EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2006)
Thomas Stahovich and Mario Costa Sousa (Editors)

A New Sketch Based Interface using the Gray-level Co-occurrence Matrix for Perceptual Simplification of Paper Based Scribbles

A. Bartolo[1], K. P. Camilleri[1], P. J. Farrugia[2], J. C. Borg[2]

[1]Department of Electronic Systems Engineering, University of Malta, Malta
[2]Department of Manufacturing Engineering, University of Malta, Malta

Abstract

The sketching activity has an important role in conceptual design and a variety of tools exist which help designers to facilitate the generation of 3D models form sketched drawings. This paper describes a new sketch-to-3D tool, which uses annotations to aid the interpretation of the drawing. Over-traced lines present in the designer's scribbles provide an interpretation challenge, which must be resolved in order to obtain 3D models from these sketches. Perceptual grouping techniques used to interpret such images require that the drawing is represented as vectors. These are generally obtained through thinning or edge detection. However, we show that processing scribbles using these techniques result in a large number of vectors which do not provide a faithful representation of the drawing. This paper investigates the use of the co-occurrence matrix to perceptually simplify these drawings, thus obtaining a smaller number of vectors which describe the drawing more faithfully.

Categories and Subject Descriptors (according to ACM CCS): I4.6 [Image Processing and Computer Vision]: Edge and feature detection I.5.4 [Pattern Recognition]: Computer Vision

1. Introduction

A designer is sitting at a cafeteria table, pondering upon the design of a perfume bottle that complements the marketing program of a new perfume. Suddenly, the designer manages to come up with an idea, which is quickly externalised on a readily available piece of paper. A 'dialogue' between the sketch and the designer is created leading to a candidate form solution of the perfume bottle. The designer would now like to obtain a three dimensional (3D) virtual model of this sketch, which will aid the designer in visualising better the form intent. This will help the designer verify whether the bottle will look as appealing in 3D. Situations such as this are not uncommon and studies show that very often, ideas come to mind when we least expect them [SH99]. Studies also show that despite the increasing number of portable computer systems, people still prefer to use paper and pen as a sketching medium, especially in the conceptual design stages [FBCS05].

What makes sketches so important in design? Before addressing this question, it is necessary to distinguish between different drawing categories. This may be done by classifying the drawings into a hierarchy according to their accuracy and detail. Accurate technical drawings are found at the highest level, giving a detailed description of an object, including its dimensions. The freehand sketch is less accurate, yet it is drawn after certain shape features have been clearly established. Thus, it is placed at a lower level than the technical drawing. At the lowest level one finds scribbles, which are drawn when the shape of the object is still being explored, since scribbles are the type of paper drawings often used during the conceptual design stage.

In the scenario above, the designer captured an idea on paper. In this sense, the paper based scribble acts as a storage device. This ensures that when the designer returns to the office, the idea can be remembered. Once the mental model has been externalised on a visualising medium, the designer may need to adjust some aspects of the preliminary form. This is one of the roles of scribbles in the designer's visual thinking process. Moreover, scribbles can be a source of knowledge and insight for an alternative or even completely different idea [Mul01]. The scribble can also be considered as a thought process, where the evolution of the designer's ideas are captured on paper. Finally, to obtain a better visualisation of the design idea, the designer would like to generate a 3D model of the scribble. To do this, the designer would have to either manually transfer the sketch into a CAD system, or use a sketch-based interface which acts as an intermediary step between the drawing and the CAD system.

This paper describes a method which can be used to perceptually simplify the scribble strokes for subsequent processing as well as an intuitive method for annotating the scribble to enable the correct interpretation of the intended form. The main problems involved in the interpretation of scribbles and the perceptual grouping techniques which may be applied to the interpretation of scribbled drawings are described in Section 2. A different image preparation method is suggested in Section 3 whilst a novel interface which facilitates the interpretation of these scribbles is given in Section 5. Section 6 compares the results obtained by the proposed perceptual simplification methods and the standard methods image preparation methods.

