A Knowledge-Based Videotheodolite Measurement System for Object Representation/Monitoring

Alexander Reiterer [a] Uwe Egly [b] Thomas Eiter [b] Heribert Kahmen [a]

[a] *Institute of Geodesy and Geophysics, Engineering Geodesy, Vienna University of Technology, Vienna, Austria.*

[b] *Institute of Information Systems, Knowledge Based Systems Group, Vienna University of Technology, Vienna, Austria.*

Abstract

High-precision online 3D-measurement systems can perform their measurements with and without targeting. Systems which are able to measure without artificial targets use the texture on the surface of the object to find 'interesting points'. However, well-trained 'measurement experts' are required to operate such a measurement system.

In order to make such systems easy to use even for non-experts, we extend it by a knowledge-based component which supports the operator. We report on the architecture and functionality of the respective knowledge-based system, its development stage and the promising results obtained in experimentation.

Key words: Knowledge-Based System, Videometric System, Interest Operator, Image Processing.

1 Introduction

In science and industry (like, e.g., in architecture, medicine, or construction), highly accurate 3D representations of objects are required. A great variety of optical 3D measurement techniques like laser scanners, photogrammetric systems, or image-based measurement system is available to achieve this need.

In comparison with laser scanners, image-based systems measure objects with higher accuracy; compared with photogrammetric systems, they can be easier used for on-line measurement processes (e.g. object monitoring). This will especially be the case, if the measurements can be performed with a high degree of automation.

Recently, research interest in the area of image-based measurement systems has been increased. *Leica Geosystems* [15] developed a prototype of an "image-assisted total station" with the purpose of defining a hybrid or semi-automatic way to combine the strength of the traditional user-driven surveying mode with the benefits of modern data processing. Furthermore, *Sokkia* [14] introduced a prototypical tacheometer which provides focused color images. At the *Technische Universität München*, an image-based measurement system for object recognition was developed [16].

The central topic of all image-based measurement systems is the calculation of 3D object coordinates from 2D image coordinates for subsequent processing steps, like deformation analysis or object reconstruction. Such a system consists of an image sensor, components for image acquisition and image processing, a computer for system control, and some output devices [5]. Image-based measurement systems perform their measurements with or without targeting. Some applications like, e.g., the monitoring of buildings definitely require measurement systems *without* artificial targets because they would highly disturb the architectonic impression. A method to replace these targets is to use the texture on the object surface to find 'interest points' by *interest operators*. The location of an interest point is determined by consecutive measurements (e.g., once a month) in order to detect a displacement of the building. The disadvantage of such systems is the need for a well-trained 'measurement expert' having special skills and experience to properly operate the complex system. Details about such a measurement system can be found in [6,12].

In a complex measurement system with many algorithms for image processing and many interest operators, the selection of suitable algorithms, their order of application, and the choice of input parameters is a non-trivial task. To provide automated support, a knowledge-based approach has been chosen for representing the knowledge necessary for this decision-making, allowing for a declarative and modular representation of a decision policy together with easy extendibility. To the best of our knowledge, the described system is the first intelligent system for such a complex measurement application.

The developed program system is written in two different languages. The knowledge-based system has been carried out in CLIPS, a productive development tool which provides a complete environment for the construction of *rule- and object-based systems* [1]. The remaining parts of the implementation (image analysis, image processing, interest operators, graphical user interface, etc.) have been carried out in C++. A system overview is shown in Figure 1 which is described in detail below.

2 Knowledge-based Image Processing

Experiments [6,12,10] have shown that a necessary precondition for the successful application of algorithms for finding interesting points is the "quality" of the image. It is often required to improve the visual appearance of an image (like for the picture in Figure 2 (a)). This can be done by image preprocessing and enhancement processes. Furthermore, flexible image processing makes the measurement system more independent of variable illumination during image capturing.

