Real-Time Object Recognition Using a Modified Generalized Hough Transform

MARKUS ULRICH[1,2], CARSTEN STEGER[2], ALBERT BAUMGARTNER[1] & HEINRICH EBNER[1]

Abstract: An approach for real-time object recognition in digital images based on the principle of the generalized Hough transform is proposed. It combines robustness against occlusions, distortions, and noise with invariance under rigid motion and local illumination changes. The computational effort is reduced by employing a novel efficient limitation of the search space in combination with a hierarchical search strategy using image pyramids. This approach uses the shape of the object, i.e., the edge information in the image, as feature and it is general with regard to the type of object.

1 Introduction

In many industrial applications, e.g., quality control, inspection tasks, or robotics, there is a particularly high demand on the object recognition approach to find the object in the image under certain aggravating circumstances. The recognition approach must fulfil real-time requirements, the method should be highly robust against occlusion and clutter and it should additionally be robust against non-linear contrast changes. Since in a great number of industrial applications the appearance of the object to be found has limited degrees of freedom, in this study only rigid motion, i.e., translation and rotation, is considered, which is sufficient in many cases.

In the literature different object recognition approaches can be found. All recognition methods have in common that they require some form of representation of the object to be found, which will be called *model* below. The model can be extracted, e.g., from a CAD representation or from one or more images, called *reference images*, of the object itself. The image, in which the object should be recognized, will be refered to as the *search image*. Almost all object recognition approaches can be split into two successive phases: the *offline phase* including the generation of the model and the *online phase*, in which the constructed model is used to find the object in the search image. Thus, only the computation time of the online phase is critical considering the real-time requirement.

One possibility to group object recognition methods is to distinguish between gray value based, e.g., GONZALEZ & WOODS (1992), BROWN (1992), LAI AND FANG (1999), and feature based strategies, e.g., BORGEFORS (1988), OLSON AND HUTTENLOCHER (1997). Gray value based matching has several disadvantages and does not meet most of the above mentioned demands. It is too computationally expensive for real-time applications and is not robust against occlusions or clutter. Features, e.g., points, edges, polygons, or regions characterize

[1] Lehrstuhl für Photogrammetrie und Fernerkundung, Technische Universität München, Arcisstr. 21, 80290 München, +49 89 289-22671, www.photo.verm.tu-muenchen.de

[2] MVTec Software GmbH, Neherstr. 1, 81675 München, +49 89 457695-0, steger@mvtec.com

the object in a more compressed and efficient way than the gray value information and thus are better suited for real-time recognition.

In our approach edges and their orientations, i.e., the shape of the object, are used as features. A representation (model) of the object is automatically generated solely from one reference image of the object itself. The model consists of the extracted object shape and the corresponding gradient directions along this shape. The basic principle of the generalized Hough transform (GHT) (BALLARD, 1981) is employed, which is an efficient method to compare the class of features used in this work and therefore allows a rapid computation. After analyzing the GHT and its major drawbacks in section 2 we further optimize the GHT by considering modifications to fulfil industrial demands (section 3). Experimental results and analyses concerning the achieved accuracy of the refined parameters complete this study (section 4).

2 The Generalized Hough Transform

2.1 Principle

A prominent property of the conventional Hough transform (HOUGH, 1962) is that its applicability is restricted to detect analytic curves. Therefore, BALLARD (1981) generalizes the Hough transform to detect arbitrary shapes. He also takes the edge orientation into account, which makes the algorithm faster and also greatly improves its accuracy by reducing the number of false positives.

