

A Circle-based Vectorization Algorithm for Drawings with Shadows

Alexandra Bonnici * and Kenneth Camilleri
University of Malta

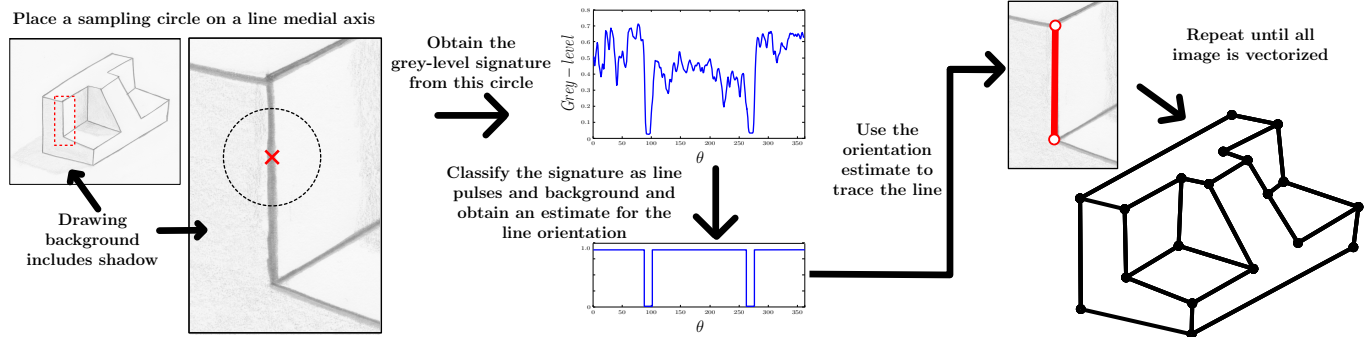


Figure 1: Circle-based vectorization

Abstract

Vectorization algorithms described in the literature assume that the drawings being vectorized are either binary images or have a clear white background. Sketches of artistic objects however also contain shadows which help the artist to portray intent, particularly in potentially ambiguous sketches. Such sketches are difficult to binarise since the shading strokes make these sketches non bimodal. For this reason, we describe a circle-based vectorization algorithm that uses signatures obtained from sample points on the line strokes to identify and vectorize the line strokes in the sketch. We show that the proposed algorithm performs as well as other vectorization techniques described in the literature, despite the shadows present in the sketch.

CR Categories: I.4.3 [Image Processing and Computer Vision]: Enhancement—Grayscale manipulation I.4.6 [Image Processing and Computer Vision]: Segmentation—Edge and feature detection;

Keywords: vectorization, shadows, sketches

1 Introduction

Sketches of three dimensional objects can be ambiguous particularly since the spatial relation between different objects is lost in the flat representation of the scene. Artists and designers therefore introduce a number of cues, among which are shadows or shading which would help the artist portray the desired spatial representation in the sketch [Oliver 1979]. These shadows help the machine interpretation of the sketch. In particular, shadows can be used as additional constraints in the line-labelling interpretation of sketches, limiting the possible interpretations of the edges they bear upon. In [Bonnici and Camilleri 2012] we show how the shadows can be used in conjunction with trihedral junctions dictionaries to obtain the desired interpretation of the sketch. However, the presence of shadows poses an additional burden on the interpretation of the sketch since it is now necessary to distinguish between ink markings that form the shading and those that form the line strokes.

In online sketch based interfaces, this problem can be resolved if

temporal information as well as pen-contact and pen pressure are available, since the drawing action for shading and line strokes is different. In off-line sketching, and in particular in paper-based sketching, such information is unavailable such that all ink markings are indistinguishable from one another.

In [Bonnici and Camilleri 2012] the separation of the ink strokes from the line strokes was achieved by requiring that the artist sketches the strokes and shading in different colours. While such a method does allow an easy distinction between the two, it shifts the burden onto the artists, who do not generally use different colours for edges and shading. For this reason, an alternative method for acquiring the line strokes from the shading strokes is required in order to successfully interpret artistic sketches.

In this paper, we build on the concept introduced in [Bonnici and Camilleri 2009] and use circles to sample the sketched drawing, obtaining from each sampling circle a grey-level signature from which we can discriminate between line strokes and surrounding shading strokes and hence obtain line vectors directly from the grey-level representation of the sketched drawing. The rest of this paper is organized as follows; Section 2 gives a review of vectorization algorithms described in the literature, Section 3 introduces the concept of the circle signature, Section 4 describes how this signature can be used to vectorise drawings with shading, Section 5 presents the evaluation methodology, Section 6 presents the results obtained while Section 7 concludes the paper.

2 Related Work

In off-line interpretation algorithms, line strokes are identified using vectorization techniques whose scope is to locate the line strokes and represent these strokes as vectors which can be interpreted by computer-aided design (CAD) tools [Tombré et al. 2000]. Vectorization algorithms have been used in applications involving engi-

*e-mail:alexandra.bonnici@um.edu.mt

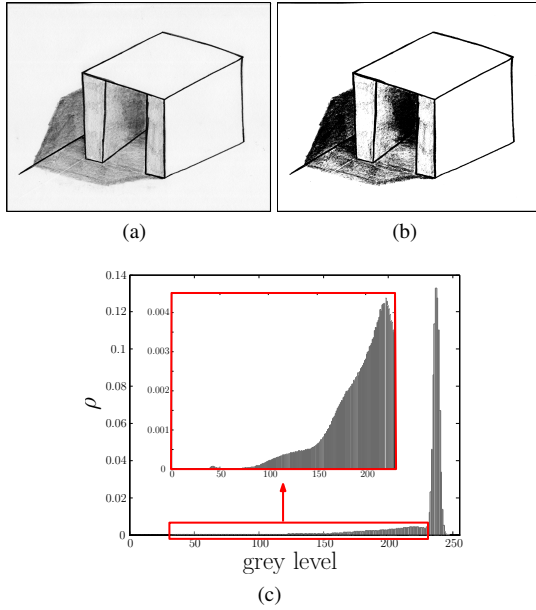


Figure 2: A sketch with shading (a) is not bimodal as can be evidenced from the shape of its histogram (c). As a result, binarisation techniques which segment the image pixels into a foreground and a background classes do not allow for distinction between the sketched strokes, the shading and the background, erroneously clustering the shading strokes with either the foreground class or the background class.

neering drawings [Liu and Dori 1999], architectural drawings [Hilaire and Tombre 2006] and maps [Rööslü and Monagan 1996]. In such drawings, the task of the line location problem is to identify the smallest group of pixels that represent the drawing’s topology, and this is often referred to as the medial axis of the drawing [Katz and Pizer 2004]. This medial axis can be determined using a variety of techniques among which are algorithms based on skeletonization [Hilaire and Tombre 2006], drawing contours [Ramel et al. 1998], the Hough-transform [Song and Lyu 2005], run-graphs [Keysers and Breuel 2006] and sparse-pixel sampling techniques [Song et al. 2002]; [Bonnici and Camilleri 2009]. In all three methods however, there is the underlying assumption that the drawing can be easily separated into two classes, namely a foreground class, containing the inked edge stroke pixels and a background class containing all the remaining pixels. Such a separation into foreground and background classes is typically achieved through binarisation [Davies 2004] or by assuming that the background class consists solely of pixels with uniform grey level. This is possible because the images vectorized by these algorithms are generally bimodal. However, the introduction of shading strokes to the drawing forces the drawing to loose its bimodality such that, as shown in Figure 2, the use of a threshold to separate the image grey levels into two classes is not suitable to distinguish between the line strokes, the shaded regions and the image background.