We have implemented the following image processing algorithms: histogram equalization, gray-level scaling (image brightening/darkening), median and gauss filtering,

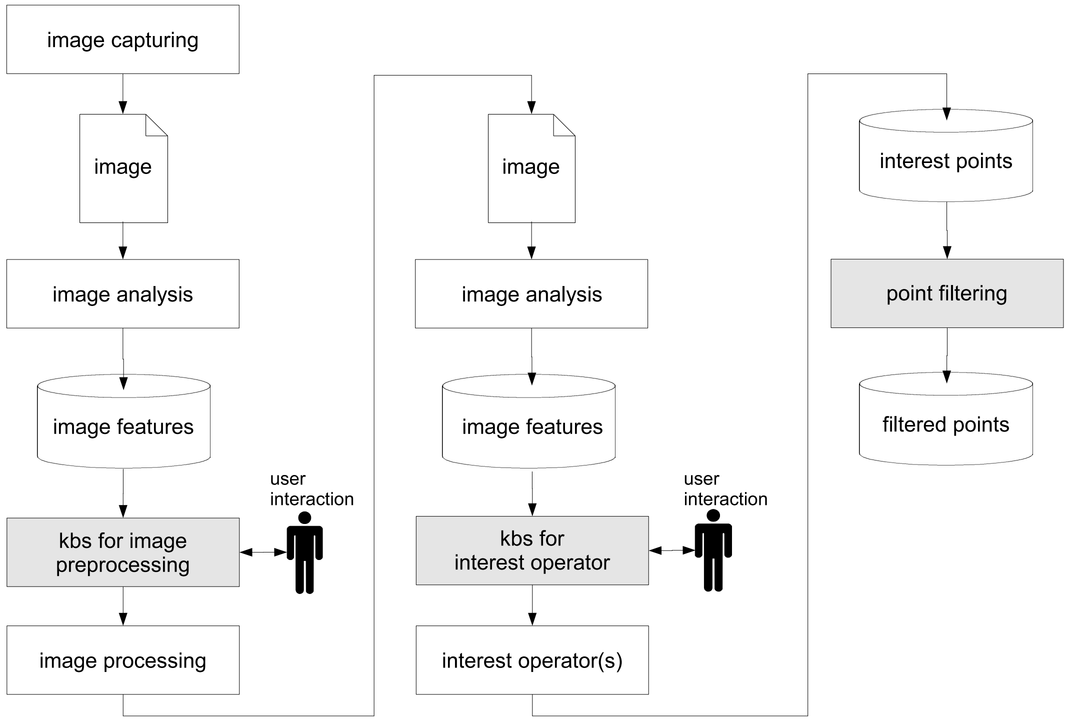

Fig. 1. System overview and data flow.

edge detection (Sobel-, Prewitt-, Laplace operator) and thresholding. The implementation of further algorithms for image preprocessing and enhancement is envisaged. We leave this for future extensions of the system.

The aim of the developed knowledge-based image processing system is to select, on the basis of extracted image features, suitable image processing algorithms. This is done by three processing steps (see the left part of Figure 1).

First step: image analysis. After the image is captured, the image analysis is carried out. Image features are stored in a file in a suitable form.

Second step: choice of image processing algorithms. Based on the extracted image features, the knowledge-based system (KBS) chooses a single algorithm or a combination of algorithms for image preprocessing and enhancement in order to improve the image for the subsequent application of interest operators. At critical processing steps (e.g., edge detection, median filtering), the user can overrule the system decision.

Third step: image processing. The last step in the processing chain is the application of the chosen processing steps. The result is an improved image.

The knowledge which was required to be included in this part of the knowledge base was obtained in different ways: from technical literature [9], other projects [12,6] and from extensive experiments [10]. The knowledge base (the part for the knowledge-based image preprocessing and image enhancement system) is divided into three groups of rules: (1) rules for the choice of suitable algorithms for image preprocessing and enhancement, (2) rules to define their execution order and (3) rules for the predefinition of necessary parameters.

Extensive experimentation showed that the algorithms for image processing and image enhancement can be chosen and combined on the basis of parameters extracted from the image (low-level feature extraction).

This is done by calculating the so-called **histogram features**.