To perform the offline phase of the GHT, the so-called *R*-table is constructed using information about the position and orientation of the edges in the reference image. The *R*-table is generated by choosing a reference point $\boldsymbol{o}$, e.g., the centroid of all edge points $\boldsymbol{p}_i^r$ ($i = 1...N_{\boldsymbol{p}^r}$) in the reference image, i.e., $x_o = 1/N_{\boldsymbol{p}^r} \sum x_{\boldsymbol{p}_i^r}$, $y_o = 1/N_{\boldsymbol{p}^r} \sum y_{\boldsymbol{p}_i^r}$, calculating $\boldsymbol{r}_i = \boldsymbol{o} - \boldsymbol{p}_i^r$ for all points and storing $\boldsymbol{r}$ as a function of the corresponding gradient direction Φ. If the orientation of the shape in the search image is not constant, i.e., the object may undergo rigid motions, for every possible orientation a separate *R*-table must be constructed. Assuming the case of rigid motion, in the online phase a three dimensional accumulator array *A* is set up over the domain of parameters, where the parameter space is quantized and range restricted. Each finite cell of that array corresponds to a certain range of positions and orientations of the reference image in the search image, which can be described by the three variables *x*, *y*, and θ. Here, *x* and *y* describe the translated position of $\boldsymbol{o}$ in the search image and θ the orientation of the object in the search image relative to the object in the reference image. For each edge pixel $\boldsymbol{p}_j^s$ in the search image and each *R*-table corresponding to one orientation θ_k, all cells $\boldsymbol{r}_i + \boldsymbol{p}_j^s$ in *A* receive a vote, i.e., they are incremented by 1, within the corresponding two dimensional hyper plane defined by $\theta = \theta_k$ under the condition that $\Phi_j^s = \Phi_i^r$. Maxima in *A* correspond to possible instances of the object in the search image.

2.2 Major Drawbacks

One weakness of the GHT algorithm is the - in general - huge parameter space. This requires large amounts of memory to store the accumulator array as well as high computational costs

in the online phase caused by the initialization of the array, the incrementation, and the search for maxima after the incrementation step. In addition, the accuracies achieved for the returned parameters depend on the quantization of translation and rotation. On the other hand, in practice the quantization cannot be chosen arbitrarily finely taking again memory requirements and computation time into account.

3. Optimizing the Generalized Hough Transform

The reduction of the high computational complexity of both, the conventional Hough Transform (HT) and the GHT, has been the subject of several publications. YACOUB & JOLION (1995), for example, propose an HT algorithm based on a hierarchical processing for line detection. It performs a classical HT on small subimages and merges the extracted lines with similar parameters by successively joining four neighboring subimages until the original image size is reached. In object recognition this approach is not reasonable because the object may be spread over several subimages, which results in a high sensitivity to clutter and noise. Other approaches reduce the dimension of the parameter space by introducing additional information: SER & SIU (1994) use relative gradient angles in the *R*-table, whereas MA & CHEN (1988) consider the slope and the curvature as local properties. These approaches have a reduced computational complexity in common but on the other hand have serious limitations. The use of relative gradient angles supposes the object not to be occluded, whereas considering the slope and the curvature fails when dealing with shapes that are composed mainly of straight lines. Additionally, the curvature is a very instable feature with regard to noise.

In this section we tackle the problems, which are mentioned in section 2.2: A hierarchical search strategy in combination with an effective limitation of the search space is introduced. Furthermore, a technique is presented to refine the returned parameters without noticeably decelerating the online phase. In addition, some quantization problems and their solutions are discussed.

3.1 Hierarchical Strategy

To reduce the size of the accumulator array and to speed up the online phase in our approach both, the model and the search image are treated in a hierarchical manner. First, an image pyramid of the reference image is generated. Every pyramid level of the reference image is rotated by all possible orientations, in which the object may appear in the search image, using $\boldsymbol{o}$ as fix point. Then, the gradient amplitude and the gradient direction are computed from the rotated image using the Sobel filter[1]. The edge pixels are extracted by thresholding the gradient amplitude. In the online phase the recognition process starts on the top pyramid level without any a priori information about the transformation parameters x, y and θ available. The cells in A that are local maxima and exceed a certain threshold are stored and used to initialize approximate values on the lower levels. Therefore, only on the top level an *R*-table is built for each rotation, whereas on the lower levels a modified strategy is necessary to take advantage of the a priori information returned from the next higher level.

[1] We prefer to use the Sobel filter because it represents a good compromise between computation time and accuracy. Its anisotropic response and its worse accuracy can be balanced by choosing an adequate quantization of the gradient directions (c.f. section 3.5).