2. Scribbles: The interpretation challenges

Scribbles are challenging for a number of reasons. A scribble reflects the drawing habits of an individual and one cannot determine a single correct method of representing an object. A typical example of this is demonstrated by the different possibilities with which one can represent a 3D object on a 2D plane. The interpretation of the scribble must be in accordance to the projection with which it is drawn and humans can do this intuitively, because they know what views are normally used. Paper based scribbles provide an additional challenge as they are by their nature made up of several overlapping line strokes rather than single crisp strokes, thus 'fuzzifying' the shape drawn. This fuzziness arises from the fact that in the conceptual design stage, the designer does not yet have a definite object shape. Human interpretation of multiple line strokes follows the Gestalt laws [Sau03], which form the basis of perceptual grouping techniques. Since scribbling is part of a thought and object exploration process, the designer might find it necessary to adjust some of the features in the scribble, without necessarily rubbing off the old features. It is therefore necessary to select as more salient those line strokes that form the object's features over others which were discarded by the designer when scribbling.

Sha'ashua and Ullman [SU88] do this by describing the image as a network of orientation vectors. A vector ρ_i may have one of two states; it is active if it has an underlying line segment and virtual otherwise. The state and orientation of each vector ρ_i contributes to the saliency of the curve of which it is part. This algorithm will thus classify each line stroke in the image according to its saliency, favoring long, smooth, continuous curves. Gui and Medioni [GM93] argue that the physical evidence extracted locally from images is ambiguous and does not correspond to the expected perception of the image. For this reason, they impose global perceptual constraints by introducing the concept of an *extension field* which is a maximum likelihood directional vector field describing the contribution of a single unit-length edge element to its neighbours. Each pixel site in the image accumulates votes from the vector field of its neighbours, such that the most salient sites will obtain the highest vote count. A global saliency measure is also used by Guichard and Tarel [GT99], who define saliency as the gain in 'energy' obtained after introducing a new edgel to an existing set of edgels describing a curve. Guichard and Tarel assume that the salient curves in the image may be modelled as parametric curves of the form $x = \mathrm{f}(y)$ where x and y are the coordinates of points on the curve. The Kalman filter framework is used to recursively update the curve parameters and hence determine the edgel groups forming a salient curve. Whilst recognizing the need for a global saliency measure, Saund [Sau03] proposes a saliency measure based on figural closure, arguing that this is considered as a salient feature in sketches and drawings. The algorithm proposed traces through a number of segments in attempt to obtain closed contours. Segment tracing may be done according to two preferred directions, namely maximally turning or smooth continuation. The smooth continuation preference is similar to the saliency criterion proposed by Sha'ashua and Ullman, and highlights paths drawn with the same pen stroke. In contrast, maximally turning preferences seeks the most compact and tightly closed paths. Junction preference scores are used to weight decisions taken at each junction and are required to resolve ambiguities when multiple tracing options are available. Whilst the junction preference scores provide a local saliency measure, Saund defines a global figural goodness, which is based on the compactness of the figure, and the distance between the endpoints of the traced curve. The algorithms discussed so far required either an iterative computation of the saliency measure, or an iterative path tracing. Kelley and Hancock [KH00] propose a single pass grouping algorithm, based on a measure of geometric affinity between segments. This affinity is obtained using a probabilistic linking field based on the length and orientation of a virtual linking segment formed by the endpoints of two segments. Thresholding this linking field using an adaptive entropy based threshold, will result in an affinity matrix. Eigen decomposition can then be applied to this matrix to separate the image segments into clusters according to the objects in the image.

The techniques described above require some image preprocessing before the saliency measures may be obtained. Sha'ashua and Ullman [SU88], assume that the image is binarised, whilst the algorithms proposed in [GT99, Sau03, KH00] require that the image data is represented as line segments. These segments may be obtained by either performing image binarisation, line thinning and then segmentation or, as proposed in [GT99] by performing edge detection followed by segmentation. In either case, this may result in a large number of segments, and one of the problems encountered by these algorithms is the need to avoid an exhaustive search amongst all the segments. Figure 1 shows the performance of a simple iterative global thresholding technique and Sobel's edge detection on a noisy binary test image. In Figure 1(c) the binarised image shows misclassification of several foreground pixels, which will be in interpreted as short line segments. This results in a number of vector data for which there is no supporting image foreground. In Figure 1(d) the foreground noise breaks the smooth edges, hence splitting a single vector into a number of smaller vectors. Although better thresholding or edge detection techniques will reduce the effect of noise, these do not necessarily reduce the number of redundant segments. Scribbles consist of multiple line strokes, most of which are supporting line strokes which the human vision system can immediately group as one single stroke. However, pre-processing scribbles with binarisation or edge detection techniques will result in separate segments for each of these strokes. These line strokes must be grouped by a perceptual grouping algorithm. Ideally, the image pre-processing would be able to provide the perceptual grouping algorithm with a single segment for these supporting line strokes. This would allow the perceptual grouping algorithm to identify curve saliency from a smaller number of segments, thus reducing the search required by the algorithm. The following section investigates the use of the co-occurrence matrix, a technique traditionally applied to texture analysis, to group these supporting line strokes into a smaller number of appropriate line strokes.

