PERFORMANCE EVALUATION IN NATURAL AND CONTROLLED ENVIRONMENTS APPLIED TO FEATURE EXTRACTION PROCEDURES

Marc Luxen

University of Bonn
Institute for Photogrammetry, Nußallee 15, 53115 Bonn, Germany
http://www.ipb.uni-bonn.de/∼marc luxen@ipb.uni-bonn.de

Commission III/8

KEY WORDS: Algorithms, Performance, Environment, Reference Data, Feature, Extraction, Comparison

ABSTRACT:

The paper highlights approaches to reference data acquisition in real environments for the purpose of performance evaluation of image analysis procedures. Reference data for the input and for the output of an algorithm is obtained by a) exploiting the noise characteristics of Gaussian image pyramids and b) exploiting multiple views. The approaches are employed exemplarily in the context of evaluating low level feature extraction algorithms.

1 INTRODUCTION

As most image analysis procedures are addressed to real world applications, performance evaluation in natural environments is needed for the design, optimization and selection of algorithms (Canny, J.F., 1983; Förstner, 1996; Maimone and Shafer, 1996). Considering the fact that many approaches to performance evaluation rely on reference data, we propose two methods for reference data acquisition in natural environments.

The first approach provides reference output data based on a large number of multi-perspective images. It assumes that individual outputs of an algorithm on images of different views can be fused in an estimation process that yields practically error-free estimates for the outputs on each individual image.

The second approach provides almost noise-free images with natural image structures by exploiting the noise characteristics of Gaussian image pyramids. It is applicable to single images and provides reference input at least for investigating the noise sensitivity of algorithms.

Both methods are applied exemplarily for characterizing low level feature extraction algorithms.

2 REFERENCE DATA IN PERFORMANCE CHARACTERIZATION

To sketch the impact of reference data for performance characterization purposes, this section discusses the role of reference data in characterizing and evaluating algorithms.

2.1 Characterizing and evaluating algorithms

Algorithms generally fulfill the requirements of specific tasks only to a limited extent. As an example, a point detection algorithm may only partly fulfill the requirements of object reconstruction, as it erroneously may leave out relevant points.

We refer to *evaluating* an algorithm as the process and the result of deriving statements on the usefulness of an algorithm with respect to a specific task, resulting e. g. in a score or expected costs. For example, an algorithm detecting screws on an assembly line may be evaluated by means of the expected costs which are associated with misdetection.

Evaluation may be based on the results of *performance characterization*. Performance characterization wants to provide application independent models $C_o(\cdot)$ describing relevant properties c_o of algorithms f and of their output dependent on properties c_i of the input. As an example, performance characterization of corner extraction modules may provide models describing the precision of extracted points dependent on the image noise (cf. section 4).

A general scheme for evaluating an algorithm may follow fig. 1. Firstly, the input is characterized using methods for input data characterization, resulting in input characteristics c_i. These input characteristics are used to instantiate the models $C_o(\cdot)$ which result from the characterization and describe the behavior of the algorithm dependent on the input characteristics, yielding estimates $\hat{c}_o = C_o(c_i)$ for characteristics c_o of the algorithm and its output on the given input. Based on user-specified output requirements $R(c_o)$ and cost functions $E(c_o)$, the estimated characteristics $\hat{o}_i$ may be used to estimate the costs $\hat{e}_o = E(\hat{c}_o)$ which are to be expected in case the algorithm is applied to the given data.

Abstraction hierarchy. Algorithms are implemented in programs to fulfill a function with a certain intention. In designing a computer vision system, several alternative functions may be considered to follow a certain intention. For example, feature based matching and intensity based matching may be considered as functions following the intention of image matching. For each function there may exist multiple algorithms, e. g. cross correlation and least squares matching as algorithms for intensity based matching. Again, several different implementations of the same algorithm may be available.

Characterization and evaluation may therefore take place on the level of the intention, the level of the function, the level of the algorithm or the level of the implementation, with the levels building a characterization hierarchy of decreasing abstraction (cf. fig. 2).