Such a two-tier approach to the line location has further disadvantages, particularly since errors introduced in the binarisation process will inevitably effect the outcome of the vectorisation. In addition, the binarisation algorithms are applied on a pixel-by-pixel basis, whereas the decision on whether a pixel forms part of the background (and shading) or the line strokes would be more informed if this is driven by decisions made on other line stroke pixels. This calls for a vectorization algorithm that locates the lines

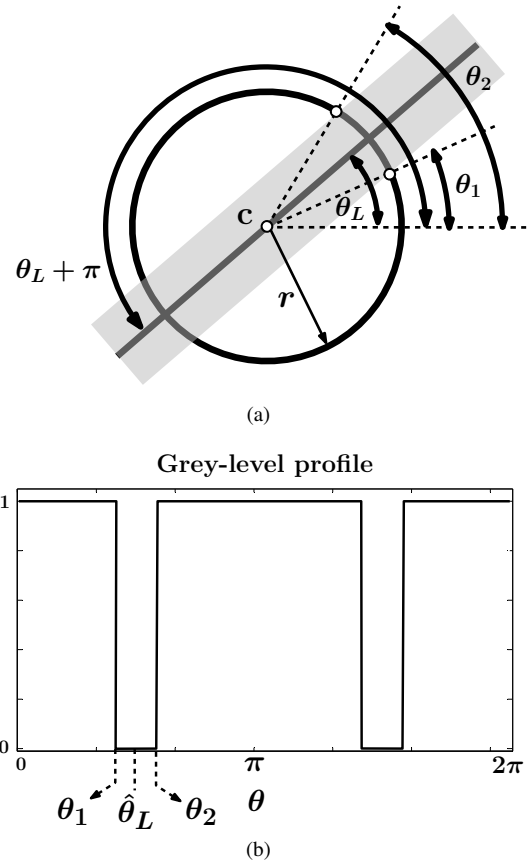


Figure 3: (a) A circle with radius r is placed such that its centre \mathbf{c} coincides with the line’s medial axis. The angles at which the circle intersects with the line will provide an estimate of the line orientation θ_L . (b) The grey-level profile along the circumference of such a circle will consist of two pulses centred at θ_L and $\theta_L + \pi$

from the grey-level image itself, performing the separation of foreground strokes from the rest of the sketch while these strokes are being vectorized. Therefore, we propose a vectorization algorithm based upon a signature obtained along the circumference of a circle centred on points of interest, as described next.

3 The circle signature

Consider a discrete circle, defined as $\mathbf{x} = \mathbf{c} + r\mathbf{u}_\theta$ where \mathbf{c} is the centre of the circle, r is the circle’s radius, $\mathbf{u}_\theta = [\cos \theta, \sin \theta]^T$, $\theta = [0, \delta\theta, 2\delta\theta, \dots, 2\pi]$ and $\delta\theta$ is the angular resolution with which sample points on the circle’s circumference are obtained. Such a circle can be used to sample the environment of a given centre pixel at a locus of points equidistant from the centre pixel. For this reason, we define the *circle signature* $S_{r,\mathbf{c}}(\mathbf{u}_\theta)$ as the grey-level profile of the sample points on the circle’s circumference, that is, $S_{r,\mathbf{c}}(\mathbf{u}_\theta) = I(\mathbf{x})$, where I is the sketch image. Since \mathbf{x} are not necessarily integer co-ordinates, we use the bilinear interpolation to obtain $I(\mathbf{x})$.

In order to obtain a well defined circle, the sample points that define the circle’s circumference should be at least 8-connected, ensuring that the signature $S_{r,\mathbf{c}}(\mathbf{u}_\theta)$ samples all the pixels that lie on the circle’s circumference. Thus the distance between two pixels on the circumference should be $\sqrt{2}$ which will be achieved if $\delta\theta = \frac{\sqrt{2}}{r}$

4 The circle signature for vectorization

Consider a line having an unknown orientation θ_L . If the sampling circle is placed such that \mathbf{c} is on the medial axis of the line, then, as shown in Figure 3(a), the signature $S_{r,c}(\mathbf{u}_\theta)$ will contain two pulses corresponding to the intersection of the circle circumference with the line as shown in Figure 3(b). The position of these pulses on $S_{r,c}(\mathbf{u}_\theta)$ will provide an estimate $\hat{\theta}_L$ for the line orientation.

From an initial sample point \mathbf{c}_t on the medial axis of the line, a subsequent sample point \mathbf{c}_{t+1} can be obtained using $\mathbf{c}_{t+1} = \mathbf{c}_t \pm d\Phi$, where \mathbf{c}_t is a sample point on the line's medial axis and the centre of the sampling circle, $\Phi = [\sin \hat{\theta}_L \cos \hat{\theta}_L]'$ and d is the distance between consecutive sample points. In this way, it is possible to span the line in both directions from a given initial sample point, thereby obtaining the line's medial axis which can in turn be parameterized by the equation of the line.

Each sketched stroke in the sketch occurs in the context of other strokes and by sampling the immediate neighbourhood of the selected point on the medial axis, it is possible to capture the local context of the line in the signature $S_{r,c}(\mathbf{u}_\theta)$. In particular, as the sample points approach junction points, the sampling circle intersects with other lines present at the junction which will be present as additional line pulses in $S_{r,c}(\mathbf{u}_\theta)$. By comparing the signature obtained from each sample point with an idealised line signature, it is therefore possible to determine when the sampling circle approaches the end of the line or a junction point and hence identify the end-points of the line being traced.