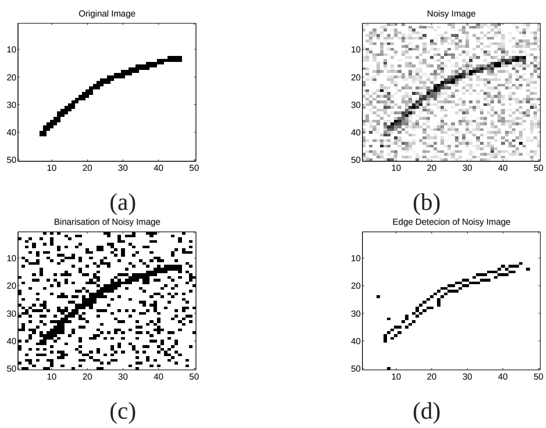


Figure 1: *Performance of simple image pre-processing techniques under noise conditions. (a)A binary test image. (b) image corrupted with additive zero mean Gaussian distributed noise, having a variance of 0.12. (c) result of global thresholding (d) result of Sobel's edge detection.*

3. The co-occurrence matrix

The co-occurrence matrix [HS01] records the number of times a pixel with gray level g_1 occurs in the vicinity of a pixel with gray level g_2. The location defining the pixel's vicinity may be determined by a vector of length d and orientation θ. Thus the co-occurrence matrix will compare the gray levels of pixel pairs having co-ordinates (r,c) and $(r+d\sin(\theta), c+d\cos(\theta))$ where c and r represent the horizontal and vertical axis of the image respectively. Thus for each gray-level pair combination in an image f, the matrix $T_{g_1,g_2,d,\theta}$ given by Equation 1 may be evaluated,

$$T_{g_1,g_2,d,\theta}(r,c) = \begin{cases} 1 & , \quad (f(r,c)=g_1) \textit{ and } (f(r',c')=g_2) \\ 0 & , \quad \textit{otherwise} \end{cases} \tag{1}$$

where $r' = r + d\sin(\theta)$ and $c' = c + d\cos(\theta)$. The co-occurrence matrix $C_{d,\theta}$, can thus be defined as the accumulation of the values in $T_{g_1,g_2,d,\theta}$ for each gray level in the image, as given in (2)

$$C_{d,\theta}(g_1,g_2) = \sum_{c=1}^{C}\sum_{r=1}^{R} T_{g_1,g_2,d,\theta}(r,c) \tag{2}$$

As shown in Figure 2 the co-occurrence matrix for line drawing images has a maximum point located in the background-background region. Since scribbled images contain a small number of foreground pixels in comparison to the number of background pixels this maximum point will be present for all combinations of d and θ since, given any vector length and orientation, the number of background-background matches will always be greater than any other match combination. Figure 2 also shows a local maximum in the foreground-foreground region. Unlike the global maximum, a local maximum in this region of the co-occurrence matrix will occur only when the vector (d,θ) is aligned with the image foreground. Thus if the co-occurrence matrix were to be evaluated for a range of θ values, as shown in (3) the

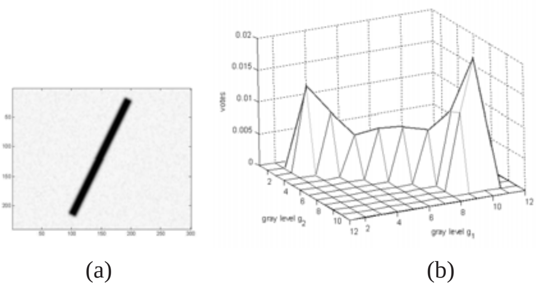


Figure 2: *A 3D plot of the co-occurrence matrix for the test line with orientation of 110° shown in (a). The matrix parameters were chosen as $d = 8$ and $\theta = 110^\circ$*

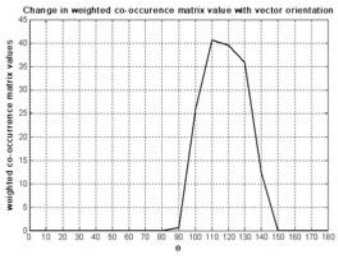


Figure 3: *The weighted co-occurrence vector*

presence of a local maximum in the foreground-foreground region of the matrix will indicate the orientation θ of the image line.