2.2 The role of reference data

Reference data serves for investigations on the lowest level of abstraction, i. e. for empirical investigations on the level of the

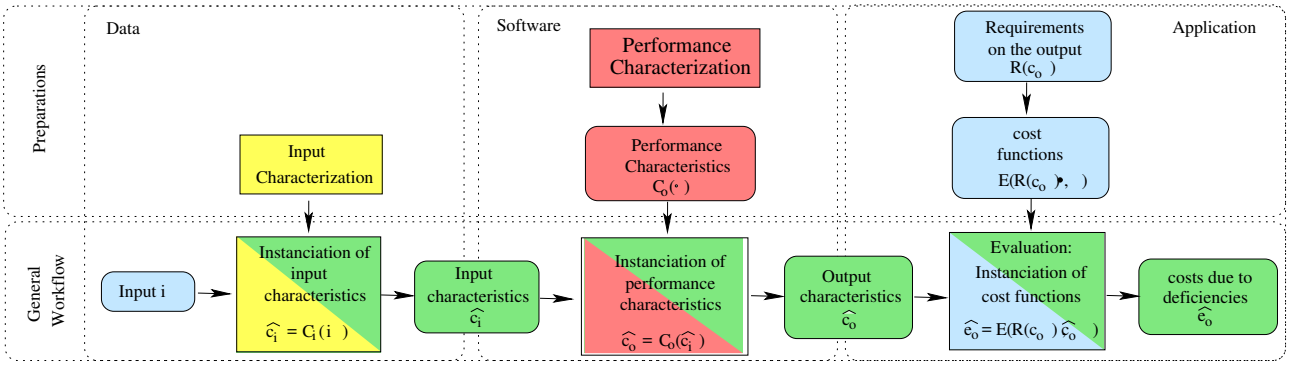

Figure 1: Work-flow for characterization and performance evaluation of algorithms.

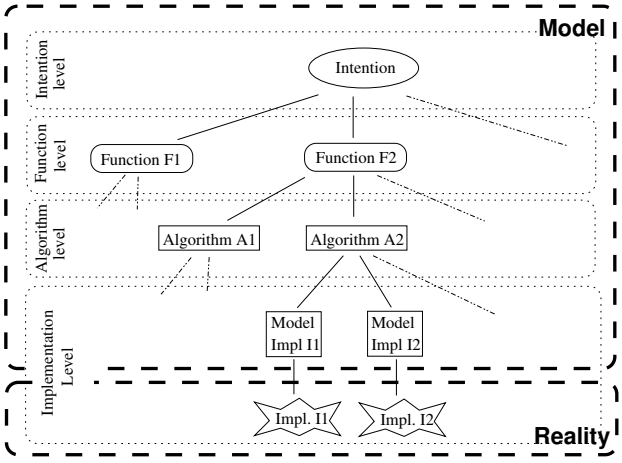

Figure 2: Abstraction hierarchy of algorithms

implementation.

In the model area, functions, algorithms or implementations are analyzed theoretically based on models that describe the data and models that describe the function, the algorithm or the implementation. In the reality domain, empirical tests on synthetic data are needed to validate the implementation or to simulate the behavior of the implementation on data that follows a certain model.

Neither theoretical analysis nor empirical testing on synthetic data gives full information about how well real world input data fits the data model that is inherent in an algorithm or an implementation. To validate model assumptions in reality, tests in real environments have to be carried out using reference data to measure deviations of system-immanent model assumptions from reality.

3 METHODS FOR REFERENCE DATA DEFINITION IN REAL ENVIRONMENTS

There are several ways to define reference data for characterizing implementations of algorithms in real environments. Reference data for the output of an implementation may be specified explicitely by a *human* or be defined as the result of a *reference implementation* of a *reference algorithm* on given data. Furthermore, reference data for the output of an arbitrary implementation I may be defined as the output of I on *reference input* data.

Our first approach to generate reference data matches best the concept of reference data from a reference algorithm. In our case, the reference algorithm employs an algorithm A on many heterogeneous data sets D_i and combines the results o_i in a robust estimation process to provide the reference output of the algorithm A on each data set. The second approach supports defining reference output based on reference input. It provides noiseless reference input data by exploiting the noise characteristics of Gaussian image pyramids.

3.1 Reference data from multiple views.

The character and the amount of deficiencies in the output of computer vision algorithms may depend on the perspective under which an object is observed. For characterizing view-dependent properties of an algorithm, test scenarios are needed based on multiple views, with reference data defining the true result for each view. In the following, we propose a method for automatically generating reference data from multiple views.

Procedure. Assuming that the outputs an of algorithm on multi-perspective images can be combined in such a way that the combination of all results yields an error-free or nearly error free result, reference data for the output on each data set may be estimated as follows:

1. A large number of images is taken from the same (parts of) an object. To minimize viewpoint-dependent errors in step 2, the exposure setup chosen is such that effects of errors in analyzing individual views are widely compensated for over the whole set of views.