Thus, the vectorization of the sketch image can be achieved by first locating a suitable starting sample point, from which an initial circle signature, and hence an initial signature $S_{r,c_0}(\mathbf{u}_\theta)$ is obtained. An estimate $\hat{\theta}_L$ of the line orientation can then be obtained from the signature $S_{r,c_0}(\mathbf{u}_\theta)$ and this is used to locate a subsequent sample point on the medial axis and hence a new signature $S_{r,c_1}(\mathbf{u}_\theta)$. The signature is compared to the idealised line signature \bar{S} , hence verifying whether the sample point is a valid point on the line. This is repeated until no other sample point can be located on the line at which point, the line tracing is terminated, as shown in Figure 4. This is repeated until all lines in the sketch are located and vectorized. Details of each step are described hereunder.

4.1 Obtaining the line orientation estimate

The line orientation $\hat{\theta}_L$ can be defined as $\frac{1}{2}(\theta_1 + \theta_2)$, where θ_1 and θ_2 are the angles defining the edges of the line pulse in $S_{r,c}(\mathbf{u}_\theta)$, providing that the sampling circle is centred on the medial axis of the line. In ideal, isolated line strokes, the values of θ_1 and θ_2 can be determined from the derivative of the signature, where θ_1 corresponds to the negative gradient peaks and θ_2 corresponds to the positive gradient peaks.

However, the sketched lines can be embedded in shading or shadow. When this happens, the circle signature would have additional pulses that correspond to the shading strokes and, as a result, the signature derivative will contain peaks at other locations besides those at θ_1 and θ_2 . It is therefore necessary to discriminate between gradient magnitudes that define the edges of line strokes and others that define shadow strokes in order to determine the line orientation.

As shown in Figure 5, the local peaks in the derivative of the signature that correspond to the line stroke boundaries will typically have a larger magnitude than other peaks. However, selecting a single threshold on the gradient magnitude or the signature itself can result in misclassification of the shadow and line strokes, particularly in instances where the shading is variable or considerably

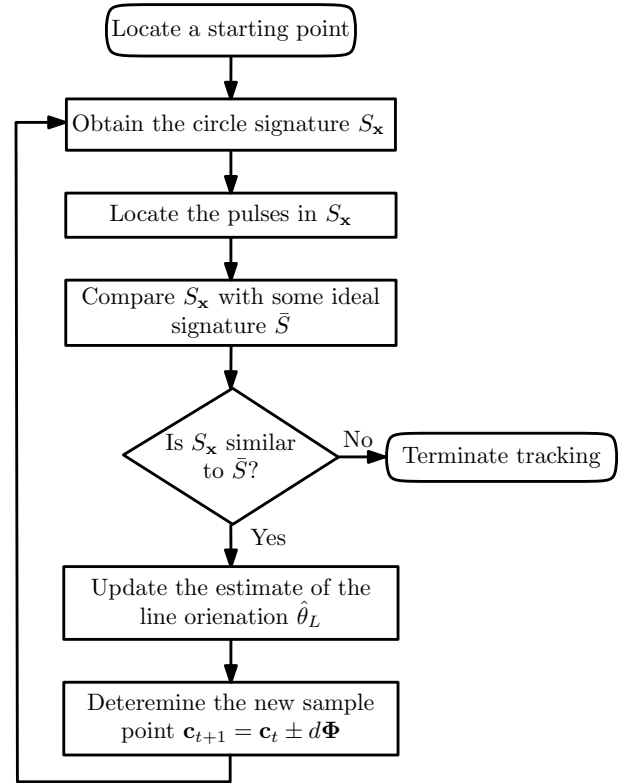


Figure 4: The vectorization of a single line. Each step is described in further detail in the text

dark. For this reason, we use a hysteresis-like thresholding to obtain a clean signature $\hat{S}_{r,c}(\mathbf{u}_\theta)$ in which pulses corresponding to line strokes are classified as such and assigned a label ω_l while all other signature values are classified as background and assigned a label ω_b .

If two sample points on the line are aligned with the line's medial axis, then any two consecutive signatures $S_{r,c_t}(\mathbf{u}_\theta)$ and $S_{r,c_{t+1}}(\mathbf{u}_\theta)$ are expected to have the line pulses at the same location on the circle circumference, such that the clean signature obtained from $S_{r,c_t}(\mathbf{u}_\theta)$ can guide the search for line pulses in $S_{r,c_{t+1}}(\mathbf{u}_\theta)$. As the sampling circle is propagated along the line, the sample circle can intersect with new lines, particularly when the sampling circle approaches a junction. Similarly, the circle may cease to intersect with a line if this has reached an open-ended termination point. In such cases, it should be possible to detect the insertion or removal of line pulses in $S_{r,c_{t+1}}(\mathbf{u}_\theta)$ as these effect the line vectorization. Thus, we introduce two thresholds T_l and T_h such that any signature component less than T_l is labelled as a line pulse while any signature component greater than T_h is labelled as a shadow or background. Ambiguous parts of the signature that are greater than T_l but less than T_h are labelled according to the previous signature, such that:

$$\hat{S}_{r,c_{t+1}} = \begin{cases} \omega_l, & \text{if } S_{r,c_{t+1}}(\theta) \leq T_l \\ \omega_b, & \text{if } S_{r,c_{t+1}}(\theta) \geq T_h \\ \omega_l, & \text{if } T_l \leq S_{r,c_{t+1}}(\theta) \leq T_h, \hat{S}_{r,c_t}(\theta) = 0 \\ \omega_b, & \text{if } T_l \leq S_{r,c_{t+1}}(\theta) \leq T_h, \hat{S}_{r,c_t}(\theta) = 1 \end{cases} \quad (1)$$

where ω_l is the label assigned to a line pulse and ω_b is the label

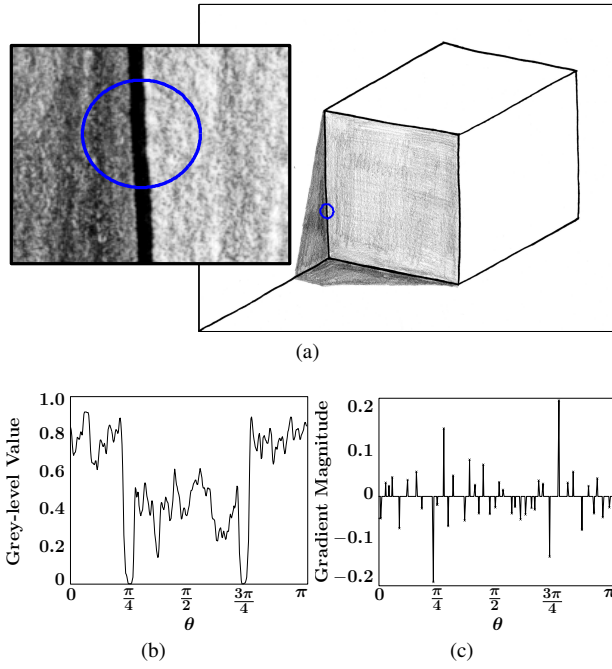


Figure 5: (a) A sketch image with a sample circle centered on one of the sketched lines. (b) The signature S_x obtained from this circle. (c) The derivative of the signature in (b)

assigned to a shadow or background pulse in the signature, $\theta_n \leq \theta \leq \theta_p$ and $[\theta_n, \theta_p]$ are pairs of consecutive negative and positive gradient magnitude peaks in the signature derivative.