$$M_d(g_1,g_2,\theta) = \sum_{c=1}^{C}\sum_{r=1}^{R} T_{g_1,g_2,d,\theta}(r,c) \tag{3}$$

It is therefore beneficial to identify and separate the local maxima as these will indicate the line orientations in the image. This may be done by multiplying the co-occurrence matrix obtained for each value of θ with a weight matrix such as that given by Equation 4, which attenuates the background-background transitions.

$$W(g_1,g_2) = e^{\frac{(1-g_1g_2)}{(2^l-g_1)(2^l-g_2)}} \tag{4}$$

This matrix returns a value of 1 when both g_1 and g_2 are 1 (black) and a value of 0 when either one of g_1 or g_2 is equal to 2^l (white), where l defines the bit-depth of the image. We can now represent the co-occurrence matrix as a function of the orientation θ (5), indicating the significance, or otherwise, of foreground-foreground transitions in relation with orientation.

$$D_d(\theta) = \sum_{g_1=1}^{2^l}\sum_{g_2=1}^{2^l} W(g_1,g_2)M_d(g_1,g_2,\theta) \tag{5}$$

Figure 3 gives a plot of the weighted co-occurrence vector obtained for the image shown in Figure 2. As given in (6) the orientation $\hat{\theta}$ of lines in the image may be deduced from the maxima of this vector.

$$\hat{\theta} = \arg\max_{\theta} D_d(\theta) \tag{6}$$

Since images consist of many lines, it will be necessary to localise the orientation information derived from the co-occurrence matrix. In order to localise the orientation information, the image is split equally into sub-regions, using a quad-tree split for each subsequent region. This allows the co-occurrence matrix to focus on regions containing the line strokes, disregarding others which consist only of background pixels. The quad-split is carried out up to a certain resolution depth, which depends on the length d, since reasonable data may be obtained only for sub-regions whose size is greater than the value of d. Choosing d such that it is slightly larger than the stroke width, will ensure that the smallest subregion spans the width of a line stroke. The orientation deduced from each subregion may be applied to all the pixels within the region, thus representing the line strokes falling within that region with a single orientation vector.

Although the co-occurrence matrix can be used as described here to represent the scribble with less clutter, the 3D form represented by the scribble is generally ambiguous. Thus, in order to resolve the intended form, additional information about the 3D form has to be provided by the designer. Online systems provide the designer a set of tools via an interactive user interface, which the designer may use while drawing. Paper-based offline systems cannot provide an interactive interface, however, the designer may make use of specific pre-defined sketching languages which will aid the interpretation process.

4. The sketch based interface: what should it be like?

Several sketch-based interfaces have been developed, each catering for a particular drawing method. Interfaces such as Celesstin [VT92] and MDUS [DW99] may be used to convert 2D paper-based machine drawings to CAD systems. If the designer wishes to sketch directly in 3D, interfaces such as CIRGO [NCC*03] may be used. This interface distinguishes between different types of strokes according to the pressure with which they are drawn. Thus, the interface must be used in conjunction with tablet PCs, requiring the designer to reproduce the final sketch on an active medium. From a computational point of view, including symbols and using pre-defined drawing languages to incorporate the designer's intent will facilitate the interpretation process [FBC*05]. However, the designer would like an interpretation system that functions on sketches and scribbles that are as close as possible to the designer's natural drawing habits. The sketch based interface should therefore provide a suitable compromise between ease of use and ease of interpretation.

Ideally a sketch based interface should be capable of making a distinction between different view points and interpret the sketch accordingly. The interface should also support all kinds of geometries, including objects having 'form features' such as through or blind holes, and 3D primitives such as spherical or conical sections. Since the interpretation of the paper-based scribble requires the aid of symbols, these should be chosen such that they resemble standard drawing

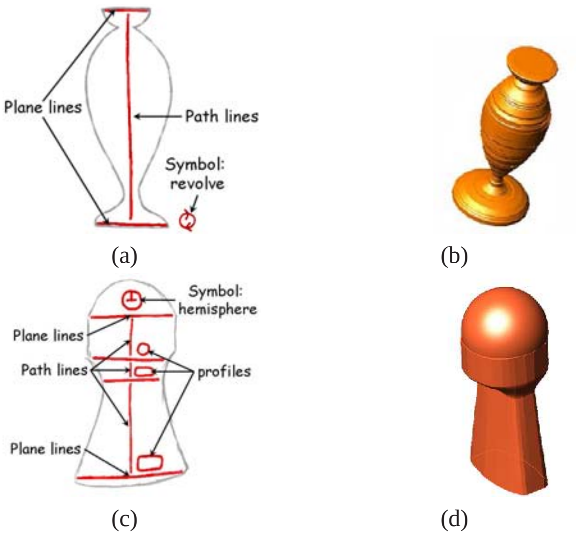


Figure 4: *An example of the sketching procedure*

symbols. This ensures that the symbols are easily remembered, making the language easier to use and develop.

