Global Curvature Estimation for Corner Detection

Simon Hermann[+] and Reinhard Klette[*]

[+] Institut für Angewandte Informatik, Mathematische Fakultät der Universität Göttingen
D-37083 Göttingen, Germany

[*]CITR, Department of Computer Science
The University of Auckland, Auckland, New Zealand

Abstract

The paper starts with presenting three curvature estimators which follow definitions (approaches) in differential geometry. Digital-straight segment (DSS) approximation is used in those estimators, we point to problems caused by this approach, and propose simple ways for eliminating those problems. The paper then informs about multigrid analysis experiments, where all estimators appear to be multigrid convergent when digitizing an ellipse. The paper also applies these estimators for corner detection and compares their performance with a recently published heuristic corner-detection approach by means of multigrid analysis. Experiments indicate that corner detectors (based on curvature estimation) perform about as good as the heuristic method for large grid resolutions, and one detector might be even superior.

Keywords: curvature estimation, corner detection, digital curve segmentation

1 Introduction

'Corner' or 'dominant point' detection is important for pattern or picture analysis. Many approaches have been proposed, often based on heuristics. We are particularly interested in those approaches which follow (i.e., by means of digitization) defined mathematical concepts in continuous spaces, and how they perform compared to heuristically defined corner detectors. Higher resolution pictures support the applications of methods derived from differential geometry.

A *corner* is (informally) defined as a high-curvature point on a simple digital arc or curve. Corners can be used to segment arcs or curves, for example in concave and convex segments.

We refer to [6] for a recent comparison of algorithms and methods for corner detection. Algorithms are classified in this unpublished PhD with respect to underlying methodology, which is often based on heuristics rather than on curvature definitions in geometry. See [7] for an early example. ([5] contains a review of [6].)

The following section describes three methods for estimating curvature along a digitized smooth curve based on definitions in differential geometry and compares those in an experiment focused on multigrid convergence. Section 3 applies these curvature estimators for corner detection, and experiments on multigrid convergence also include a heuristic method published in [1].

2 Curvature Estimation

We consider simple 8-curves ρ in the digital plane $\mathbb{Z}^2$. Pixels p_i in such a digital curve $\rho = p_0, p_1, \cdots, p_{n-1}$ have coordinates (x_i, y_i). Subscripts are modulo n; for example, pixel p_{i-k} with $i-k<0$ coincides with pixel $p_{n-1+i-k}$.

In order to detect a corner at pixel p_i on a curve ρ, it is common practice that a corner detector considers an angular measure based on a predecessor p_{i-k_b}, p_i itself, and a successor p_{i+k_f}, where $k_b, k_f > 0$ are fixed (e.g., both equal to $k = 0.02 \cdot n$) or variables within a defined interval. Angular measures resulting for possible values of k_b and k_f are taken into account to identify p_i as a corner or not.

Non-adaptive specifications of possible values of k_b or k_f do not reflect the shape of the given digital curve. For example, a fixed value of k (e.g. $k = 6$) defines a local approach for corner detection. Adaptive specifications of k_b or k_f can be based, for example, on digital straight segment or DSS approximation; see [2, 4]. [1] The benefit of this approach is to have uniquely defined k_b and k_f for

[1]For DSS approximation we apply the linear-time algorithm **DR1995** of [3]. This algorithm is based on arithmetic geometry. Let $\mu \in \mathbb{Z}$, and a and b be relatively prime integers. The set

$$D_{a,b,\mu} = \{(i,j) \in \mathbb{Z}2 : \mu \leq ai + bj < \mu + \max\{|a|,|b|\}\}$$

is a DSS. Algorithm **DR1995** segments a digital curve into a sequence of subsequent maximum-length DSSs.

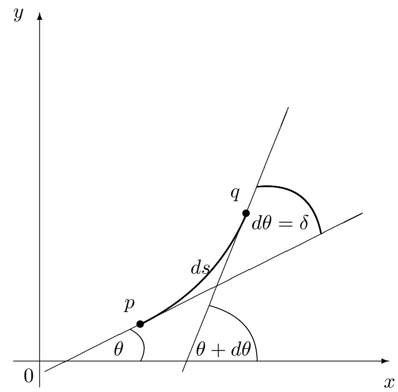

Figure 1: Tangent based curvature estimation.

every point p_i, and those values reflect the shape of the curve.