3.2 Blurred Region

The use of a hierarchical model and the use of an image pyramid enable efficient limitations of the search space, because at lower levels approximate values $\tilde{x}$, $\tilde{y}$, and $\tilde{\theta}$ are known from a higher level. To obtain an optimal search region the model shape is blurred using the uncertainties of these a priori parameters. The proceeding is illustrated in the Figures 1, 2, and 3. The object is overlaid on the search image at the approximate position and orientation (Fig. 1). The positioning error $\delta x, \delta y$ is regarded by dilating the shape, i.e., the edge region, with a rectangular mask of size $(2\delta x+1)\times(2\delta y+1)$ (Fig. 2). The *blurred region* is finally obtained by successively rotating the dilated shape in both directions until the maximum amplitudes of the orientation error $\pm\delta\theta$ are reached, and merging the resulting regions (Fig. 3). The blurred regions are calculated for every quantized orientation in the offline phase and stored together with the model. In the online phase the blurred region enables us to restrict the edge extraction, which greatly reduces the computational effort. In addition, the size of the accumulator array *A* can be narrowed to a size corresponding to the uncertainties of the a priori parameters, which decreases the memory amount drastically.

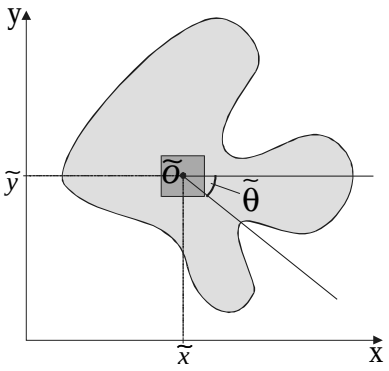

Fig. 1. Approximate values are given from the level above.

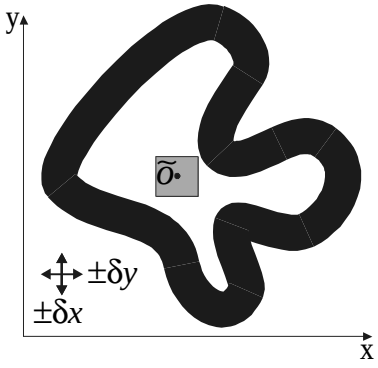

Fig. 2. Taking the translation error into account: Blurring by dilating.

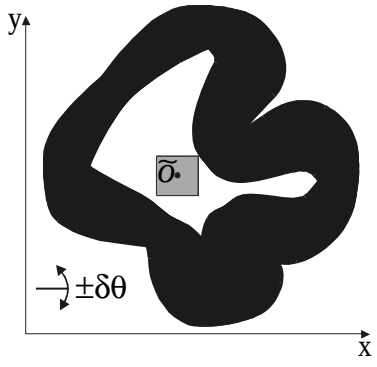

Fig. 3. Taking the orientation error into account: Blurring by rotating.

3.3 Tile Structure

After the edge extraction the third improvement is utilized. The principle of the conventional GHT is shown in Figure 4. The a priori information is displayed as a dark gray box representing the maximum error of the approximate translation values from the next higher level, which will be refered to as the *approximate zone*. The edge pixels $\boldsymbol{p}_1^r$, $\boldsymbol{p}_2^r$, and $\boldsymbol{p}_3^r$ have identical gradient directions. Thus, if any of these edge pixels is processed in the online phase of the conventional GHT each of the three vectors $\boldsymbol{r}_1, \boldsymbol{r}_2$, and $\boldsymbol{r}_3$ is added and the corresponding three cells are incremented, which is inefficient. One possible solution is to check during the voting process, whether the added vectors fall in the approximate zone or not. However, this query and the summation of the vectors would take too much time to allow for real-time operation. Therefore, the opposite approach is taken: Already in the offline phase the information about the position of the edge pixels relative to the centroid is calculated and stored in the model. This is done by overlaying a grid structure over the rotated reference image and splitting the image into tiles (Fig. 5). In the online phase the current tile is calculated using the approximate translation parameters and the position of the current edge pixel. Only the

vectors in the respective tile with the appropriate gradient direction are used to calculate the incrementation cells.

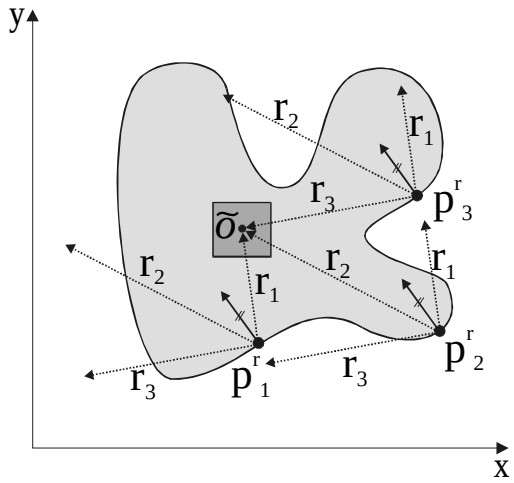

Fig. 4. The conventional GHT without using the a priori information of the translation parameters. Many unnecessary increments are executed.