5. The Annotator

The sketch based interface being proposed is a paper-based interface which uses the raw scribble drawn by the designer as the core mechanism for interpretation. This interface is a continuation of the interface described in [FBY*06], however, rather than requiring that the designer redraws the sketch according to some prescribed language, the designer is asked to annotate the scribble *after* this has been drawn. In this way, the sketch based interface is closer to the designer's natural drawing habits, hence achieving an easy-to-use interface. The annotations serve to facilitate the automated interpretation of the designer's scribble by the sketch-based interface and are made in such a way that they complement the designer's perception of the scribble.

In order to evaluate the concept proposed in this paper and develop a prototype tool which may be tested with designers, the extent of forms that can presently be represented by the implementation of the proposed sketch-based interface is limited to objects that may be represented using a single planar view and which contain a single axis. These include objects whose geometry may be described by rotation, extrusion or lofting operations. Furthermore, the sketch based interface is designed to extract the object's geometry; this work does not attempt to investigate the extraction or representation of form features, such as pockets or blind holes.

5.1. The annotation procedure

Figure 4 gives an example of two annotated scribbles and their corresponding 3D models. As can be seen in Figure 4(a) and (c), the designer is presented with four different annotation tools, namely cross-sectional profiles, plane lines, path lines and symbols. In order to facilitate the separation of the annotations from the scribble, the annotation is drawn in a colour that contrasts with the colour of the scribble.

Profiles are used to specify the cross-sectional shape of the object at particular planes orthogonal to the drawing plane. These are necessary to indicate the geometric shape of the object, which cannot be deduced from the scribble since this is assumed to be a front-elevational view of the object. The relation between the cross-sectional profiles and the scribble is shown by *plane lines*. These give the position and inclination of the plane at which the profiles are taken. It is not necessary to draw the cross-sectional profiles to scale, since their aspect ratio may be scaled according to the length of the plane lines. In some instances, the cross-sectional profiles may be replaced by *symbols*. The 'revolve' symbol in Figure 4(a) indicates that the intended form has a rotational symmetry along the nearest path line. In this case, profiles are unncessary since rotational objects must have a circular cross-section. On the other hand, in Figure 4(c), the symbol indicates that the object terminates with a 3D primitive, namely a hemisphere. *Path lines* are drawn in the center of the scribble and perpendicular to the plane lines. The meaning associated with the path lines differs according to the context in which they are used. In Figure 4(a), which represents a rotational object, only one path line was necessary and this reflects the axis of rotation of the object. Figure 4(c), which has four plane lines, three path lines - one between each pair of plane lines was necessary. In this case, the object is to be represented by a loft operation, and so the path line indicates the path along which the operation is to be carried out.

5.2. Interpreting the annotated scribble

The first step in the interpretation process would be to distinguish between the scribbles and the annotations. Since the designer makes the annotations in a different colour, the two may be separated on this basis. However, since the designer is free to use any two contrasting colours, it is also necessary to identify which of the coloured line strokes form the scribble and which form the annotations. This distinction is carried out after perceptual simplification of the two coloured components, since after simplification of the scribble, the number of disconected line strokes forming the annotations will be greater than the number of line strokes forming the scribble.

Once the annotations have been identified, they are further classified into path lines, plane lines, profiles or symbols. The scribbled drawing is used to help identify between the plane lines and path lines. This is done by taking two directed run-lengths perpendicular to the line in consideration at specific intervals along the line. Since the path line is drawn towards the center of the scribble, the directed run-

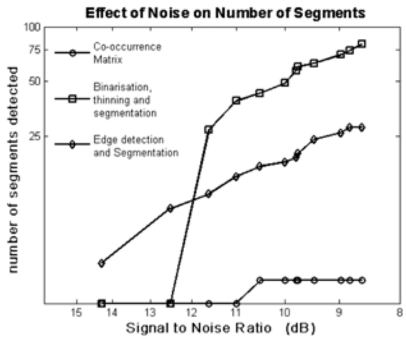


Figure 5: *Effect of noise on number of segments.*

lengths at each specific interval should be approximately equal. Thus if the average difference between the directed run-lengths along the line is less than a predefined threshold, the line is classified as a path line.