In this way, line pulses are introduced to the signature if there is sufficient evidence that the pulse in the signature corresponds to a line stroke. Similarly, lines are removed from the signature if there is sufficient evidence that the pulse present in previous signatures is no longer present in the current signature. Such evidence is obtained through the grey-level value at each point on the sampling circle and hence the values of T_l and T_h provide the limit for which pulses in $S_{r,c}(\mathbf{u}_\theta)$ can be considered as switching from ω_b to ω_l or vice versa.

We suggest that the value of T_h and T_l can be found through a clustering algorithm. In general, the sketch with shadows has clusters of grey-levels. The darkest grey-level cluster, which we denote by \bar{g}_l corresponds to the line strokes, the brightest grey-level cluster, which we denote by \bar{g}_b corresponds to the sketch background, while a range of intermediary grey-level clusters, which we denote by \bar{g}_{s_n} , where $n = 1 \dots N$ and N is the number of distinguishable shades, defines the sketched shadows. In sketches that have uniform, consistent shading throughout the sketch, $N = 1$ and there is only one cluster that characterises all shadow pixels. However, with variable shading, it is possible to identify unique shades such that $N > 1$ in such sketches. In our classification of the clean signature $\hat{S}_{r,c}$, the class defined by ω_l corresponds to the cluster with grey-level \bar{g}_l whereas the class defined by ω_b which defines the sketch background, incorporates the clusters defined by \bar{g}_{s_n} and \bar{g}_b , that is, all parts of the image that are not line strokes. Typically, $\bar{g}_l < \bar{g}_{s_1}, \dots, \bar{g}_{s_N} < \bar{g}_b$, in order to allow the observer to distinguish between the line strokes, the shading strokes and the image background, however, the grey-level distributions of these categories may overlap such that clustering alone can possibly misclassify pixels in each category.

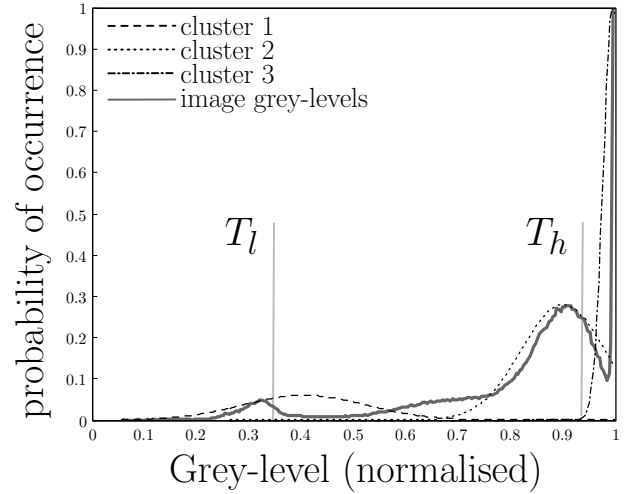


Figure 6: Using k -means clustering to identify the three grey-level clusters in the image and hence the thresholds T_h and T_l

If we were to model the grey-level distribution in each cluster by a Gaussian distribution, then we can choose the thresholds $T_l = \mu_l - \sigma_l$ and $T_h = \mu_s + \sigma_s$, where μ_l, σ_l are the mean and standard deviation of the grey-level values of pixels classified as cluster \bar{g}_l and μ_s, σ_s are the mean and standard deviation of the grey-level values of pixels classified as cluster \bar{g}_{s_1} . Figure 6 illustrates the selection of these thresholds when $N = 1$. Thus, T_l and T_h represent the non-overlapping grey-levels between the \bar{g}_l and \bar{g}_s clusters. In this implementation, we use the K-means algorithm [Davies 2004] to perform the clustering, although other clustering algorithms can be used [Hamerly and Elkan 2002].

4.2 Measuring the validity of the signature

While tracing the line stroke, it is important to verify the validity of the signature. This is necessary so that we determine whether the sampling circle is centred on the medial axis of the line stroke and hence establish whether the orientation obtained from the signature is a valid estimate of the line orientation. Such verification is therefore necessary to ensure accurate line tracing. In order to verify the validity of the signature, we require an idealised signature \bar{S}_r against which the cleaned signature \hat{S}_r can be compared. Such a signature can be obtained from a binary, ground-truth line, whose orientation and medial axis are known. By placing the sampling circle on the medial axis of such a ground-truth line, we obtain a signature which is naturally segmented into the two classes ω_l and ω_b , making it suitable as an ideal signature against which actual line signatures, obtained from the sketched strokes can be compared.

Comparison of the idealised signature and the cleaned signature requires matching the location of the pulses of the two signatures, confirming that the signature is obtained from a sampling circle centred on the line medial axis. Such a comparison can be obtained by determining the sum of squared distances between the two signatures. The line stroke however, is not necessarily of the same orientation as that of the ground-truth line such that it is necessary to first align the signatures such that the sum of squared distances between the signatures is minimized. Such an alignment of the signatures can be achieved through correlation. However, the cleaned and idealised signatures can also have discrepancies in the pulse widths which effect the correlation between the two signatures. These discrepancies in the pulse widths are due to two main causes. In the first instance, line strokes may have different line widths to that of

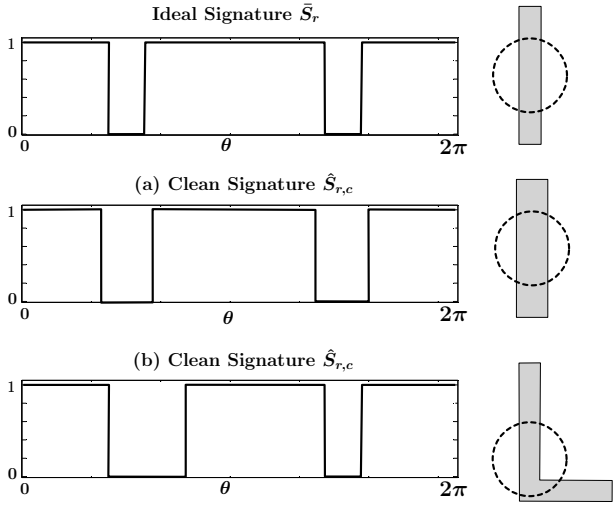


Figure 7: An idealised signature consists of two pulses separated by π . (a) If the line stroke from which the cleaned signature is obtained has a larger line width, the signature pulses will also be uniformly wider. (b) As the sampling circle approaches a junction, the circle will intersect with the line stroke tangentially, causing a non-uniform increase in the pulse widths.