2.1 Derivative of Tangent Angle

The curvature estimation method of [4] follows the definition of curvature based on changes in orientations of the tangent.

Let p and q be two points on a plane curve, and δ the angle between positive directions of both tangents at those points (see Figure 1). Curvature κ at p is defined to be the limit

$$\kappa(p) = \lim_{pq \to 0} \frac{\delta}{pq}$$

Algorithm **HK2003** uses backward (ending at p_i) and forward (beginning at at p_i) DSSs for approximating the tangent at p_i.

Algorithm 1 Curvature Estimation **HK2003**

Compute-curvature(Curve ρ)

For point p_i in ρ **do**

compute k_b and k_f with **DR1995**

$l_b = d_2(p_{i-k_b}, p_i)$ and $\theta_b = \tan^{-1}\left(\frac{|x_{i-k_b} - x_i|}{|y_{i-k_b} - y_i|}\right)$

$l_f = d_2(p_{i+k_f}, p_i)$ and $\theta_f = \tan^{-1}\left(\frac{|x_{i+k_f} - x_i|}{|y_{i+k_f} - y_i|}\right)$

compute $\theta = \frac{1}{2} \cdot \theta_b + \frac{1}{2} \cdot \theta_f$

compute $\delta_b = |\theta_b - \theta|$ and $\delta_f = |\theta_f - \theta|$

return $\frac{\delta_b}{2l_b} + \frac{\delta_f}{2l_f}$

(Note that $\delta_b = \delta_f$.)

We use this algorithm (without alterations) as proposed in [4]. Only positive values are returned and therefore we are not able to obtain information about convexity or concavity.

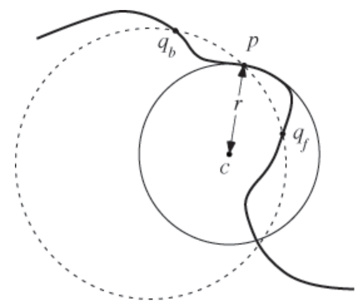

Figure 2: The dashed circle is incident with points q_b, p, and q_f; it 'moves' into the osculating circle centered at c.

2.2 Radius of Osculating Circle

The osculating circle at a point p on a smooth curve γ can be defined in differential geometry by a circle that intersects γ at p and two points p_b and p_f (left and right of p). Moving both points into p results into the osculating circle at p with center c (see Figure 2). The absolute value of curvature at point p is then defined as the reciprocal value of the radius $r = d_2(c, p)$.

The following calculation of the osculating circle makes use of the geometric property that three points uniquely define a circle. At point p_i we compute two DSSs as in **HK2003**. The algorithm is as follows:

Algorithm 2 Curvature Estimation **HK2005**

Compute-curvature(Curve ρ)

For point p_i in ρ **do**

compute k_b and k_f (with **DR1995**)

compute bisecting lines g_b and g_f of segments $p_{i-k_b}p_i$ and $p_{i+k_f}p_i$

compute c as intersection of g_b and g_f

compute radius $r = d_2(c, p_i)$

return $\frac{1}{r}$

2.3 Derivative of Curve

A parametrized curve $\gamma(t) = (x(t), y(t))$ allows to calculate curvature based on derivatives; the curvature is as follows

$$\kappa = \frac{\begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}}{(x'^2 + y'^2)^{\frac{3}{2}}} \tag{1}$$

[6] proposes to use second order polynomials to approximate the digital curve ρ locally at p_i by using also pixels p_{i-k_b} and p_{i+k_f}. The approximating polynomial $\gamma(t) = (x(t), y(t))$ is defined by

$$\begin{aligned} x(t) &= a_2 t^2 + a_1 t + a_0, \text{ and} \\ y(t) &= b_2 t^2 + b_1 t + b_0 \end{aligned}$$

with $t \in [-1, 1]$. Let $t = -1$ define p_{i-k}, $t = 0$ specifies pixel p_i, and $t = 1$ defines p_{i+k}. In this particular case, Equation (1) takes the following form

$$\kappa = \frac{2(a_1 b_2 - a_2 b_1)}{(a_1^2 + b_1^2)^{\frac{3}{2}}} \tag{2}$$

at point p_i.