If a 'revolve' symbol is identified in the annotations, the interpretation process is only required to specify the shape profile to rotate and the axis of rotation about which the shape profile will be rotated. The axis of rotation is defined by the end points of the path line, whilst the shape profile is obtained from the scribble. If no rotational symbol is identified, it will be assumed that the 3D object can be generated by using a combination of extrude or loft operations. To achieve the 3D model, the annotations must first be grouped into sets. This is done by ordering the path lines such that the first path line is the bottom-most path line, and then locating the plane line and profile closest to each of the path line's endpoints. Thus, each annotation set will consist of a path line, two plane lines and two profiles. The minimum bounding box of each profile is used to determine the profile's size. The width of the minimum bounding box is compared to the length of the corresponding plane line and the profile is scaled accordingly. The position of the profile is then shifted such that the center of the minimum bounding box lies on the center of the plane line, thus aligning all the profiles in the drawing.

It is now necessary to identify the CAD operation required to propagate the profile from one plane to the next. This is carried out by taking the width of the scribble at intervals along the path line. An extrude operation will be assumed when the width of the scribble remains constant and the profiles at the start and end of the operation have the same geometric shape. If any one of these conditions fails, then a loft operation is assumed.

6. Results and Discussion

Perceptual simplification using the co-occurrence matrix is expected to have a good noise immunity since it is unlikely that random noise pixels give a high response to a particular vector (d,θ). This is verified in Figure 5 which compares the number of segments returned by binarisation, edge detection and the co-occurrence matrix for the image shown in Figure 1 which was corrupted with varying degrees of noise.

The co-occurrence matrix can also group together multiple line strokes falling within the same subregion. This is illustrated in Figure 6 which compares the line segments generated after processing part of a scribble with the co-occurrence matrix, binarisation and thinning and edge detection. Fewer and 'cleaner' line segments are generated by the co-occurrence matrix, making these segments more suitable for further processing by perceptual grouping algorithms.

Figure 8 shows the results obtained after applying the co-occurrence matrix on three test scribbles and a comparison of the performance of the co-occurrence matrix with simple binarisation and edge detection pre-processing techniques is given in Figure 7 (refer to `http:www.eng.um.edu.mt/~inpro/activities/` for further results). Comaparison of the computinal times yields average time of 80s for the co-occurrence matrix 5s for binarisation and 4s for edge detection, thus the co-occurrence matrix causes a 94% and 95% increase in computational times respectively. However, this additional computational time is compensated for by the decrease in the number of line segments generated, which reflects the computations required in subsequent perceptual grouping algorithms. The co-occurrence matrix results in an average of 9 segments whereas binarisation and edge-detection give an average of 200 and 485 segments respectively. Assuming a computational complexity of $O(n^2)$ defined in [GT99, GM93], where n is the number of line segments, pre-processing the images using the co-occurrence matrix reduces the computations required by 98% and 99% for the low noise images in comparison with the pre-processing by binarisation and edge detection tech-

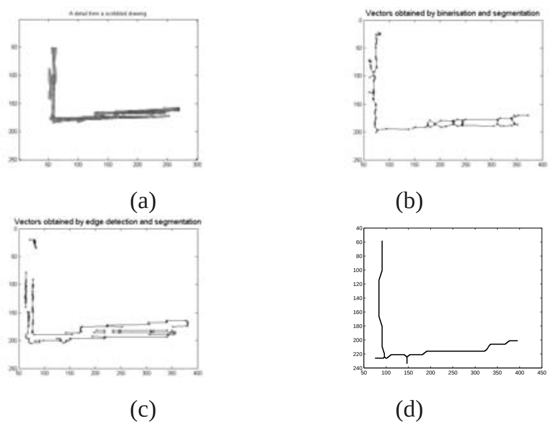
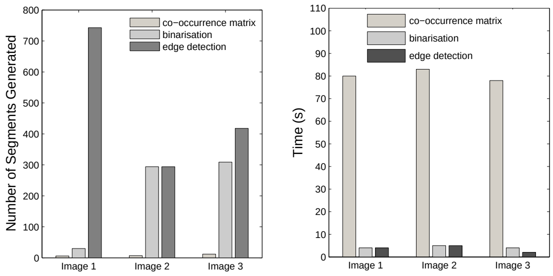