the ground-truth line. In such a case, the two cleaned signature pulses would have a uniform increase (or decrease) in the pulse width when compared to the ideal signature as shown in Figure 7. Such a pulse width difference does not detract from the accuracy of the orientation estimate or the location of the medial point, and therefore, should not be penalised in the comparison of the cleaned signature to the ideal signature. On the other hand, when a sampling circle approaches a junction, the sampling circle starts to intersect tangentially with the other lines at the junction. As a result, the signature would experience a non-uniform increase in the pulse width as shown in Figure 7. Such a non-uniform increase in the pulse width would cause a deviation in the line orientation estimates and should therefore be penalised. Thus, changes in pulse widths between the ideal line signature and the cleaned signature should be treated differently. Correlation however does not provide the mechanism for this distinction and any change from the line width of the ideal signature would be penalised as a sign of non-conformity between the two signatures. For this reason, we propose an alternative method for comparing the two signatures which first aligns the signature $\hat{S}_{r,c}(\mathbf{u}_\theta)$ with the ideal signature $\bar{S}_r(\mathbf{u}_\theta)$ after which the difference between the two signatures is quantified.

To align the two signatures, we first shift $\hat{S}_{r,c}(\mathbf{u}_\theta)$ until there is the least discrepancy between the position of the pulses in $\hat{S}_{r,c}(\mathbf{u}_\theta)$ and in $\bar{S}_r(\mathbf{u}_\theta)$. In this way, we would have effectively aligned the stroke orientation with that of the ideal line. Once the orientation is aligned, we alter the stroke width of $\hat{S}_{r,c}(\mathbf{u}_\theta)$ until there is the least discrepancy between the pulse widths in $\hat{S}_{r,c}(\mathbf{u}_\theta)$ and in $\bar{S}_r(\mathbf{u}_\theta)$. One way of doing this is by changing the number of samples that define the pulse, effectively performing angular-scaling on the pulses in the signature [Kamen and Heck 2007]. If this angular-scaling is applied uniformly to the two pulses in the signature, the discrepancy between the ideal signature and the cleaned signature would be eliminated if the difference in pulse width is a result of a difference in the line width. If the difference in pulse width between the two signatures is caused by a tangential intersection of the sampling circle with the line strokes, then, although $\hat{S}_{r,c}(\mathbf{u}_\theta)$ is adjusted to fit as close as possible to $\bar{S}_r(\mathbf{u}_\theta)$, a discrepancy in the

pulse widths will still remain.

To quantify the differences between the resulting signature and the idealised signature, we then use a comparison measure that is made up of two components. The first measures the separation between the pulses, which is expected to be π for a sampling circle centred on the line's medial axis such that

$$D_\theta = 1 - \frac{|\pi - (\hat{\theta}_{L_2} - \hat{\theta}_{L_1})|}{\pi} \quad (2)$$

where $\hat{\theta}_{L_1}$ and $\hat{\theta}_{L_2}$ are the line orientation estimates obtained from the two signature pulses that best match with the ground-truth pulses. In this way, $D_\theta = 1$ when the line pulses are separated by π and decreases as the separation differs from this expected value.

The second component measures the difference between the ground truth signature and adjusted sample signature, hence measuring the differences incurred by changes in line width as well as changes in the number of pulses present in the signature, hence determining if new lines are being detected or if the sampling circle has approached an open-ended line end-point. Thus, we define:

$$D_w = 1 - \frac{|\hat{S}_{r,c}(\mathbf{u}_\theta) \cap (\bar{S}_r(\mathbf{u}_\theta) = \omega_l)|}{2|\hat{S}_{r,c}(\mathbf{u}_\theta) = \omega_b|} + \frac{|\hat{S}_{r,c}(\mathbf{u}_\theta) = \omega_l \cap (\bar{S}_r(\mathbf{u}_\theta) = \omega_b)|}{2|(\bar{S}_r(\mathbf{u}_\theta) = \omega_b)|} \quad (3)$$

which has a value of $D_w = 1$ when the two signature are perfect matches and decreases when the pulse widths are mismatch and when the number of pulses in $\hat{S}_{r,c}(\mathbf{u}_\theta)$ differs from the expected two pulses.

Hence, we define the comparison measure as $D = \frac{1}{2}D_\theta + \frac{1}{2}D_w$

4.3 Locating a suitable starting point

The line tracing requires the location of an initial starting point on the line. Although there are various techniques which can be used to locate such a point, such as those described in [Liu and Dori 1999] and [Song et al. 2002] among others, we choose to use the circle signatures to determine suitable initial starting points as this allows us to locate these points more efficiently.

We do this by selecting an initial point on the image. This could be the geometric centre of the image, or determine through image moments [Davies 2004]. Once located, a sampling circle is centred on this point and a signature is obtained. The pulses on this signature that have a grey-level value which is less than T_l are likely to correspond to line strokes. However, since the sampling circle is not centred on the stroke itself, the sampling circle does not necessarily pass through the medial axis of the stroke segment. Thus, given this initial location of a line segment, we create a sequence of sample points that lie on a line perpendicular to the circle circumference at the point of intersection with the line segment as shown in Figure 8. The signature at these points is obtained and compared to the ground truth signature such that the sample point that provides the best matching signature is used as a starting point on the line.

This sampling circle can therefore locate multiple starting points on different lines which are traced individually, providing that the line would not have already been traced. Once all starting points have been exhausted, the radius of this sampling circle can be increased in order to search for new, untraced lines. This can be repeated until the circle radius is equal to half the image size, hence exploring all the lines in the image.

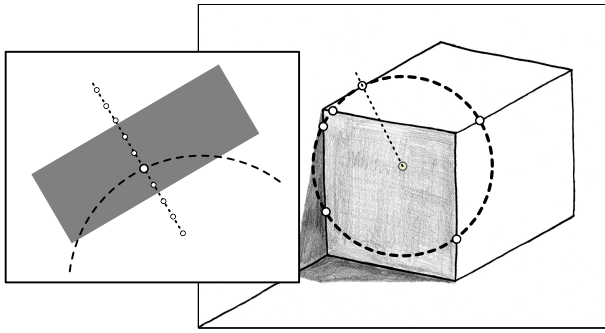


Figure 8: Initial sample points on the sketched lines are located by using sampling circles centered on the image centre. Sample points perpendicular to the point of intersection of the circle circumference with the sketched lines are identified at each candidate starting point and the signature at each of these sample points is compared to the ground truth to determine the most suitable starting point on each line.