Values of a_1, a_2, b_1 and b_2 follow from the equational system

$$\begin{aligned} a_2 - a_1 + a_0 &= x_{i-k_b} \\ a_0 &= x_i \\ a_2 + a_1 + a_0 &= x_{i+k_f} \end{aligned}$$

and analogously for y; we obtain

$$\begin{aligned} a_1 &= \frac{x_{i+k_f} - x_{i-k_b}}{2} \\ a_2 &= \frac{x_{i+k_f} + x_{i-k_b}}{2} - x_i \\ b_1 &= \frac{y_{i+k_f} - y_{i-k_b}}{2} \\ b_2 &= \frac{y_{i+k_f} + y_{i-k_b}}{2} - y_i \end{aligned}$$

Those values are used in Equation (2).

2.4 Multigrid Analysis

We perform multigrid experiments for those three estimators in which we digitize elliptical discs of a certain shape with increasing grid resolution. We use Gauss digitization for digitizing these elliptical discs

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1, \text{ where } a = 2 \cdot b \text{ and } 10 \leq b \leq 520$$

We extract 8-curves via border tracking of those digital ellipses which are the input for curvature estimators.

For every resolution b, we compute the mean m_b of absolute errors of estimated curvature at every border pixel. Resulting scattered points are filtered by a sliding mean using $\frac{1}{39}\sum_{i=-19}^{19} m_{b+i}$ and drawn in increments of 5 into the diagrams shown in Figures 3 and 4.

The curvature at $p = (x, y)$ on the frontier of the elliptical disk is

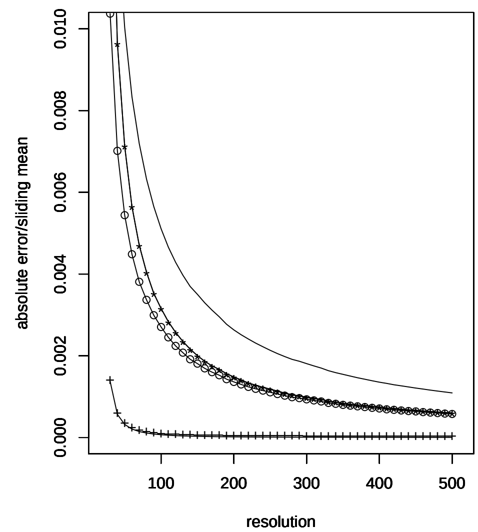

Figure 3: Error curves for DSS-based estimators.

$$\frac{1}{\kappa} = a^2 b^2 \left(\frac{x^2}{a^4} + \frac{y^2}{b^4} \right)$$

How to find a border pixel p_i the corresponding point $p = (x, y)$ on the ellipse in order to compute the absolute error?

A first option is that we identify p with p_i. A second option is that we choose p as the intersection of the ellipse $\frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1$ with the straight line $y = \frac{y_i}{x_i} x$.

We can see from Figure 3 that all three estimators seem to be multigrid convergent. The smooth

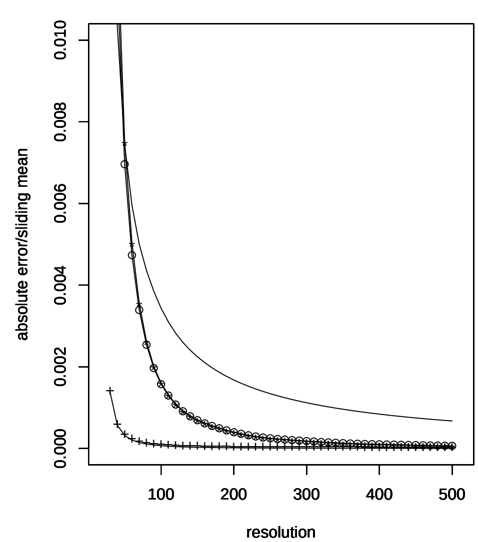

Figure 4: Error curves using just $k = 0.02n$.

curve represents **HK2003**, the curve with circles **HK2005**, and the curve with stars represents **M2003**. The line with plus signs shows the difference between first and second option of error measurements, which proves to be of marginal impact. Errors are close to zero for $b > 200$.

Alternatively, we replaced all DSS-based calculated values of k_b and k_f simply by a uniform value of $k = 0.02 \cdot n$. The results are shown in Figure 4. We obtain slightly better results for all estimators! This is probably due to the fact that an ellipse is such a 'uniformly smooth' curve. We will see later that derived corner detectors perform equally good when applying curvature estimators being either DSS-based or using uniformly a constant such as $k = 0.02n$.