2. A reference algorithm or the algorithm to be characterized itself is applied to each image I_i, yielding the individual outputs o_i.

3. The individual outputs o_i are fused to obtain a common error free result a.

4. Reference outputs for the individual views are derived from the error free result, e.g. by projection.

3.2 Reference data from multiple resolutions

Our approach to reference data acquisition from multiple resolutions provides nearly noise-free input images with natural image structures by exploiting the noise characteristics of Gaussian image pyramids. It is applicable to single images and serves reference data for at least two questions:

1. Investigations concerning the noise sensitivity of an algorithm may compare its results on noise-free input with the results on noisy data with known noise characteristics. For this purpose, noiseless test data and test data with known noise characteristics is required.

2. Results of computer vision algorithms may contain scale-dependent errors which can be estimated from the output on images of different scales. Therefore, test images are required realizing multiple levels of resolution.

Gaussian image pyramids. An image pyramid stores an image I at multiple resolutions. For a high resolution image I we build up a Gaussian image pyramid in the following way, cf. (Crowley, J.L. et al., 2002):

We start from the full resolution image $I^{(0)} = I$ of size $N_r \times N_c$ $[pel^2]$ as the lowest level image. On each pyramid level ν, the image $I^{(\nu)}$ is smoothed by two-dimensional convolution

$$I_s^{(\nu)}(r,c) = G_\sigma(r,c) * I^{(\nu)}(r,c)$$

with an isotropic Gaussian kernel

$$G_\sigma(r,c) = \frac{1}{2\pi\sigma^2} e^{-\frac{r^2+c^2}{2\sigma^2}}$$

of filter width $\sigma = 2$ [pel]. The smoothed image $I_s^{(\nu)}$ is subsampled using

$$I^{(\nu+1)}(r,c) = I_s^{(\nu)}(2r, 2c)$$

for all $(r,c) \in \{1,\ldots,N_r/2^{\nu+1}\} \times \{1,\ldots,N_c/2^{\nu+1}\}$, yielding the higher level image $I^{(\nu+1)}$.

Noise characteristics of Gaussian image pyramids.

Smoothing with a Gaussian kernel of width σ reduces image noise. The relation between the noise variance σ_0^2 of the original image and the noise variance σ_s^2 of the smoothed image is given by (cf. (Fuchs, 1998), eq. 4.8

$$\sigma_s^2 = \sigma_0^2 \int\int_{-\infty}^{\infty} G_\sigma^2(r,c)drdc = \frac{1}{4\pi\sigma^2}\sigma_0^2.$$

Thus, if σ_0^2 denotes the noise variance of the level-0 image of an image pyramid, the noise variance of the level k image is given by

$$(\sigma_n^{(k)})^2 = \frac{1}{(4\pi\sigma^2)^k}\sigma_0^2.$$

Obviously, the amount of image noise decreases rapidly with increasing pyramid levels.

Procedure. The method exploits the fact that the higher level images of an image pyramid are practically noiseless but contain the same image structure as the full resolution image – a fact that qualifies the higher level pyramid images for the use as reference data for the input signal of an algorithm. The natural image structure is widely prevented even in the higher level pyramid levels, as the decimation step on each level of the image pyramid widely compensates for smoothing the signal.

4 EXAMPLE: CHARACTERIZING FEATURE EXTRACTION ALGORITHMS

To demonstrate the feasibility of the approaches to reference data acquisition, we employ both methods for characterizing low level feature extraction modules. We consider the following issues:

1. Straight line and edge detection: Some modules for extracting linear features, i. e. straight lines and edges from images, for some reason provide line and edge segments which are systematically too short. We investigate the shortening of linear features exemplarily for the feature extraction module presented in (Fuchs, 1998).

2. Point detection: Most investigations on the noise behavior of point extraction algorithms are based on synthetic data for the signal and for the noise. Unlike these, we investigate the noise sensitivity of the point extraction proposed by (Förstner and Gülch, 1987) based on an almost noiseless *real* signal with only the added noise being synthetic.

4.1 Basics

4.1.1 Notation. We use Euclidean and homogeneous representations of points and straight lines in 2D and 3D. Euclidean coordinates of a 2D point are denoted with lowercase slanted letters $\boldsymbol{x}$, its homogeneous coordinates are denoted with lowercase upright letters $\mathbf{x} = (\boldsymbol{x}^\mathsf{T}, 1)^\mathsf{T}$. Homogeneous coordinates of a 3D point are denoted with upright uppercase letters $\mathbf{X}$. Segments of straight lines and edges in 2D are represented by their bounding points $l : (\mathbf{x}_s, \mathbf{x}_e)$. The line joining two end points $\mathbf{x}_s$ and $\mathbf{x}_e$ is given homogeneous with $\mathbf{l} = \mathbf{x}_s \wedge \mathbf{x}_e = \mathbf{x}_s \times \mathbf{x}_e$.