The histogram of an image is a plot of the gray-level values versus the number of pixels at that value. It can be utilized to generate a class of image features (histogram features). The first-order probability distribution of the image amplitude may be estimated from

$$h(z) = \frac{H(z)}{N}; \qquad \sum_{z=z_u}^{z_o} h(z) = 1. \tag{1}$$

N represents the total number of pixels in the full image and $H(z)$ the number of pixels of amplitude z. Based on the histogram, several features have been formulated. *Mean* (M_1), *standard deviation* (M_2) and *skewness* (M_3) describe the shape of the image histogram and are calculated by

$$M_1 = \overline{z} = \sum_{z=z_u}^{z_o} z h(z); \tag{2}$$

$$M_2 = \sum_{z=z_u}^{z_o} (z - \overline{z})^2 \, h(z); \tag{3}$$

$$M_3 = \sum_{z=z_u}^{z_o} (z - \overline{z})^3 \, h(z). \tag{4}$$

Mean (M_1) is correlated to the brightness of the image; a bright image will have a high mean and a dark image will have a low mean. *Variance* (M_2) is a measure of the average distance between each gray-level and the mean value. The *standard deviation* is the square root of the *variance*. It describes the spread of the gray-levels; a high-contrast image will have a high *standard deviation* and a low-contrast image will have a low *standard deviation*. *Skewness* (M_3) is a measure of the symmetry of a distribution of gray-levels around their mean. Symmetrical data have a skewness near zero. *Skewness* indicates the balance of the bright and dark areas in the image. The *skewness* value will be positive for an image with darker areas and negative for an image with brighter areas.

To make the image features more suitable for the knowledge-based decision system, we use a special classification procedure. This procedure translates the input values (image features) into linguistic concepts like, the entropy is high, low, etc. The use of these concepts permits us to write rules in terms of easily-understood word descriptors, rather than in terms of numerical values.

As an example, we list in the following the relevant rule for the detection of necessity of *edge detection*, which is used only if the image has poor quality and the subsequent application of an interest operator would not succeed. The rule has the following syntax (CLIPS syntax), and should be self-explanatory ("|" means "or"):

```
(defrule edge
  (Stat_Moments (M1_f very_low))
  (Stat_Moments (M2_f very_low))
  (Stat_Moments (M3_f middle_negative | high_negative |
                      very_high_negative | middle_positive |
                      high_positive | very_high_positive))
=>
  (assert (condition (edge yes))))
```

If edge detection is necessary, a suitable algorithm has to be selected. The implemented edge detection algorithms are able to detect edges in vertical direction separated from edges in horizontal direction. Therefore, these methods have a precondition (horizontal or vertical homogeneity), which relies on the homogeneity in these two directions. Homogeneity is inspected by separate rules. The relevant rule for the Sobel edge detection has the following syntax:

```
(defrule sobel_h
  (condition (edge yes))
  (condition (horizontal_homogeneity yes))
  (9_Haralick (H9_AVG_f very_low | low))
 =>
  (assert (condition (sobel yes))))
```

Due to the moderate number of implemented image preprocessing and image enhancement algorithms, the knowledge base could be kept compact and thus easily modifiable and extensible. The complete knowledge base for the choice of suitable image preprocessing and image enhancement algorithms comprises approximately 40 rules.

3 Point Detection

The second component of our system is the knowledge-based point detection by means of interest operators.

3.1 Interest Operators

Interest operators (IOPs) play an important role in computer vision and image processing. They highlight points which can be found using correlation methods. There exist many IOPs [2,4,7]; however, no IOP is suitable to find all desired points. For this reason, we have implemented in our system different IOP algorithms (Förstner operator, Harris operator and the Hierarchical Feature Vector Matching operator). The choice of one or more suitable algorithm(s), their combination and parameter(s) is made by the KBS. More details about the implementation, accuracy and stability of interest operators in a videometric measurement system can be found in [6,12].