Figure 6: *A scribbled L-junction and its vector data after pre-processing with (b) Global binarisation (c) Sobel's edge detection (d) Co-occurrence Matrix.*



(a) (b)

Figure 7: *Comparison of (a) number of line segments and (b) computational times of the proposed algorithm and simple global thresholding and edge detection techniques.*

niques respectively. Simple binarisation and edge detection do not perform well under noise conditions, in fact, for the image shown in Figure 1 binarisation and edge detection would increase the computations of the perceptual grouping algorithm by a factor of 99.86% and 97% respectively. The co-occurrence matrix would only increase the computations by a factor of 44%, thus 55.86% and 53% less than the increase brought about by binarisation and edge detection.

7. Conclusion

The co-occurrence matrix determines the orientation of a pattern in a sub-region of the image. This effectively groups together multiple line strokes falling within that region, whilst giving a good noise immunity. Thus the co-occurrence matrix performs perceptual simplification of the image, making it easier for perceptual grouping algorithms to identify the most salient parts of the scribble. This is important in sketch-based interfaces, since the ability to interpret a designer's scribble brings the interface closer to the designer's natural drawing habits, and hence more user friendly.

8. Acknowledgements

This research is part of the research project 'Innovative Early Stage Design Product Prototyping' (InPro), (`http:www.eng.um.edu.mt/~inpro`) which is supported by the University of Malta under research grant IED 73-529-2005 .

References

[DW99] DORI D., WENYIN L.: Automated CAD Conversion with the Machine Drawing Understanding System: Concepts, Algorithms, and Performance. *IEEE Transactions On Systems, Man, and Cybernetics Part A: Systems and Humans 29*, 4 (July 1999), 411 – 416.

[FBC*05] FARRUGIA P. J., BORG J. C., CAMILLERI K. P., GIANNINI F., YAN X.: Extracting 3D shape models and related life knowledge from paper-based sketches. *International Journal on Computer Application in Technology (IJCAT), special issue on "Models and methods for representing and processing shape semantics. 23 (2/3/4)* (2005), 120 – 137.

[FBCS05] FARRUGIA P. J., BORG J., CAMILLERI K. P., SPITERI C.: Experiments with a Cameraphone-Aided Design (cpad) System. In *15th International Conference on Engineering Design (ICED05)* (2005), pp. 130 – 131.

[FBY*06] FARRUGIA P., BORG J. C., YAN X. T., CAMILLERI K. P., GRAHAM G.: A Sketching Alphabet for Paper-based Collaborative Design. *Journal of Design Research (JDR), special issue on Fostering Innovation During Early Informal Design Phases (to appear)* (2006).

[GM93] GUY G., MEDIONI G.: Iferring Global Perceptual Contours from Local Features. *Image Understanding Workshop* (1993), 881 – 892.

[GT99] GUICHARD F., TAREL J.: Curve Finder Combinng Perceptual Grouping and a Kalman-like Fitting. In *IEEE Conference on Computer Vision* (1999), pp. 1003 – 1008.

[HS01] HARALICK R. M., SHAPIRO L. G.: *Computer and Robot Vision*. Addison-Wesley, 2001.

[KH00] KELLEY A. R., HANCOCK E. R.: Grouping Line-segments using Eigendecomposition. In *Proceedings of the 11th British Machine Vision Conference* (2000), pp. 586 – 595.

[Mul01] MULLER W.: *Order and Meaning in Design*. Lemma Publishers, 2001.

[NCC*03] NAYA F., CONESA J., CONTERO M., COMPANY P., JORGE J.: Smart Sketch System for 3D Reconstruction Based Modelling. In *Third International Symposium on Smart Graphics* (2003), pp. 58 – 68.

[Sau03] SAUND E.: Finding Perceptually Closed Paths in Sketches and Drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 4 (Apr. 2003), 475 – 491.

[SH99] STAPPERS P. J., HENNESSEY J. M.: Towards Electronic Napkins and Beermants: Computer Support for Visual Ideation Skills. In *VRI'99* (1999), pp. 220 – 225.

[SU88] SHA'ASHUA A., ULLMAN S.: Structural Saliency: The Detection of Globally Salient Structures using a Locally Connected Network. In *Second International Conference on Computer Vision* (Dec. 1988), pp. 321–327.

[VT92] VAXIVIERE P., TOMBRE K.: Celesstin: CAD Conversion of Mechanical Drawings. *IEEE Computer Magazine 25* (1992), 46 – 54.