Stochastic entities are underlined, e. g. $\underline{\boldsymbol{x}}$, their expectation values are marked with a bar, e. g. $\bar{\boldsymbol{x}}$. Estimated entities wear a hat, e. g. $\hat{\boldsymbol{x}}$ and reference values have a tilde, e. g. $\tilde{\boldsymbol{x}}$.

4.1.2 Shortening of linear features at junctions. An example for shortened lines and edges at junctions is depicted in fig. 3c). It is drawn from the output of the feature extraction software FEX, which was applied to an intensity image of a polyhedral object. The software expects a multi-channel intensity raster image as input and provides a symbolic image description containing lists of points, blobs and linear features, the linear features being segments of straight lines or intensity edges.

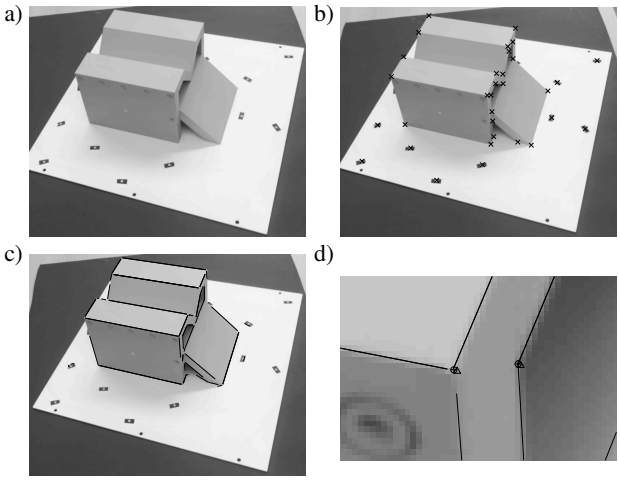

Figure 3: Situation for a single image: a) Rectified image b) Rectified image with projected reference points c) Rectified image with extracted edges d) Local situation for two junctions.

Given the true values $\tilde{\boldsymbol{x}}_s$ and $\tilde{\boldsymbol{x}}_e$ of the two end points $\underline{\boldsymbol{x}}_s$ and $\underline{\boldsymbol{x}}_e$ of a straight line or edge, the expectation value $\bar{d}$ of the shortening d is defined as

$$\bar{d} = E(d) = E(|\,\tilde{\boldsymbol{x}}_s - \tilde{\boldsymbol{x}}_e\,| - |\,\underline{\boldsymbol{x}}_s - \underline{\boldsymbol{x}}_e\,|).$$

Assuming that the line $\underline{\mathbf{l}} = \underline{\mathbf{x}}_s \times \underline{\mathbf{x}}_e$ is sufficiently parallel to the reference $\tilde{\mathbf{l}} = \tilde{\mathbf{x}}_s \times \tilde{\mathbf{x}}_e$, it approximately holds

$$\begin{aligned}\bar{d} &\approx E(|\,\tilde{\boldsymbol{x}}_s - \underline{\boldsymbol{x}}_s\,| + |\,\tilde{\boldsymbol{x}}_e - \underline{\boldsymbol{x}}_e\,|) \\ &= E(|\,\tilde{\boldsymbol{x}}_s - \underline{\boldsymbol{x}}_s\,|) + E(|\,\tilde{\boldsymbol{x}}_e - \underline{\boldsymbol{x}}_e\,|)\end{aligned}$$

and in a symmetric situation, i. e.

$$\bar{\delta} = E(|\,\tilde{\boldsymbol{x}} - \underline{\boldsymbol{x}}\,|) = E(|\,\tilde{\boldsymbol{x}}_s - \underline{\boldsymbol{x}}_s\,|) = E(|\,\tilde{\boldsymbol{x}}_e - \underline{\boldsymbol{x}}_e\,|),$$

it holds

$$\bar{d} = 2\bar{\delta} = 2E(|\,\tilde{\boldsymbol{x}} - \underline{\boldsymbol{x}}\,|),$$

$\bar{\delta}$ denoting the expectation value of the one-sided shortening at each end point of a line or edge segment.