The ***Förstner operator*** [2] is based on the assumption that a corner point is the point that is statistically closest to all the edge elements along the edges intersecting at that corner (the point location is determined through a least squares adjustment procedure). The operator evaluates the quality of the corner points by analyzing the shape and size of error ellipses describing the variance covariance matrix associated with the derived corner point location. To quantify these properties, 'roundness' q and 'size' W of the error ellipse are defined:

$$q = \frac{4 \cdot detN}{(traceN)^2} , \quad W = \frac{detN}{traceN} , \tag{5}$$

$$N^{-1} = \begin{bmatrix} gu^2 & gu \cdot gv \\ gu \cdot gv & gv^2 \end{bmatrix} , \tag{6}$$

where gu and gv are the derivatives of the grey values of image pixels across the rows (u) and columns (v) of the window. An interest point is defined by values of W and q greater than some thresholds (W_{min} and q_{min}) and extreme maximum values in the neighborhood. Reliable corner points should have a near circular error ellipse with a small size. A larger W indicates a smaller error ellipse and a circular error ellipse will have a maximum q value of 1.

The second IOP is the ***Harris operator*** [4]. Instead of using a simple sum, a Gaussian is used to weight the derivatives inside the window. Interest points are detected if the auto-correlation matrix has two significant eigenvalues.

The Harris operator consists of the following process steps [4]:

- smooth the image by convolving it with a gauss filter $G(x, y)$;
- compute the image gradient $\nabla I(x, y)$ for each pixel:

$$\nabla I(x, y) = \left[\frac{\partial I(x, y)}{\partial x} , \frac{\partial I(x, y)}{\partial y} \right] ; \tag{7}$$

- compute the symmetric positive semi-definite 2×2 matrix A for each pixel and a given size of N_0 (the integrative scale σ_I) as follows:

$$A = \sum_{(x,y) \in N_0} \nabla I(x, y) \nabla I(x, y)^T ; \tag{8}$$

- evaluate the response function for each pixel $R(x, y)$:

$$R(x, y) = corn = detA - \kappa \, trace^2 A , \tag{9}$$

  where $\kappa = 0.04$;
- choose the interest point as local maximum of function $R(x, y)$.

Hierarchical Feature Vector Matching (HFVM) was developed at the *Institute of Digital Image Processing* of *Joanneum Research* in Graz (Austria), as a new matching technique [7]. Part of the whole process is the detection of *interesting points*. This part is used here and will be called in the following text in a simplified way as *Hierarchical Feature Vector Matching operator*.

The HFVM operator is based on the idea of creating a feature vector for each pixel in the image (constructing a *feature image*). This feature vector contains all the features of one location for the corresponding pixel. Finding a match means comparing a feature vector of a reference image (the so-called reference vector), with all feature vectors of the search area which is part of the search image. The robustness and efficiency of the algorithm will be improved by creating pyramids of the input image (pyramid levels). The result of each pyramid level, the so-called disparity map, is used as input for the matching of the next level.

3.2 Knowledge-based Point Detection

As the knowledge-based system for image processing, also this sub-system selects algorithms on the basis of parameters extracted from the image. Additionally to the histogram features (described in Section 2), the so-called **Haralick features** and features collected by **user-queries** (object type, lighting conditions, etc.) are used.

Haralick et al. [3] proposed 13 measures of textural features which are derived from the co-occurrence matrices, a well-known statistical technique for texture feature extraction. Texture is one of the most important defining characteristics of an image. The grey-level co-occurrence matrix is the two dimensional matrix of joint probabilities $p(i,j)$ between pairs of pixels, separated by a distance *d* in a given direction *r*. It is based on the repeated occurrence of some grey level configuration in the texture. Here, we use only four of the features. In order to simplify notation, all sums $\sum_k$ with $k \in \{i,j\}$ range from 1 to N_g, i.e., the number of grey levels in the image.

The used Haralick moments are defined as

$$H_1 = \sum_i \sum_j p(i,j)^2; \tag{10}$$

$$H_2 = \sum_{n=0}^{N_g-1} n^2 \Bigg(\sum_{\substack{1 \le i,j \le N_g \\ n = |i-j|}} p(i,j) \Bigg); \tag{11}$$

$$H_5 = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j); \tag{12}$$

$$H_9 = \sum_i \sum_j p(i,j) \log\left(p(i,j)\right). \tag{13}$$

The *angular second moment* (H_1) is a measure of the homogeneity of an image, i.e., it detects disorders in textures. For homogeneous textures, H_1 turns out to be small compared to non-homogeneous textures. The *contrast* (H_2) is a measure of the amount of local variations present in an image. This information is specified by the matrix of relative frequencies $p(i,j)$ with which two neighboring pixels occur on the image, one with grey-value i and the other with grey-value j. The *inverse difference moment*