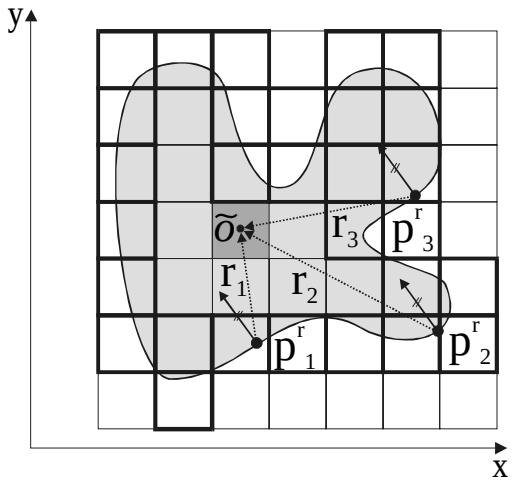

Fig. 5. Taking advantage of the a priori information using a tile structure. For each occupied tile (bold border) a separate R-table is generated.

3.4 Refinement of Position and Orientation

The accuracy of the results of the GHT on the lowest pyramid level depends on the chosen quantization of the parameter space. To refine the parameters of position and orientation we use the principle of the 3D facet model (HARALICK & SHAPIRO, 1992). The 3D parameter space *A* is assumed to be a 3D piecewise continuous intensity surface $f(x',y',\theta')$, in which the intensity values are represented by the number of entries in the cells of the accumulator array. The refinement of the parameters can be done by extrapolating the maximum of the continuous function in the neighborhood of the maximum of *A* (see also STEGER, 1998). This refinement is not expensive because the only thing to do is to solve a 3×3 linear equation system.

3.5 Problems and Solutions Concerning Quantization

When applying the principle of the GHT several problems occur concerning the quantization of the transformation parameters and of the gradient directions. A similar difficulty occurs using the tile structure described in section 3.3. In the following section we name these problems and present our solutions. A more detailed explanation is given in (ULRICH, 2001).

Rotation. In general, the step size $\Delta\theta$ for the discrete orientations must be chosen the smaller the bigger the searched object is. If $\Delta\theta$ is chosen too large, the maximum possible peak height $\Gamma^{\max}$ in *A* will be reduced. However, the computational effort Ω increases linearly with the number of discrete orientations. To find the optimum value for $\Delta\theta$ we have to minimize the computational effort $\Omega(\Delta\theta)$ while maximizing the peak height $\Gamma(\Delta\theta)$. The latter can be simulated using knowledge about the pixel distribution within the shape.

Translation. Under ideal conditions the peak in the parameter space is equal to the number of edge pixels contained in the model. If the object in the search image is translated by subpixel values in *x* and *y* direction relative to its position in the reference image the peak height

decreases because the votes are distributed over more than one cell in the accumulator array *A*. Under the assumption that the neighborhood of the peak is rotational symmetric the peak height can be made independent of subpixel translation by smoothing the translation hyper planes, i.e., $\theta = \text{const.}$.

Gradient Direction. The best suitable quantization of the gradient direction intervals within the *R*-table depends on various factors. The determination of the interval defines the range of gradient directions that are treated as equal. The smaller the interval the faster the computation. On the other hand, an interval that is chosen too small leads to instable results. The appropriate interval for the gradient direction depends on the variance of the gradient direction due to noise in the image and on the inherent absolute accuracy of the Sobel filter, i.e., the difference between the real partial derivatives and the Sobel response. These two effects are independent from the shape of an object and can be calculated precisely. Another factor that affects the gradient direction is subpixel translation. The gradient variation caused by subpixel translation (also due to small rotations) depends on the curvature along the shape, i.e., the curvature of the edge contours. One possible solution for this problem is to introduce only stable edge points into the model whose gradient direction at most vary in a small range. A final important detail is how to avoid boundary effects of the gradient intervals. This can be solved by establishing overlapping intervals.