Given N observed end points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ of linear features with their reference points $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N$, the mean one-sided shortening $\bar{\delta}$ and the variance $\sigma^2_{\delta_n}$ of the one-sided shortenings $\delta_n = |\,\tilde{\boldsymbol{x}}_n - \boldsymbol{x}_n\,|$ may be estimated from

$$\hat{\delta} = \frac{1}{N}\sum_{n=1}^{N}\delta_n \tag{1}$$

and

$$\hat{\sigma}^2_{\delta} = \frac{1}{N-1}\sum_{n=1}^{N}(\delta_n - \hat{\delta})^2. \tag{2}$$

4.1.3 Noise characteristics of point extraction. The noise sensitivity of point extraction algorithms may be characterized in terms of the quality of the point localization under varying image noise, quantified with the bias $\boldsymbol{b} = \boldsymbol{b}(\sigma_n^2)$ and covariance matrix $\boldsymbol{\Sigma}_{xx}(\sigma_n^2)$ of extracted points dependent on the image noise variance σ_n^2 .

Given N independent point observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ of equal precision, an estimate $\hat{\boldsymbol{x}}$ for true point $\tilde{\boldsymbol{x}}$ is given by the mean

$$\hat{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n \tag{3}$$

and the bias $\boldsymbol{b}$ of the observations and their covariance matrix $\boldsymbol{\Sigma}_{xx}$ may then be estimated from (cf. (Luxen, 2003))

$$\hat{\boldsymbol{b}} = \tilde{\boldsymbol{x}} - \hat{\boldsymbol{x}} \quad \text{and} \quad \widehat{\boldsymbol{\Sigma}}_{xx} = \frac{1}{N-1}\sum_{n}(\boldsymbol{x}_n - \hat{\boldsymbol{x}})(\boldsymbol{x}_n - \hat{\boldsymbol{x}})^{\mathsf{T}}. \tag{4}$$

4.2 Test procedures

With a calibrated digital camera Kodak DCS 460, images of a polyhedral object were taken from 46 different perspectives, cf. fig. 4. All images were corrected referring to distortion.

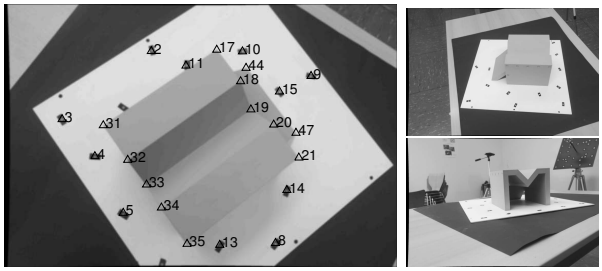

Figure 4: Images of a polyhedral object (sample). Image size: 2036×3060 [pel].

4.2.1 Characterizing the shortening of linear features The shortening of linear features provided by the feature extraction software FEX is investigated by comparing the end points of extracted straight lines and edges with ground truth resulting from the multiple view approach.

Reference data estimation by multiple view approach. In case of a precise polyhedral object, the end points of straight line and edge segments coincide with imaged object corners, and reference values for the image coordinates of object corners can be considered as reference for the end points of straight line and edge segments.

Therefore, to estimate reference data, the object corners were extracted from each image using the corner extraction proposed by (Förstner and Gülch, 1987). Based on approximate values for the image orientations as well as for the object coordinates, the point correspondence problem was solved and spurious features were eliminated. A bundle adjustment was carried out for simultaneously estimating the projection matrices $\mathbf{P}_j$ of all images I_j and the coordinates $\mathbf{X}_i$ of the corresponding corners C_i in object space. As the comprehensive exposure setup realizes heterogeneous viewing angles for almost every object point and due to the fact that in the estimation process the redundancy is very high, effects of small errors in the mensuration process were assumed to be negligible and the result of the object reconstruction to be complete. Therefore, the estimated coordinates $\hat{\mathbf{X}}_i$ and the estimated projection matrices $\hat{\mathbf{P}}_j$ were considered as reference data in object space. Reference data in the image domain was obtained by projection

$$\tilde{\mathbf{x}}_{ij} \doteq \hat{\mathbf{x}}_{ij} = \hat{\mathbf{P}}_j\hat{\mathbf{X}}_i, \tag{5}$$

resulting in reference values $\tilde{\mathbf{x}}_{ij}$ for the image coordinates $\mathbf{x}_{ij}$ of each corner C_i in each image I_j.