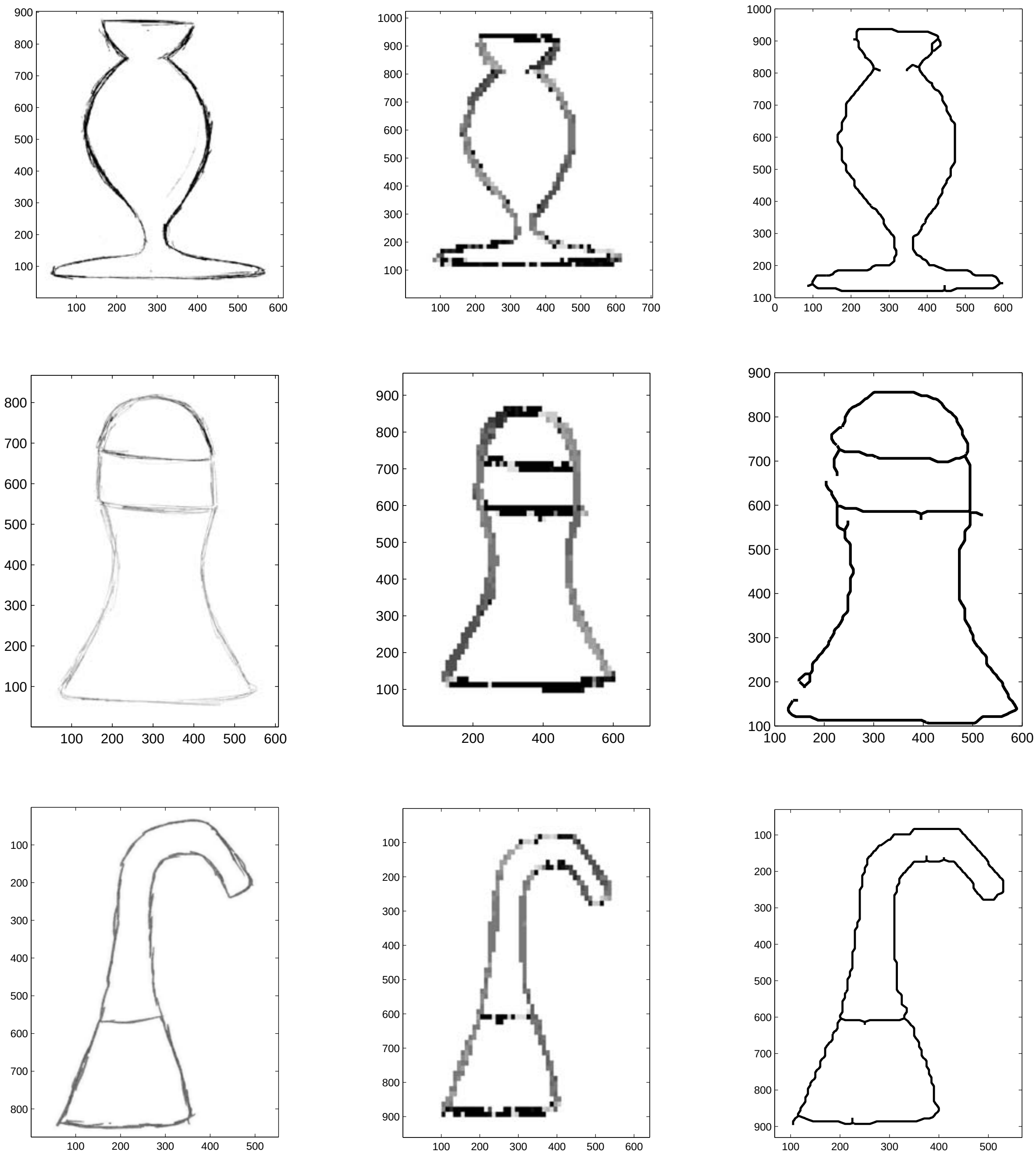


Figure 8: *The results obtained for three test scribbles which have been drawn with a black pencil on plane white paper and scanned at 96dpi. The first column shows the original scribble. The second column shows the orientations $\hat{\theta}$ obtained from the co-occurrence matrix, where black regions indicate an orientation of 0° and white represents an orientation of 180°. The last column shows the number of line segments that are extracted from the orientation response matrix. In these examples, the length of the vector d, used to evaluate the co-occurrence matrix, was set to 8*