(H_5) measures image homogeneity, too. It achieves its largest value when most of the occurrences in co-occurrence matrices are concentrated near the main diagonal. H_5 is inversely proportional to the contrast of the image. The *entropy* (H_9) is related to the information-carrying capacity of the image. H_9 is maximized when the probability of each entry is the same. Thus, a high value for entropy means that the gray level changes between pixels are evenly distributed, and the image has a high degree of visual texture.

The Haralick features are extracted between pairs of pixels for four directions ($0°$, $45°$, $90°$ and $135°$) and constant distance $d = 1$. Additionally, the average value (*avg*) for each Haralick feature is calculated.

Also these image features are translated into linguistic concepts to permit us to write rules in terms of easily-understood word descriptors, rather than in terms of numerical values.

The knowledge to be included in this part of the knowledge base was obtained by theoretical considerations and extensive experiments. We have used for the evaluation of interest operators several methods: (1) visual inspection, (2) ground-truth verification on the basis of good and bad areas, and (3) a new evaluation method by means of distances between sets of interest points. More detail about the evaluation of interest operators can be found in [13,10].

As an example, the following listing shows the rule for the selection of the Förstner interest operator:

```
(defrule foerstner
  (or (or (or (and (1_Haralick (H1_0_f low | very_low | middle))
                   (5_Haralick (H5_0_f low | very_low | middle)))
              (and (1_Haralick (H1_90_f low | very_low | middle))
                   (5_Haralick (H5_90_f low | very_low | middle))))
              (and (and (1_Haralick (H1_0_f low | very_low))
                        (1_Haralick (H1_90_f low | very_low)))
                   (not (Stat_Moments (M3_f very_high_negativ)))))
              (and (and (5_Haralick (H5_0_f low | very_low))
                        (5_Haralick (H5_90_f low | very_low)))
                   (not (Stat_Moments (M3_f very_high_negativ)))))
=>
  (assert (iop (foerstner yes))))
```

Like image processing algorithms in the previous section, a combination of suitable IOPs is selected together with its parameters. Operator parameter values are specified on the basis of image features and features collected by user-queries.

The data flow of the knowledge-based point detection is shown in the middle part of Figure 1. The knowledge base for the choice of suitable IOPs consists of about 20 rules.

In spite of choosing suitable image preprocessing algorithms and suitable interest operators, the number of detected points is often too high. In most cases, the elementary object structure can be represented by simple line geometry. Points detected apart from this line structure are undesirable and not useful for subsequent process steps, like object reconstruction or deformation analysis. Therefore, the next step has to be the reduction of the interest points detected apart from this line structure by a special point filter.

4 Point Filtering

The point reduction is not a removing process of undesirable points (all detected points will be preserved such that no information is lost), but is based on the weighting of each point. Only points with a specific minimal weight are considered to be useful. The filtering process (shown in the right part of Figure 1) is done by means of two methods:

(1) point filtering on basis of defined rules (knowledge-based),
(2) interactive point filtering (user-based).

4.1 Knowledge-Based Point Filtering

For this first method, several criterions are used to weighten each point. The most important are (1) how many *interest operators* detect the (same) point, and (2) which "property-parameters", obtained from the *interest operator(s)*, the point has.

The *first criterion* is very simple but effective. The point filter scans all point lists (one point list for each applied *interest operator*) and weights each point in correspondence with the number of *interest operators*, from which this point has been detected. In practice, this is a search routine which finds points with the same co-ordinates in different point lists. The weights are fixed on the basis of this simple coherence.

The *second criterion* is based on "property-parameters" (available in our implementation) obtained from the corresponding *interest operator* for each point, e.g., the Förstner and Harris operator return the standardized grey value (between 0 and 1) for the corresponding point, the Förstner operator provides the values for q and W for each point, etc. On the basis of these values, we can formulate several rules for point filtering respectively weighting.