Analysis of extracted lines and edges. The feature extraction software FEX was applied to each image, with the control parameters being optimized by visual inspection. For each image I_j, the end points $\boldsymbol{x}_{kj,s}$ and $\boldsymbol{x}_{kj,e}$ of all extracted line segments l_{kj} were matched to the reference points $\tilde{\boldsymbol{x}}_{ij}$ by employing a distance threshold $\epsilon = 20$ [pel], for each reference point $\tilde{\boldsymbol{x}}_{ij}$ leading to a set

$$\begin{aligned}\mathcal{L}_{ij} = \Big\{(\boldsymbol{x}_{kj,s}, \tilde{\boldsymbol{x}}_{ij}) \;\Big|\; |\,\boldsymbol{x}_{kj,s} - \tilde{\boldsymbol{x}}_{ij}\,| < \epsilon\Big\} \cup \\ \cup \Big\{(\boldsymbol{x}_{kj,e}, \tilde{\boldsymbol{x}}_{ij}) \;\Big|\; |\,\boldsymbol{x}_{kj,e} - \tilde{\boldsymbol{x}}_{ij}\,| < \epsilon\Big\}\end{aligned} \tag{6}$$

of point-to-reference-point correspondences. Using eq. 1, the one-sided-shortening $\bar{\delta}$ was estimated junction-wise for each $\mathcal{L}_{ij}$, leading to estimates $\hat{\delta}_{ij}$, image-wise over $\mathcal{I}_j = \bigcup_i \mathcal{L}_{ij}$, leading to estimates $\hat{\delta}_j^{(\mathcal{I})}$ and for each junction over all images, i. e. over $\mathcal{J}_i = \bigcup_j \mathcal{L}_{ij}$, leading to estimates $\hat{\delta}_i^{(\mathcal{J})}$.

4.2.2 Characterizing the noise sensitivity of corner extraction.

Reference data from multiple resolutions approach. To investigate the robustness of the corner extraction (Förstner and Gülch, 1987) with respect to noise, image pyramids were generated for all images.

The third level image $I_j^{(3)}$ was taken from each pyramid, embodying an almost noiseless image $\tilde{I}_j = I_j^{(3)}$ with real image structure. Reference coordinates $\tilde{\boldsymbol{x}}_{ij}^{(3)}$ for the object corners in the third level images were derived from the reference coordinates $\tilde{\boldsymbol{x}}_{ij}$ (eq. 5) by scaling,

$$\tilde{\boldsymbol{x}}_{ij}^{(3)} = 4\,\tilde{\boldsymbol{x}}_{ij}. \tag{7}$$

Analysis of points under image noise. To generate noisy image data, the reference images $\tilde{I}_j$ were contaminated with zero mean Gaussian white noise

$$\underline{n} \sim \mathcal{N}(0, \sigma_n^2),$$

the noise variance being varied from $\sigma_n = 0.1$ grey values ([gr]) to $\sigma_n = 12$ [gr] in steps of $\sqrt{2}$ [gr].

$N = 100$ test images $I_{j,1}^{(\sigma_n)}, \ldots, I_{j,100}^{(\sigma_n)}$ were generated for each image $\tilde{I}_j$ and each noise level σ_n. The point extraction was applied to each test image, for each point $\tilde{\boldsymbol{x}}_{ij}^{(3)}$ leading to a set

$$\mathcal{P}_{ij}^{(\sigma_n)} = \left\{ {}^{(\sigma_n)}\boldsymbol{x}_m \;\middle|\; |{}^{(\sigma_n)}\boldsymbol{x}_m - \tilde{\boldsymbol{x}}_{ij}^{(3)}| < \epsilon \right\}$$

of observations ${}^{(\sigma_n)}\boldsymbol{x}_m$.

Using eq. 4, the bias $\boldsymbol{b}$ and the covariance matrix $\boldsymbol{\Sigma}_{xx}$ of the observations were estimated point-wise for each $\mathcal{P}_{ij}^{(\sigma_n)}$ over all noise levels σ_n.

4.3 Provisional results

First results of our experiments are plausible, indicating that the proposed methods for reference data definition may be successfully used in characterizing image processing algorithms.

4.3.1 Noise characteristics of corner extraction. Results concerning the noise sensitivity of the corner extraction are illustrated in fig. 5 and fig. 6.