Tile Structure. A problem similar to the quantization of the gradient directions occurs when using the tile structure described in section 3.3. The size of the tiles should be chosen such that the uncertainty of the approximate position is taken into account, i.e., the dimension of the tiles in the *x* and *y* direction should be $2\delta x$ and $2\delta y$. Furthermore, it must be ensured that an error of $2\delta x$ and $2\delta y$ of the approximate position $\tilde{\boldsymbol{o}}$ does not result in omitting the relevant edge pixels as a consequence of considering the wrong tile. This problem is solved by using overlapping tiles.

3.6 Memory Requirements and Computational Complexity

To facilitate the comparison of our object recognition method with other approaches, some statements about the memory requirements and the computational complexity of our implementation are given. The memory requirements of the model M^{model} and of the Hough parameter space in the online phase M^{Hough} can be calculated with the following formulas:

$$M^{\text{model}}[\text{byte}] = \left(\frac{32}{7} n_{\text{rot}}^0 \left(\frac{m^0}{k}\right)^2 \Psi \overline{n_{\text{grad}}} + 16 n_{\text{rot}}^0 n_{\text{e,m}}^0\right)\left(1 - \left(\frac{1}{8}\right)^L\right) + \frac{40}{3}\left(m^0\right)^2\left(1 - \left(\frac{1}{4}\right)^L\right) + 3 \cdot 2^{2-3L} n_{\text{e,m}}^0 n_{\text{rot}}^0 \quad (1)$$

$$M^{\text{Hough}}[\text{byte}] = 8^{-L} n_{\text{rot}}^0 \left(m^0 + s^0\right)^2, \quad (2)$$

where

L	is the number of pyramid levels decremented by one,
$n_{\text{e,m}}^0$	is the number of model edge pixels at the lowest pyramid level (i.e., original resolution),
n_{rot}^0	is the number of quantized rotation steps at the lowest pyramid level,
$\overline{n_{\text{grad}}}$	is the average number of quantized gradient directions through the pyramid,
m^0	is the size of the model in each dimension [pixel],
s^0	is the size of the search image in each dimension [pixel],

k is the size of the tiles in each dimension [pixel], and

Ψ is a factor [0…1], which describes the distribution of the edge pixels over the bounding box, i.e. Ψ is the fraction of occupied tiles.

The computational complexity Ω of the online phase can be described by

$$\Omega = 2^{1-5L} n_{\mathrm{e,m}}^0 n_{\mathrm{e,s}}^0 \frac{n_{\mathrm{rot}}^0}{\overline{n_{\mathrm{grad}}}} + 2^{1-2L} \left(n_{\mathrm{e,m}}^0\right)^2 \frac{n_{\delta\theta}}{\overline{n_{\mathrm{grad}}}} \cdot \frac{k^2}{\left(m^0\right)^2 \Psi} . \qquad (3)$$

Here,

$n_{\mathrm{e,s}}^0$ is the number of edge pixels in the search image at the lowest pyramid level and

$n_{\delta\theta}$ is the number of orientation steps on lower pyramid levels taking the range of uncertainty of the a priori orientation parameter into account.

Table 1 shows the memory requirements and the computational complexity for some examples, which are typical in practice. To show the efficiency of our approach the corresponding values according to the conventional GHT are listed likewise. For all examples a search image size of 600×600 pixels (s^0=600), a tile size of 7×7 pixels (k=7), a fraction of occupied tiles of Ψ=0.7, and an orientation uncertainty of ± 3 steps ($n_{\delta\theta}$=7) are employed.

L	$n_{\mathrm{e,m}}^0$	n_{rot}^0	$\overline{n_{\mathrm{grad}}}$	m^0	$n_{\mathrm{e,s}}^0$	M^{model} [MB]		M^{Hough} [MB]		Ω	
						MGHT	CGHT	MGHT	CGHT	MGHT	CGHT
3	2000	360	20	200	20000	30.8	4.3	0.5	460.8	$0.066 \cdot 10^6$	$1080 \cdot 10^6$
1	2000	360	20	200	20000	28.0	4.3	28.8	460.8	$67 \cdot 10^6$	$1080 \cdot 10^6$
3	**1000**	360	20	200	20000	25.1	2.2	0.5	460.8	$0.033 \cdot 10^6$	$540 \cdot 10^6$
3	2000	**180**	20	200	20000	15.7	2.2	0.2	230.4	$0.033 \cdot 10^6$	$540 \cdot 10^6$
3	2000	360	**30**	200	20000	40.2	4.3	0.5	460.8	$0.044 \cdot 10^6$	$720 \cdot 10^6$
3	2000	360	20	**100**	20000	16.3	4.3	0.3	176.4	$0.066 \cdot 10^6$	$1080 \cdot 10^6$
3	2000	360	20	200	**50000**	30.8	4.3	0.5	460.8	$0.165 \cdot 10^6$	$2700 \cdot 10^6$

Tab. 1. Memory requirement and computational complexity for different typical situations. Our approach using a modified GHT (MGHT) is compared with the conventional GHT (CGHT).