4.4 Parameterising the line strokes

Once a line has been completely traced, it is represented by a set of points that should coincide with the medial axis of a straight line. These points can therefore be parameterised by the equation of the straight line such that each line stroke is represented by a linear equation and the line endpoints. Since the line stroke is obtained from a freehand sketch, the line stroke itself is not necessarily a perfect straight line, such that this line parameterisation must be performed by finding the best fitting line, using mathematical techniques such as linear regression.

From the line parameterisation, it is also possible to determine the location of the junction points or the intersection between adjacent lines. Once again, the sketched nature of the line strokes implies that it is unlikely that three or more line vectors will intersect at a common point, such that the junction point can be located by solving the intersection point in the least-squares sense [Salahi 2010].

5 Evaluation methodology

In order to evaluate the performance of the circle-based vectorization algorithm, we test the algorithm on sketches such as those shown in Figure 9. These sketches have been drawn on A4, plain paper, using a soft 4B pencil for the shading and line strokes. The sketches were then scanned with a flat-bed scanner, using the default resolution of 300dpi. These sketches provide examples of sketches of varying complexity and have mixtures of uniform and variable shading as would be expected in artistic sketches [Oliver 1979]. These sketches also contain different instances of *W*, *Y*, *T* and *L* junctions and therefore provide good examples of sketches of trihedral objects.

To quantify the performance of the vectorization algorithm, the performance evaluation protocol described in [Liu and Dori 1997] can be used. This protocol defines the performance of vectorization algorithms by using two indexes, namely the *pixel recovery index (PRI)* and the *vector recovery index (VRI)*.

The role of the *PRI* is that of determining the ability of the vectorization algorithm to locate the line strokes in the sketch. This index operates at a pixel level and compares the line stroke pixels detected by the vectorization algorithm with the known line stroke pixels of a ground-truth image, identifying the true detection rate, that is the fraction of ground truth pixels detected by the vectoriza-

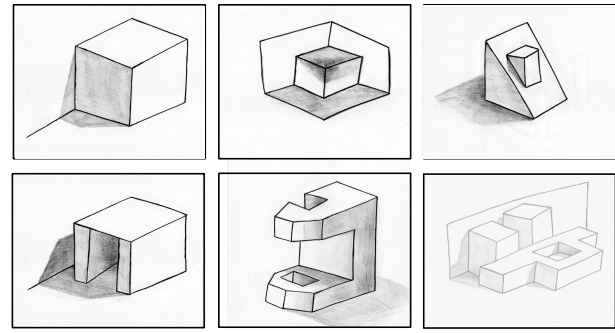


Figure 9: The sketches used to evaluate the circle-based vectorization algorithm

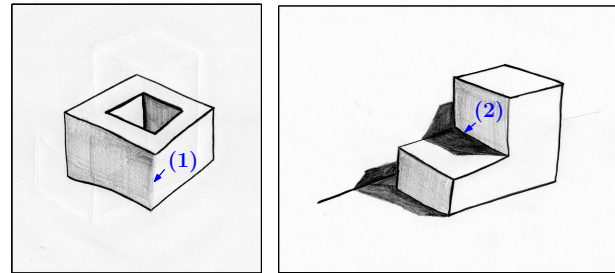


Figure 10: Sketches with instances of poor contrast between edge strokes and shadows

tion algorithm, and the false detection rate, that is, the fraction of detected pixels that do not have matching ground-truth pixels [Liu and Dori 1997]. On the other hand, the *VRI* compares the vectorization algorithm at a vector level. In this case, the quality of the detected vectors is measured by the proximity of the end-points as well as the distance between the detected vectors and the ground-truth vectors. The *VRI* also takes into account the fragmentation rate, hence determining whether the ground-truth vector is detected in its entirety [Liu and Dori 1997].

To use this evaluation protocol, it is necessary to obtain ground-truth vectors against which the detected vectors can be compared. Such ground-truth vectors cannot be obtained directly from the sketch such that it is necessary to create artificial sketches with known ground-truth vectors. Ideally, these artificial sketches are as realistic as possible, such that the performance measures obtained from these sketches reflect the expected performance of the algorithm on real sketches. To do so, we create the ground-truth vectors by manually selecting the junction points in the sketches shown in Figure 9. From these junction points we create vectors which serve as the medial axis of the sketch. These vectors are dilated, thereby obtaining line strokes. Shading strokes are introduced to these images by copying the texture of shading strokes of the sketches in 9 onto the desired locations in the new images.

In addition, the circle based vectorization algorithm was performed on drawings such as those shown in Figure 10. These drawings have instances, such as those marked (1) and (2), where there is no contrast between the shading and the line strokes. In such cases, it is difficult to obtain signatures in which two distinct poses, corresponding to the line strokes, are obtained. These drawings therefore serve to illustrate the way the vectorization algorithm handles sketches from which a good signature cannot always be obtained.