As to be expected, the empirical standard deviations $\hat{\sigma}_x$ and $\hat{\sigma}_y$ of extracted corners in x- and y-direction and the resulting mean error $\hat{\sigma}_p = \sqrt{\hat{\sigma}_x^2 + \hat{\sigma}_y^2}$ increase with increasing image noise (cf. fig. 5). For most corners, the increase of $\hat{\sigma}_x$, $\hat{\sigma}_y$ and $\hat{\sigma}_p$ is stronger than linear and thus stronger than to be expected. This may be caused by the fact that to detect all desired corners, the smoothing parameter σ_1 of the corner extraction was adapted linearly to the standard deviation σ_n of the image noise, reaching from $\sigma_1 = 0.7$[pel] for $\sigma_n = 1[gr]$ to $\sigma_1 = 0.9$ [pel] for $\sigma_n = 10[gr]$. As smoothing deteriorates the quality of the point localization (cf. (Canny, J.F., 1983)), the loss of precision may thus be partly caused by enlarging the smoothing filter for images with a larger amount of noise.

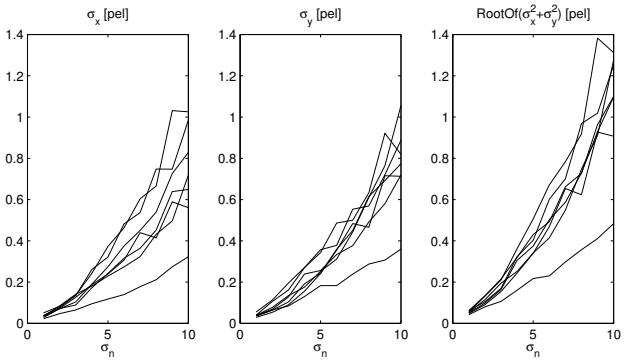

Figure 5: Empirical precision of extracted corners on noisy images. Each curve represents the uncertainty of a single point. Left: Empirical standard deviation σ_x in x-direction. Center: Empirical standard deviation σ_y in y-direction. Right: Mean localization error $\sigma_p = \sqrt{\sigma_x^2 + \sigma_y^2}$.

Also the estimated bias $|\ \hat{\boldsymbol{b}}\ |$ of extracted points increases with increasing image noise (cf. fig. 6), which will be mainly due to

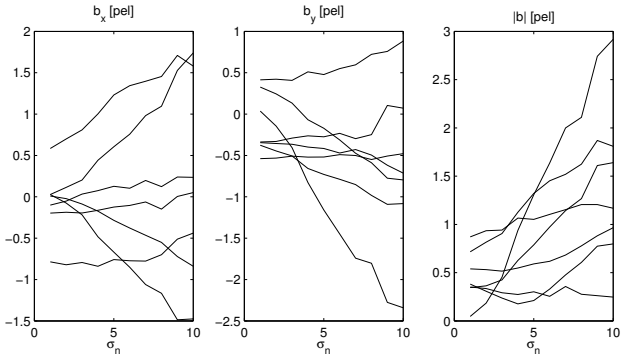

Figure 6: Estimated bias of extracted corners on noisy images, each curve representing the bias of a single corner. Left: Bias b_x in x-direction. Center: Bias b_y in y-direction. Right: Norm $|\ \boldsymbol{b}\ |$ of the bias.

enlarging the smoothing filter dependent on the image noise. As to be expected, the different behavior of the bias components $\hat{b}_x$ and $\hat{b}_y$ of different points indicates that the bias depends on the perspective under which a corner is observed.

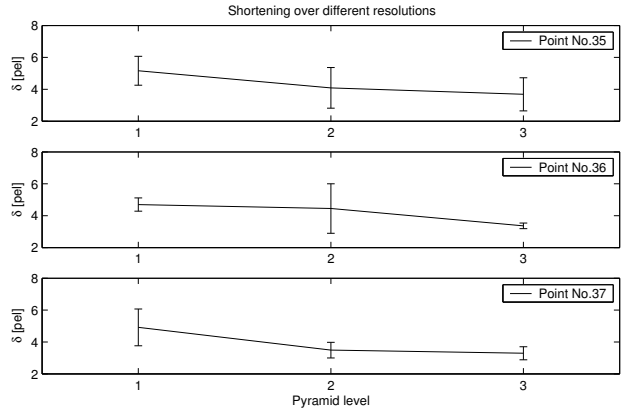

Figure 7: Shortening of straight lines and edges on different resolutions.

4.3.2 Shortening of edges at junctions The results concerning the shortening of extracted lines and edges at junctions are depicted in the figs. 7, 8 and 9.