After having applied the knowledge-based point filter on the point list(s), each point has been weighted. Finally, points with weight below a defined threshold are suppressed and not considered in future decision making. The complete knowledge base for point filtering consists of about 10 rules.

4.2 User-Based Point Filtering

The developed *interactive point filter* allows the user to choose the points or point clouds to be suppressed. This selection process is realized by means of a graphical user interaction. The user has to draw a rectangular window in the graphical output. Points inside these selected windows are considered to be not useful for the current task. In a final step the user has the possibility to display only points with the same weight and to remove point groups which are weighted differently. The described method enables

(a) (b)

Fig. 2. (a) Noisy underexposed image; (b) image after *image brightening* and a 3×3 *median filtering*

the user to adjust the *interest points* to a best qualification for all sorts of subsequent applications.

5 Experiments

We have tested our system on a benchmark suit of about 120 pictures, which include different kinds of buildings, lighting conditions, etc. In order to give an impression how the system works, we have chosen one example which is discussed in this section.

The picture in Figure 2 shows a noisy underexposed image. Relevant details (like corner points) in this image may not be (or not easily be) visible. In order to improve the visibility of relevant details, suitable image preprocessing and enhancement algorithms have to be used, which are selected by the KBS.

In this example, the system chooses the following algorithms from the extracted image features: *image brightening*, *gauss filtering*, *edge detection* and *thresholding*. Now, the KBS gives the user the possibility to overrule this decision. If the user deletes edge detection (for our example we assume that) from the list, then only *image brightening* and *median filtering* (which replace the *gauss filtering*) remain. The processed image is shown in Figure 2(b).

After preprocessing the image, its features will be recalculated. Furthermore, additional features are collected by user-queries (as mentioned above). Based on these features, a combination of IOPs, including their parameters, has been selected by the KBS: (1) the Förstner and (2) the Harris operator. The detected points are shown in Figure 3(a).

It can be seen that *interest points* are generally detected on the regular structure of the object. Only a small number of isolated single points are detected inside these “structure lines”. These points result from local grey-level differences, like “fault-pixels”. A more problematic area is the glass window, where many *interest points* are emerged by reflections. Changes of parameter values (of the *interest operators*) would remove the undesirable points on the glass windows, but the desired *interest points*,

(a) (b)

Fig. 3. (a) Interest points detected with combined Förstner and Harris operator; (b) final result after processing, application of *interest operators* and point filtering.

too (the grey-level differences in this area are the same as those of the "structure lines" of the facade). Undesirable points can only be suppressed by a suitable point filtering technique as described above. The resulting *interest points* are shown in Figure 3(b). Most of the undesirable points have been filtered by the rule-based filtering sequence; only a small number of points (inside the glass window) by user interaction.

We will conclude this section with some information about the runtime of the developed system. For an image with 8 bits-per-pixel (bpp) and 640×480 pixels the following runtime values [1] results: $\sim 2sec.$ for image analysis; $\sim 1sec.$ for image processing; $\sim 5sec.$ for the Förstner operator, $\sim 12sec.$ for the Harris operator, $\sim 1sec.$ for the HFVM operator; $\sim 1sec.$ for the knowledge-based point filtering. The runtime is correlated with image size and detected points; all listed values are mean values (deviations from the mean value are under 1 sec.) from more than 100 images.

6 Summary

In this paper, we have presented a method for selecting different algorithms for image preprocessing, image enhancement and IOPs in order to detect interesting points in a picture. Furthermore, a suitable point filtering procedure was presented. Decision-making is supported by a KBS. We have discussed the parameters which influence this decision. Finally, we showed the behavior of the system on an example.

As noted above, we have conducted extensive experiments with the KBS on about 120 pictures showing different kinds of buildings. The system yields good results and shows a reasonable performance (less than 15 seconds for a picture including the application of image processing, IOPs and point filter). The relative small number of necessary rules would permit to implement the whole knowledge base as a embedded system in the videometric system.