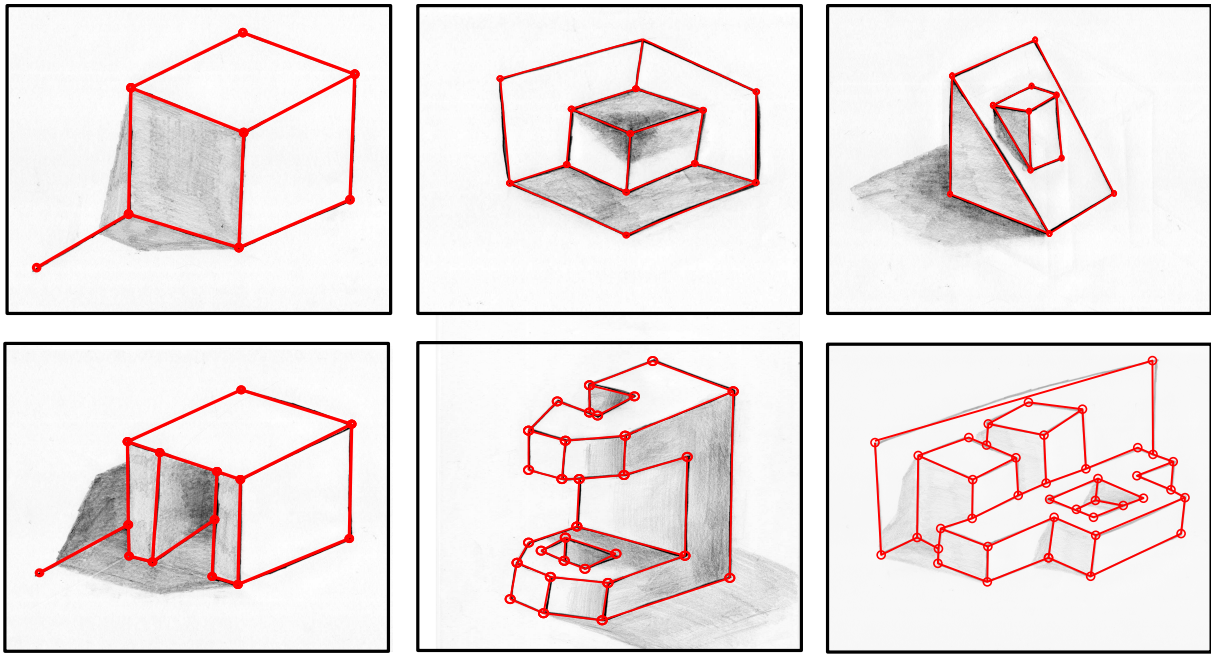


Figure 11: The vectors obtained for the sketches in Figure 9

6 Results and discussion

The results obtained by the circle-based vectorization for the sketches shown in Figure 9 are shown in Figure 11. In this implementation, we set the radius of the sampling circle to $r = 30$ pixels. This was determined empirically as being suitable for the resolution of the sketches at hand, allowing sampling circles to fit within the shortest line segments while at the same time providing sufficient context for the line signature to be meaningful. The tracing step size was set to $d = 1$ pixel, such that each point on the medial axis of the line is sampled.

The results obtained show that the vectors obtained from the circle-based vectorization algorithm are well placed with respect to the sketch image strokes. These results show that the circle-based vectorization algorithm can successfully vectorize sketches containing shadows of uniform as well as varying grey-levels. These results are corroborated by the values of PRI and VRI presented in Table 1. Ideally, the PRI and VRI values are both 1, indicating a perfect match between the detected line vectors and the ground truth vectors. In practice however, discrepancies in the estimation of the stroke width as well as pixel variations in the location of the detected vectors will reduce the values of the PRI and the VRI to less than unity. Table 1 shows that the PRI and VRI values obtained by the circle-based vectorization algorithm are close to unity. These values are in fact comparable to the values obtained by other vectorization algorithms as reported in the literature [Liu and Dori 1997]; [Liu and Dori 1999], indicating that circle-based vectorization retains the same level of accuracy while obtaining the stroke vectors from sketches with shadows.

Figure 12 shows the results obtained for the sketches with instances of poor contrast between shadows and edge strokes given in Figure 10. In both sketch examples, the edge marking the discontinuity between the two surfaces is made more evident by shading contrast rather than by the line stroke representing the edge. This can be observed in particular on the edges marked (1) and (2) in Figure 12, where, in the case of edge (1) the line stroke is not present and in the case of edge (2), the dark shading has the same grey-level as the line

Table 1: The PRI and VRI values obtained for the artificial sketches

	Sketch						Mean Value
	1	2	3	4	5	6	
PRI	0.99	0.95	0.98	0.97	0.98	0.91	0.95
VRI	0.90	0.82	0.92	0.89	0.93	0.80	0.88

strokes. In both examples, the grey-level signature obtained from circles centred on these edges do not contain the expected two line pulses as shown in Figure 13. In the case of edge (1), the grey-level of the shading falls within the range $T_l < S_{r,c}(\theta) < T_h$ such that, using Equation 3, this should be cleaned according to the previous signature. However, since no medial point on this edge matches the idealised line signature, all the signature $S_{r,c}$ is labelled as background and, in consequence, no edge medial point is determined. On the other-hand, all of the shading strokes found on the signatures obtained from edge (2) in Figure 12, have a grey level smaller than T_l , such that, all shading strokes on the circumference of the sampling circle are labelled as foreground. As a result, the resulting cleaned signature would differ from the idealised line signature such that no edge medial points are detected from this signature. In consequence, as shown in Figure 12, line strokes sketched under these conditions are either not traced by the vectorization algorithm, or, in cases of variable shading, the tracking terminates prematurely, as can be observed in the edges labelled (3) and (4) in Figure 12. In the context of vectorization as the necessary pre-processing for edge labelling applications, such missing vectors can potentially be identified through post-processing, by comparing the vectors at junctions with the permissible trihedral junctions.

7 Conclusion

In this paper, we show that the circle-based vectorization algorithm can be used to vectorize sketches that have shading and shadow marks. The performance of the algorithm is shown to be compa-

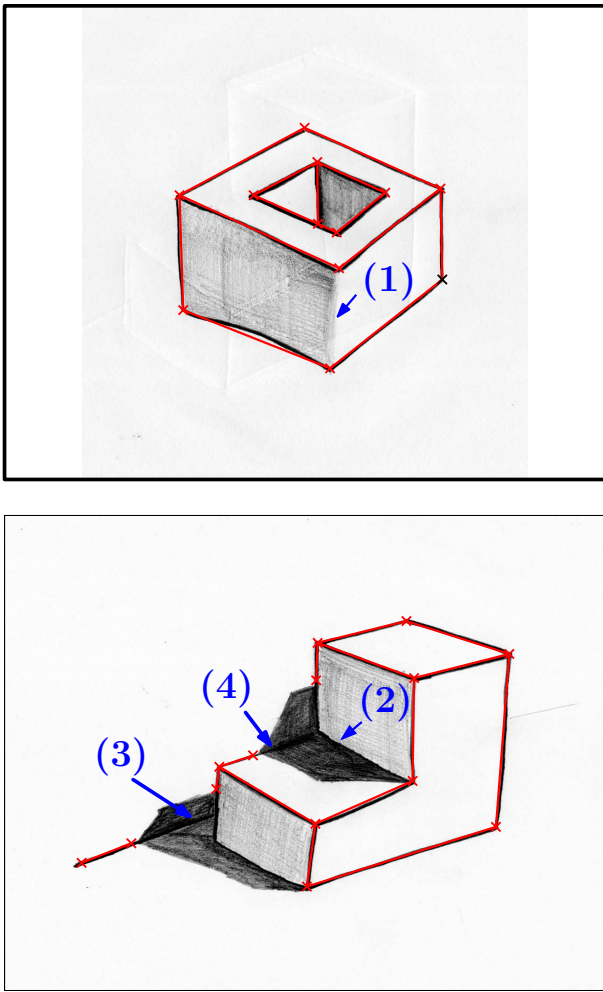


Figure 12: Vectors obtained for sketches in Figure 10. The edges labelled (1) and (2) are not vectorised since as shown in Figure 13, the signature does not contain the expected line pulses. Edges (3) and (4) show examples of instances when the tracing terminates prematurely, in these examples, this happens because the dark shading masks part of the line strokes.

able to other vectorization algorithms described in the literature, making this algorithm suitable for sketch interpretation applications which make use of shadows and shading.