2.5 Possible Alterations

We noticed that the performance of DSS-based curvature estimators is influenced by different types of problems; both may appear as minor, but they have definitely impacts on convergence analysis (as already discussed in [4]).

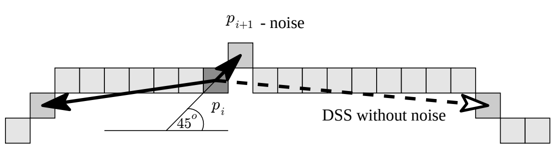

Figure 5: Single pixel, not aligned with a DSS.

(i): Suppose we have a curve segment with small curvature but containing a single pixel p_{i+1} which is 'not aligned with the segment' (e.g., to be considered as noise), see Figure 5. If a DSS ends at p_i, then the next DSS will be the segment $\overline{p_i p_{i+1}}$, which forms an angle of about 45° with the previous DSS, i.e., this will fall into the category of 'high curvature'. However, in general we rather prefer to ignore this 'noisy pixel' p_{i+1}. The conclusion here could be that we do not use DSSs of length 2 for defining k_b or k_f, but use $k = 0.02 \cdot n$ in this case.

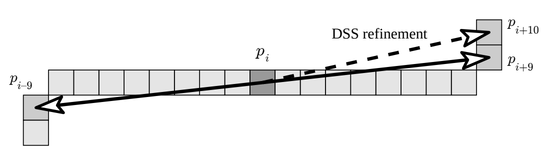

Figure 6: Parallel problem for DSS

(ii): Another problem is that we do not get an angle $\neq \pi$ at p_i if it is at a center position on a DSS. If we do accuracy experiments for digitized smooth curves (with non-zero curvature everywhere), then this will always result into an error at p_i. We could enforce non-zero curvature between $\overline{p_{i-k_b} p_i}$ and $\overline{p_i p_{i+k_f}}$ by 'adding' a pixel to the second end of such a DSS as shown in Figure 6.

Both alterations reduce the measured errors of DSS-based curvature estimators for digitized ellipses. However, if curvature estimators are used for corner detection, then the following section shows that we do not really need such alterations.

3 Corner Detection

We compare corner detectors based on the three curvature estimators discussed before with the heuristic approach of [1] (which proved to be of good performance in a particular application [8]).

3.1 A Heuristic Approach

A first run of **CS1999** through all pixels of a digital curve identifies all potential candidates of corners. To decide wether p_i is a potential corner, consider the set T_i of all triples $(p_{i-k_b}, p_i, p_{i+k_f})$ with $k_b, k_f > 0$ such that

$$d_{min} \leq d_2(p_i, p_{i+k_f}) \leq d_{max} \text{ and}$$
$$d_{min} \leq d_2(p_i, p_{i-k_b}) \leq d_{max}$$

where d_{min} and d_{max} are fixed thresholds (we used the default setting of $d_{min} = 7$ and $d_{max} = d_{min} + 2$; another option could be, for example, $d_{min} = 0.02n$). Let $\tau \in T_i$ and $a = d_2(p_i, p_{i-k_f})$, $b = d_2(p_i, p_{i+k_b})$, and $c = d_2(p_{i-k_b}, p_{i+k_f})$. Then

$$\alpha_\tau = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

is the angle between $\overline{p_i p_{i+k_f}}$ and $\overline{p_i p_{i-k_b}}$. Pixel p_i is a potential corner if $min\{\alpha_\tau : \tau \in T_i\} < \alpha_{max}$ where α_{max} is a third fixed threshold (default is $\alpha_{max} = 150$) of this heuristic method.

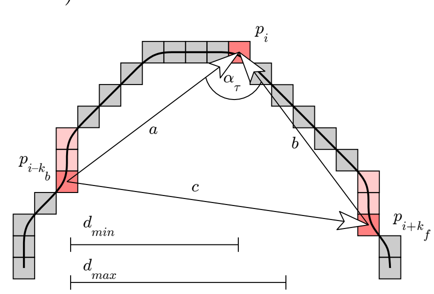

Figure 7: Defining potential corners for **CS1999**

A second run of **CS1999** through all pixels of the given digital curve deals with all situations where more than one pixel 'responded' to the same corner. Pixels p_i and p_j are neighbors iff $d_2(p_i, p_j) \leq d_{max}$. Compare each potential corner p_i with all the neighboring potential corners, and only keep it as a detected corner if its α-angle defines a minimum in this neighborhood.

