EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (2012)
L. B. Kara and K. Singh (Editors)

Exploiting Artistic Cues to obtain Line Labels for Free-hand Sketches

A. Bonnici and K. P. Camilleri

Faculty of Engineering, University of Malta, Malta

Abstract

Artistic cues help designers to communicate design intent in sketches. In this paper, we show how these artistic cues may be used to obtain a line labelling interpretation of freehand sketches, using a cue-based genetic algorithm to obtain a labelling solution that matches design intent. In the paper, we show how this can be achieved from off-line or paper based sketches, thereby allowing designers greater flexibility in the choice of sketching medium.

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Depth Cues, Shading, Shape

1. Introduction

Freehand sketching is used in disciplines such as engineering design and architecture among others as an effective way of communicating design ideas. Such sketches are typically re-drawn using computer aided design (CAD) tools by means of which the designer further develops the form concept. CAD tools typically rely on window, icon, menu and pointer (WIMP) interfaces making CAD interfaces cumbersome, especially for non-expert users. Ideally, CAD tools would be able to automatically interpret the user's freehand sketch, allowing the user to obtain 3D models of drawings with minimal effort. This is known to be a difficult problem due to the inherent ambiguity present in 2D sketchs [Hof00]. Sketch-based interfaces (SBIs) address this problem by replacing the WIMP interface with a more natural interface, allowing the user to interact with the machine by means of sketching strokes [OSSJ09]. The SBIs therefore resolve the possible ambiguity in design intent by requiring the user to use sketched gestures to guide the interpretation of the drawing. However, when sketching freely, designers resolve ambiguities by means of artistic cues which are drawn in addition to the object contour. These cues allow the designer to establish design intent, such that other human observers can reach the same interpretation of what could be an otherwise ambiguous sketch [Pip07]. Introducing artistic cues in the interpretation of the designer sketch therefore enhances the SBI, allowing it to become more transparent to the designer [OSSJ09].

This paper studies a line labelling approach to sketch interpretation. By describing each object edge as concave, convex or occluding, a line labelling algorithm should provide sufficient geometrical and spatial information about the drawing to provide the means of obtaining an initial inflation of the sketch [Coo01]. A sketch however may have multiple interpretations such that it is necessary to determine the most plausible interpretation of the sketch [LS96]. Since designers resolve ambiguity by means of artistic cues, we study the way cues are used in freehand sketches and note how the cues are used to imply particular interpretations. We also show that by means of a simple sketching language, artistic cues can be used to tune a line labelling algorithm such that the sketch interpretation matches design intent. The rest of this paper is organised as follows: Section 2 presents an overview of artistic cues used in freehand sketches; Section 3 describes a cue-based genetic algorithm (cGA) solution to the line labelling problem; Section 4 describes the sketching language and image processing necessary which allows the cGA to be applied to paper-based sketches or sketches processed off-line; Section 5 presents the evaluation methodology; Section 6 discusses the results obtained while Section 7 concludes the paper.

DOI: 10.2312/SBM/SBM12/077-086

Table 1: *A summary of artistic cues and their semantics. Cues in bold print refer to the cues used in this paper*

Cue Description			Cue Semantics				
			material information	shape information	spatial information	depth	surface feature
Lines	Existing Edge	line phrasing				✓	
		haloed lines				✓	
	Extra Lines	accent lines					✓
		table lines			✓		
Tone	Illumination Effect	**cast shadow**		✓	✓	✓	
		attached shadow		✓			
	Non-Illumination	shading	✓				✓

2. Cues used in freehand sketches

Although an object contour is generally sufficient to give an observer an impression of the form of the object, artists and designers typically introduce artistic cues that make the sketch more visually pleasing while augmenting the information presented in the sketch. Different cues such as illumination, texture, tone differences and line weights among others may be added to the sketch, each introducing information about various aspects of the form concept, allowing designers to better communicate conceptual ideas to an observer. In this section, we study the effect of cues on the interpretation of drawings, using literature of artistic sketching and non-photorealistic rendering to identify the cues most commonly used in diagrams. In addition, in order to obtain a better understanding of the cues used in practice, we asked a number of designers to submit initial sketches from existing portfolios. Although authors such as [JD09] among others have proposed alternative methodologies by which sketch data can be collected from participants, we chose this form of data collection to ensure that the sketches collected are a true representative of the designer's sketches and free from any preconceptions on the cues that should be included in the sketch. Table 1 summarizes the observations made.