At the expense of a higher model size our approach drastically reduces the memory requirement and the computational complexity of the online phase in contrast to the conventional GHT.

4 Experimental Results

To validate the accuracy of the resulting parameters x, y, and θ we generated some image sequences (652×494 pixels) containing subpixel translations and rotations of an object. The experiments, which are illustrated in (ULRICH, 2001) in more detail, show that our approach is able to locate objects with a maximum mean error of about 0.1 pixels in position and 0.12 degrees in orientation. After adding white noise with a maximum amplitude of ± 5 to the search image these values degraded only slightly. Furthermore, the approach is robust considering that occlusions merely decrease the peak in the accumulator array proportional to the percentage of occlusion. To show the real-time capability: the average time needed to find an object of size 240×130 pixels containing approximately 3000 model points in the lowest pyramid level is about 60 msec on a PENTIUM III with 667 MHz.

5 Summary

By using a hierarchical search strategy in combination with a new effective search space limitation our approach fulfils the requirements of real-time. Since the object's shape does not depend on the illumination, this method in addition is robust against illumination changes to a certain extent. Furthermore, it is extremely robust against partial occlusion and clutter, as the raw gray value information is not used directly. The coarse solution of the position and orientation parameters of the object is adjusted in a subsequent refinement to meet the demands for high precision and accuracy in industrial applications.

6 References

BALLARD, D. H. (1981): Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition, **13**(2), p. 111-122.

BORGEFORS, G. (1988): Hierarchical chamfer matching: A parametric edge matching algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, **10**(6), p. 849-865.

BROWN, L. G. (1992): A survey of image registration techniques. ACM Computing Surveys, **24**(4) p. 325-376.

GONZALEZ, R. C. & WOODS, R., E. (1992): Digital Image Processing, p. 583-586, Addison-Wesley Publishing Company.

HARALICK, R. M. & SHAPIRO, L. G. (1992): Computer and Robot Vision. **Volume 1**, p. 371-452, Addison-Wesley Publishing Company.

HOUGH, P. V. C. (1962): Method and means for recognizing complex patterns. U.S. Patent 3,069,654.

LAI, S.,FANG, M.. (1999): Accurate and fast pattern localization algorithm for automated visual inspection. Real-Time Imaging, **5**, p. 3-14.

MA, D. & CHEN, X. (1988): Hough Transform Using Slope and Curvature as Local Properties to Detect Arbitrary 2D Shapes, Proc. 9th Int. Conf. on Pattern Recognition, p. 511-513

OLSON, C. F. & HUTTENLOCHER, D. P. (1997): Recognition by Matching With Edge Location and Orientation. Automatic target recognition by matching oriented edge pixels. IEEE Transactions on Image Processing, **6**(1), p. 103-113.

RUCKLIDGE, W. J. (1997): Efficiently locating objects using the Hausdorff distance. International Journal of Computer Vision, **24**(3), p. 251-270.

SER, P.-K. & SIU, W.-C. (1994): Non-analytic Object Recognition Using the Hough Transform with Matching Technique, IEE Proc. Part E, Computers and Digital Techniques **141**(1), p. 231-235.

STEGER, C. (1998): Unbiased Extraction of Curvilinear Structures from 2D and 3D Images. Fakultät für Informatik, Technische Universität München, Dissertation, p. 92-96, Herbert Utz Verlag.

ULRICH, M. (2001): Real-Time Object Recognition in Digital Images for Industrial Applications, Technical Report PF-2001-01, Chair for Photogrammetry and Remote Sensing, Technische Universität München.

YACOUB, S. B. & JOLION, J.-M. (1995): Hierarchical Line Extraction, IEE Proc.-Vis. Image Signal Process., **142**(1), p.7-14.