3.2 Curvature-Based Corner Detection

A first run detects potential corners; a pixel p_i is a potential corner if the estimated curvature $\kappa(p_i)$ exceeds a given threshold κ_t.

In a second run we discard all potential corners which have a potential corner with higher curvature in its neighborhood. The neighborhood is defined as in **CS1999**, using the same distance threshold d_{max}.

3.3 Multigrid Analysis

This experiment aims at analyzing up to what angle a detector has the potential ability to detect a corner. Therefore we set up a perfect and measurable environment, where it should be possible to assign a detected corner to a specific corner of a synthesized object.

We generate a digital spiral defined by a parameter $\ell \in \{50, 60, ..., 300\}$: successively draw lines of length ℓ with angles $\alpha = 90$, 45, 40, 35, 30, 25, 20, 15, and 10 degrees, defining corners at all vertices. We use $d_{max} = \frac{\ell-2}{2}$ as the neighborhood threshold for corner detectors as in Figure 8.

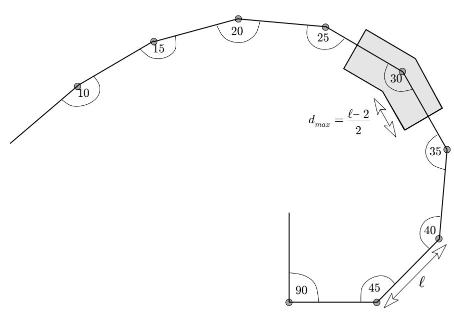

Figure 8: Corner detection multigrid experiment. Within the gray neighborhood only one corner can be detected, due to the parameter d_{max}.

A corner detector identifies a corner p_o correctly if it detects a pixel p_e with $d_2(p_o, p_e) \leq \frac{\ell}{10}$. (We do not have to test more than one detected corner p_e due to the chosen neighborhood threshold.)

For every ℓ, we initialize an error measure with zero. For every correctly detected corner p_o we increment this measure by $d_2(p_o, p_e)$, otherwise (no corresponding p_e detected) we increment by $\frac{\ell}{10}$.

In **CS1999** we used default values of d_{min} and d_{max} for the first run, but $d_{max} = \frac{\ell-2}{2}$ in the second run. Since we want to detect all corners defined by an angle of 10 degrees or more, we use $\alpha_{max} = 170$. For the curvature estimators we use $\frac{1}{\kappa_t} = 2n$ to make sure that all corners are detected.

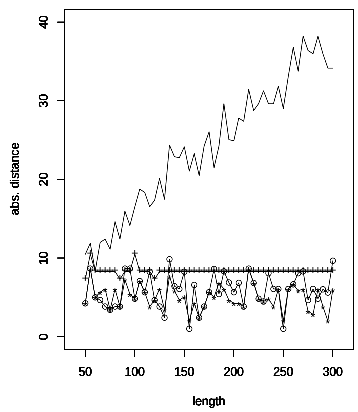

Figure 9: First experiment ($k = 0.02n$).

In a first experiment we compare results obtained with curvature estimator using $k = 0.02n$, and in a second experiment DSS-based estimators. The resulting diagrams in Figures 9 and 10 do not use sliding means. Errors at $\ell \in \{50, 60, ..., 300\}$ are connected by straight segments. The smooth polygonal line represents results of the **HK2003** corner detector, the line with circles those of the **HK2005** corner detector, the line with stars those of the **M2003** corner detector, and the line with plus signs represents those of **CS1999**.

Figure 9 illustrates the case of using $k = 0.02n$; **HK2005**, **M2003** and **CS1999** ensure equally good results while errors of the corner detector based on **HK2003** are diverging for increasing resolution.

In case of using DSS-based curvature estimators (Figure 10) we notice that errors of all estimators (for varying resolution) are in one interval; for **M2003** there might be even a convergence towards zero error. The experiments indicate that corner detectors using DSS-based estimators perform better than those with $k = 0.02n$. Problems with the latter choice are illustrated in Figure 11, where 'spiraling curves' may actually exclude relatively large values defined by the uniform rule $k = 0.02n$. A curvature estimator using $k = 0.02n$ would return a higher curvature value than one using DSS-segmentations.