For three junctions in a single image, fig. 7 shows the one-sided shortening of the junction branches dependent on the image resolution. The results were drawn from the three lowest levels of an image pyramid. The mean shortening of extracted lines reaches from 3 to 5 pixels. It decreases with decreasing image resolution.

For a single junction, in fig. 8 the mean and the variance of the shortening of edges is depicted over all images. The shortening varies over different images, depending on the perspective under which the junction is observed and on its illumination. The shortening is large especially in situations with low contrast at edges.

Fig. 9 shows for each junction the mean and the standard deviation of the shortening of adjacent edges, with the mean and the standard deviation being taken over all images. The one-sided shortening reaches up to 10 pixels. Again the worst results are obtained for junctions with low contrast at edges.

5 CONCLUSIONS AND OUTLOOK

This paper proposes two methods for generating reference data in the context of characterizing image processing algorithms. The

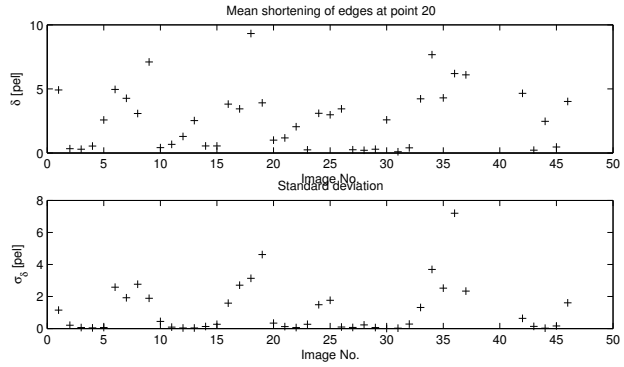

Figure 8: Shortening of straight lines and edges at a single junction over all images

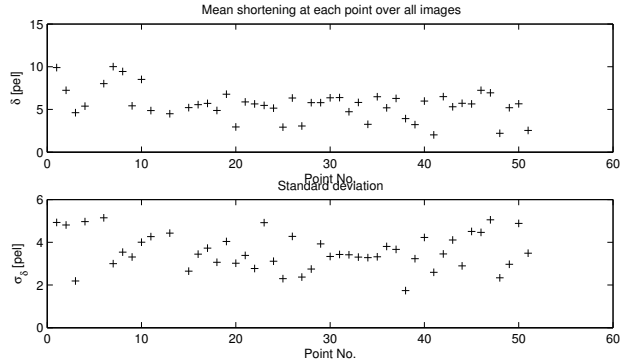

Figure 9: Shortening of straight lines and edges for all junctions. Mean and the variance are taken over all images

role of reference data in characterizing and evaluating algorithms is discussed and the approaches to reference data generation are explained.

The methods were successfully employed to investigate the noise sensitivity of point extraction modules and the shortening of straight lines and edges provided by linear feature extraction modules. First results are plausible and give reason 1) to exploit the proposed methods for reference data generation on other data sets, 2) to further investigate the noise behavior of corner extraction modules including local image characteristics such as shape or local contrast and 3) to further investigate the shortening of linear features dependent at junctions dependent on the contrast at the junction branches.

ACKNOWLEDGMENT

This work has been supported by the German Research Council (DFG).

References

Canny, J.F., 1983. Finding Edges and Lines in Images. Technical report, MIT Artificial Intelligence Laboratory.

Crowley, J.L., Riff, O. and Piator, J.H., 2002. Fast computation of characteristic scale using a half octave pyramid. In: CogVis 2002, International Workshop on Cognitive Vision, Zürich.

Förstner, W., 1996. 10 Pros and Cons Against Performance Characterization of Vision Algorithms. In: Workshop on "Performance Characteristics of Vision Algorithms" , Cambridge.

Förstner, W. and Gülch, E., 1987. A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features. In: Proceedings of the Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, pp. 281–305.

Fuchs, C., 1998. Extraktion polymorpher Bildstrukturen und ihre topologische und geometrische Gruppierung. DGK, Bayer. Akademie der Wissenschaften, Reihe C, Heft 502.

Luxen, M., 2003. Variance component estimation in performance characteristics applied to feature extraction procedures. In: B. Michaelis and G. Krell (eds), Pattern Recognition, 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003, Proceedings, Lecture Notes in Computer Science, Vol. 2781, Springer, pp. 498–506.

Maimone, M. and Shafer, S., 1996. A Taxonomy for Stereo Computer Vision Experiments. In: ECCV Workshop on Performance Characteristics of Vision Algorithms, pp. 59 – 79.