The algorithm is parameterised by the circle radius r and the tracing step d , both of which are currently determined in an empirical manner. The circle radius is directly linked to the resolution of the sketch image and has an effect on the length of the smallest segment that can be detected by the algorithm as well as the proximity of line strokes. Similarly, the tracing step size has an effect on the number of sample points obtained from a line stroke and hence, the level of detail that is captured along the line stroke. Choosing a tracing step size that is too large, can lead to the possibility of missing junctions on the line, while small tracing step sizes result in over-sampling of the line stroke, which, although has no adverse effect on the quality of the line vectors obtained, increases the computational costs required to obtain these vectors.

Ideally, the circle radius and the step size are adapted automatically according to the resolution of the sketch. Since the sketch resolution can vary on a local basis, as can be observed in Figure 9, with

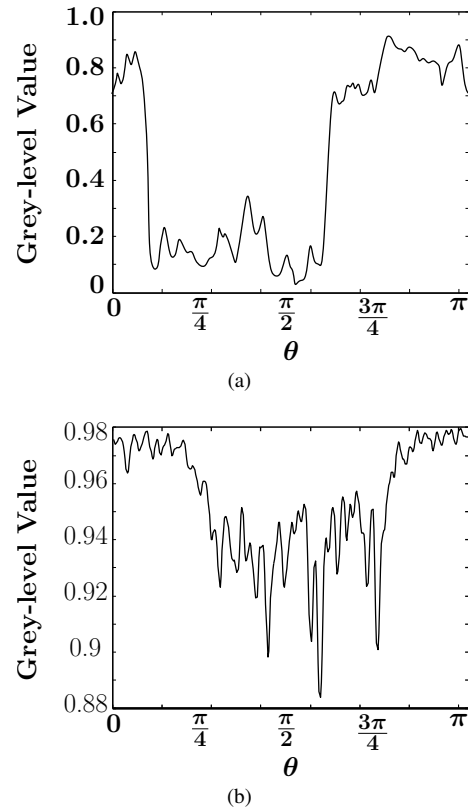


Figure 13: Grey-level signature obtained from points on the edges marked (1) and (2) in Figure 12. In (a), corresponding to the edge marked (1), the signature grey-levels fall within the thresholds T_l and T_h while in (b), which corresponds to the signature of a point on the edge marked (2), the grey-level of the shading strokes is less than T_l . In both cases, the signature does not have the two expected pulses.

an individual sketch having a mixture of sparse lines and regions with relatively high detail, we believe that this requires a multi-resolution approach to line vectorization. Such an issue is common to all vectorization algorithms but we believe that the extending this work to include circle signatures obtained from sampling circles of different radii, the vectorization algorithm would be able to handle this multi-resolution problem efficiently.

Acknowledgements

This work is funded by the University of Malta, under the research grant SCERP02-03.

References

- BONNICI, A., AND CAMILLERI, K. 2009. Scribble vectorization using concentric sampling circles. In *Advanced Engineering Computing and Applications in Sciences, 2009. ADVCOMP '09. Third International Conference on*, 89–94.
- BONNICI, A., AND CAMILLERI, K. 2012. Exploiting artistic cues to obtain line labels for free-hand sketches. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, K. Singh and L. B. Kara, Eds., 77–86.

- DAVIES, E. R. 2004. *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- HAMERLY, G., AND ELKAN, C. 2002. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management*, ACM, New York, NY, USA, CIKM '02, 600–607.
- HILAIRE, X., AND TOMBRE, K. 2006. Robust and accurate vectorization of line drawings. *IEEE Transactions on Pattern Analysis and Machine Interpretation* 28, 6, 890–904.
- KAMEN, E., AND HECK, B. 2007. *Fundamentals of Signals And Systems Using the Web And Matlab*. Pearson Prentice Hall.
- KATZ, R. A., AND PIZER, S. M. 2004. Untangling the blum medial axis transform. *International Journal of Computer Vision* 55, 2-3 (November), 139–153.
- KEYSERS, D., AND BREUEL, T. 2006. Optimal line and arc detection on run-length representations. In *Proceedings Graphics Recognition Workshop*, Springer, LNCS.
- LIU, W., AND DORI, D. 1997. A protocol for performance evaluation of line detection algorithms. *Machine Vision Applications* 9, 240 – 250.
- LIU, W., AND DORI, D. 1999. Sparse pixel vectorisation: An algorithm and its performance evaluation. *IEEE Transactions of Pattern Analysis and Machine Intelligence* 21, 3, 202–215.
- OLIVER, R. 1979. *The Sketch*. Van Nostrand Reinhold.
- RAMEL, J. Y., VINCENT, N., AND EMPTOZ, H. 1998. A coarse vectorization as an initial representation for the understanding of line drawing images. *Graphics Recognition - Algorithms and Applications, GREC'97, Lecture Notes in Computer Science, Springer-Verlag 1389*, 48 – 57.
- RÖÖSLI, M., AND MONAGAN, G. 1996. Adding geometric constraints to the vectorisation of line drawings. In *Graphics Recognition - Methods and Applications, GREC'95, Lecture Notes in Computer Science, Springer-Verlag, vol. 1072*, 49–56.
- SALAH, M. 2010. Robust least squares solution of linear inequalities. *Applied Mathematics Letters* 23, 5, 605 – 608.
- SONG, J., AND LYU, M. R. 2005. A Hough transform based line recognition method using both parameter space and image space. *The Journal of the Pattern Recognition Society* 28, 539 – 552.
- SONG, J., SU, F., CHEN, J., TAI, C., AND CAI, S. 2002. An object-oriented progressive-simplification-based vectorization system for engineering drawings: Model, algorithm, and performance. *IEEE Transactions of Pattern Analysis and Machine Intelligence* 24, 8.
- TOMBRE, K., AH-SOON, C., DOSCH, P., MASINI, G., AND TABBONE, S. 2000. Stable and robust vectorization: How to make the right choices. *Graphics Recognition - Recent Advances, GREC'99, Lecture Notes in Computer Science, Springer-Verlag 1941*, 3–18.