We also tried to solve the problems using a neural network approach. It is worth

[1] Calculated on a Personal Computer — Intel Pentium 4 with 1.3GHz and 512MB Ram

mentioning that this approach did not yield satisfactory results.

Currently, only a small number of image processing algorithms and interest operators are implemented. Future work will include several methods, e.g., image sharpening (highpass filtering), more edge detection algorithms and various interest operators. The knowledge base will be extended accordingly.

Furthermore, we will extend the KBS to a semi-automated system for monitoring displacements of buildings and deformation analysis. Additionally, an extension of the implemented image analysis process (low-level feature extraction) to a higher level of abstraction (low-level features have to be mapped to high-level features) will be envisaged. Furthermore, the use of case-based reasoning for formulation of the knowledge base might be useful [8].

Acknowledgment: This work is supported by the Austrian Science Fund (FWF) Project P14664 and P18286.

References

[1] CLIPS-Project-page: http://www.ghg.net/clips/CLIPS.html, 2004.

[2] Förstner, Wolfgang: Statistische Verfahren für die automatische Bildanalyse und ihre Bewertung bei der Objekterkennung und -vermessung, DGK, vol. C, no. 370, München, 1991.

[3] Haralick, Robert M. / Shapiro, Linda G.: Computer and Robot Vision, 1st Edition, Addison-Wesley Verlag, New York, 1993.

[4] Harris, Chris / Stephens, Mike: A combined corner and edge detector. In: Proceedings of the 4th ALVEY vision conference, Matthews (Editor), University of Manchester, 1988.

[5] Kahmen, Heribert / Niessner, Anton / De Seixas, Andrea: 3D Object Sensing Using Rotating CCD Cameras. In: Kropatsch / Bischof (Editors), Digital Image Analysis, 1st Edition, Springer, Berlin/Heidelberg/New York, 2001.

[6] Mischke, Alfred / Kahmen, Heribert: A New Kind of Measurement Robot System for Surveying of non Signalized Targets. In: Optical 3-D Measurement Techniques IV, Grün and Kahmen (Editors), Herbert Wichmann, Karlsruhe, 1997.

[7] Paar, Gerhard / Rottensteiner, Franz / Pötzleitner, Wolfgang: Image Matching Strategies. In: Kropatsch / Bischof (Editors), Digital Image Analysis, 1st Edition, Springer Verlag, Berlin/Heidelberg/New York, 2001.

[8] Perner, Petra: Why Case-Based Reasoning is Attractive for Image Interpretation. In: Aha / Watson (Editors), Case-Bases Reasoning Research and Developments, Springer Verlag, 2001.

[9] Pratt, William K.: Digital Image Processing, 1st Edition, John Wiley and Sons, New York/Chichester/Brisbane/Toronto, 1978.

[10] Reiterer, Alexander: A Knowledge-Based Decision System for an On-Line Videotheodolite-Based Multisensor System. PhD thesis, Vienna University of Technology, 2004.

[11] Reiterer, Alexander / Kahmen, Heribert / Egly, Uwe / Eiter, Thomas: Knowledge-Based Image Preprocessing for a Theodolit Measurement System, In: Optical 3-D Measurement Techniques VI - Volume I, Grün and Kahmen (Editors), Zurich, 183-190, 2003.

[12] Roic, Miodrag: Erfassung von nicht signalisierten 3D-Strukturen mit Videotheodoliten. PhD thesis, Vienna University of Technology, 1996.
[13] Schmid, Cordelia / Mohr, Roger / Bauckhage, Christian: Evaluation of interest point detectors, In: International Journal of Computer Vision - Volume 37/2, 151-172, 2000.
[14] Sokkia: http://www.sokkia.com/, 2004.
[15] Walser, Bernd / Braunecker, Bernhard: Automation of Surveying Systems through Integration of Image Analysis Methods. In: Optical 3-D Measurement Techniques VI, Grün and Kahmen (Editors), Volume I, Zurich, 191-198, 2003.
[16] Wasmeier, Peter: The Potential of Object Recognition Using a Servo-tacheometer TCA2003. In: Optical 3-D Measurement Techniques VI, Grün and Kahmen (Editors), Volume II, Zurich, 48-54, 2003.