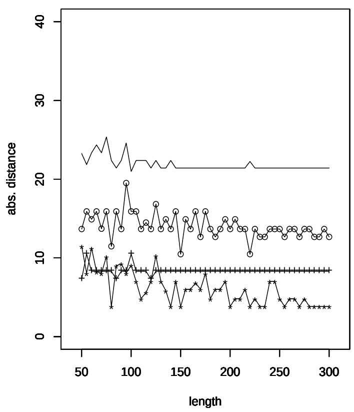

Figure 10: Second experiment (DSS-based).

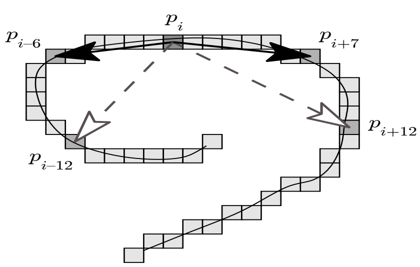

Figure 11: Dashed line for $k = 0.02n$ and $n = 600$.

The curve used in Figures 12 to 14 was generated manually. The neighborhood threshold was $d_{max} = 9$ for all detectors. For **CS1999** we used $\alpha = 135$, and for the curvature-based detectors we used $\frac{1}{\kappa} = 0.02 \cdot n$ to be consistent with those using $k = 0.02n$. The results for **HK2005**, **M2003** and **CS1999** seem to be equally good, whether DSS-based or not. (We are not showing results for **M2003** since they are very similar to those of **HK2005**).

4 Conclusions

Our experiments support in general the hypothesis that DSS-based curvature estimators (and derived corner detectors) improve with increases in grid resolution. The corner detector using the DSS-based version of **M2003** seems to be the best

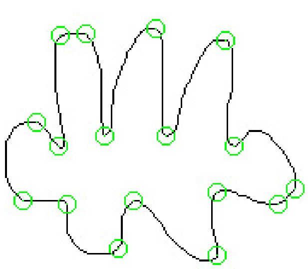

Figure 12: Detected corners using **CS1999**.

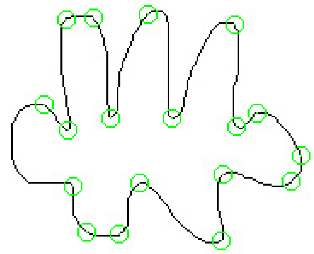

Figure 13: Corners using **HK2005** (DSS-based).

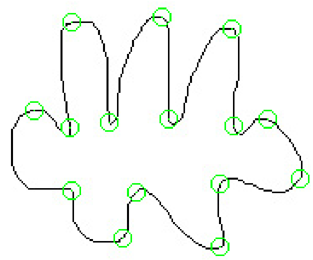

Figure 14: Corners using **HK2005** ($k = 0.02n$).

choice of all the corner detectors compared in our experiments. Of course, this may vary with selecting different types of digital curves, and a wider study might be in place.

References

[1] D. Chetverikov ad Z. Szabo. A simple and efficient algorithm for detection of high curvature points in planar curves. In Proc. *Workshop Austrian Pattern Recognition Group*, page 175–184, 1999.

[2] D. Coeurjolly, S. Miguet, and L. Tougne. Discrete curvature based on osculation circle estimation. In Proc. *Int. Workshop Visual Form*, LNCS 2059, pages 300–312, Springer, Berlin, 2001.

[3] I. Debled-Rennesson and J.-P. Reveillès. A linear algorithm for segmentation of digital curves. *Pattern Recognition*, **9**:635–662, 1995.

[4] S. Hermann and R. Klette. Multigrid analysis of curvature estimators. In Proc. *Image Vision Computing New Zealand*, pages 108–112, 2003.

[5] R. Klette and A. Rosenfeld. *Digital Geometry – Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, San Francisco, 2004

[6] M. Marji. On the detection of dominant points on digital planar curves. PhD thesis, Wayne State University, Detroit, Michigan, 2003.

[7] A. Rosenfeld and J.S. Weszka. An improved method of angle detection on digital curves. *IEEE Trans. Computers*, **24**:940–941, 1975.

[8] B. Rosenhahn, L. He, and R. Klette. Automatic human model generation. In Proc. *Computer Analysis Images Patterns*, LNCS 3691, pages 41–48, 2005.