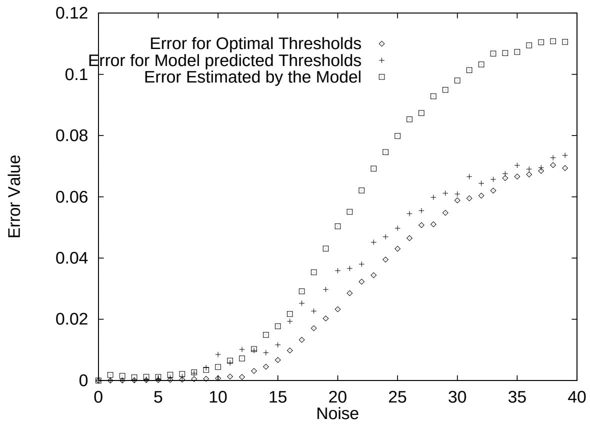

Figure 9: Evaluation of Parameter Adjustment

It can be seen that the prediction of the error of the extraction process is not very precise. The model can therefore not be used to predict the quality of the results of the contour extraction process directly. The real error obtained with the parameter settings proposed by the model however is only slightly worse than the optimal parmaeter settings. From the plot we read a precision 1 percent of the image pixels or 10 percent of the conotur pixels.

Finally we want to compare our model pased parameter adjustment with static parameter adjustment. Figure 10 shows results of the above experiment received with the optimal setting as well as several static parameter settings (chosen to be optimal for one signal to noise level only). It can be seen that no parameter setting provides good results over the complete signal to noise range.

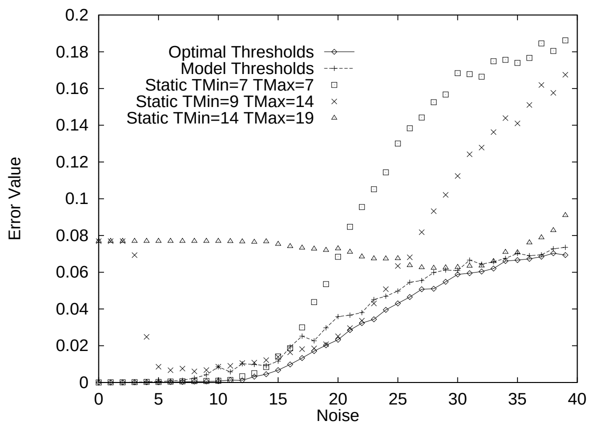

Figure 10: Model Based vs. Static Parameter Tuning

Further experimantal data can be found in [7].

6 Conclusion

Quantitative evaluation of computer vision systems is an increasingly important subject. Quantitative evaluation is only possible if methods for automatic parameter tuning are found. Currently the only approach to parameter tuning is to develop a model of the vision process and to use it to predict parameter settings. In this article we presented a scheme for the evlauation of contour extraction schemes and the adjustment of their parameters. We have shown that this systems gives valid results and have given an estimation of its precision.

Future work will include the usage of the parameter tuning scheme for tuning of parameters for real-time vision systems as well as a qunatitative analysis of image enhancement and non-linear, adaptive contour detection schemes.

References

[1] Session Authors, "Special session on the evaluation of moder edge operators," in *SPIE Con-*

ference on Computer Vision and Robotics, Apr. 1992.

[2] Session Authors, "Section ix - computer vision performance characterization," in *23. ARPA Image Understanding Workshop*, Nov. 1994.

[3] V. Ramesh, R. M. Haralick, X. Zhang, D. C. Nadadur, and K. Thornton, "Automatic selection of tuning parameters for feature extraction sequences," in *Computer Vision and Pattern Recognition*, pp. 672–677, 1994.

[4] V. Ramesh and R. M. Haralick, "Random perturbation models and performance charcterization in computer vision," in *Computer Vision and Pattern Recognition*, pp. 521–527, 1992.

[5] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[6] R. Deriche, "Fast algorithms for low-level vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 78–87, Jan. 1990.

[7] G. Appenzeller, "Image filtering for 3d reconstruction," Master's thesis, University of Karlsruhe, October 1995.