Designers can enhance the depth perception provided by the selected projection method by means of differences in line weights, using a technique known as *line phrasing* [CSP03, Elb95]. This involves changing the width and intensity of the line strokes forming the object such that bold, dark lines are used to represent objects which are in the forefront of the scene becoming fainter and thinner as the objects or parts of the object recede into the background [Elb95]. Line phrasing is also used to differentiate between visible and hidden edges of the object when it is necessary to illustrate the hidden edges in the sketch. In such cases, it is common to represent the hidden edges using fainter line strokes than those used to represent the visible edges. Associated with line phrasing is the use of *haloed lines* [Elb95]. This involves tapering the edges that are occluded by or pass behind other edges, giving the viewer the impression that the occluded edge is further back than the occluding edge as shown in Figure **??**. Haloed lines introduce gaps in the object edges which are often proportional to the angle of intersection between the occluded and occluding line strokes.

Depth perception can also be obtained by means of illumination cues. Unlike line phrasing, illumination cues reflect physically observable characteristics of a scene and reflect the depth of the object by means of illumination differences on the object surfaces as well as the relationship between objects in the scene by means of cast shadows. Illumination cues also serve

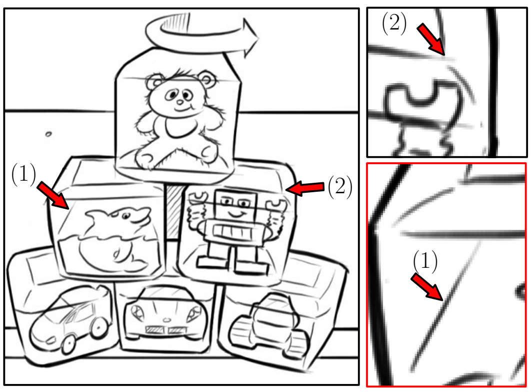
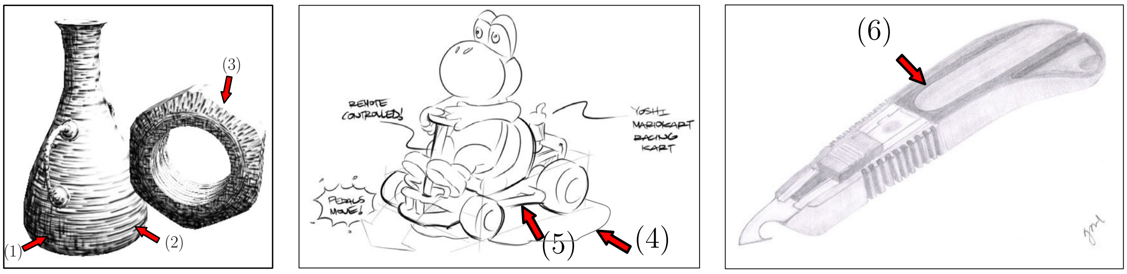


Figure 1: *Examples of haloed lines and line phrasing: Edges like (1) that are distant from the observer are sketched with fainter strokes than those which are closer while intersecting edges like (2) are tapered to create* haloed lines.



(a) (b) (c)

Figure 2: *Illumination variances can be represented by cross-hatching (1) or very faint lines (2) while the direction of the hatch strokes can also define object shape (3). Shadows can also be abstracted with the shadow boundary (4) or as thick, bold, lines (5). Shading is also used to illustrate the use of different materials in the object as shown in (6).*

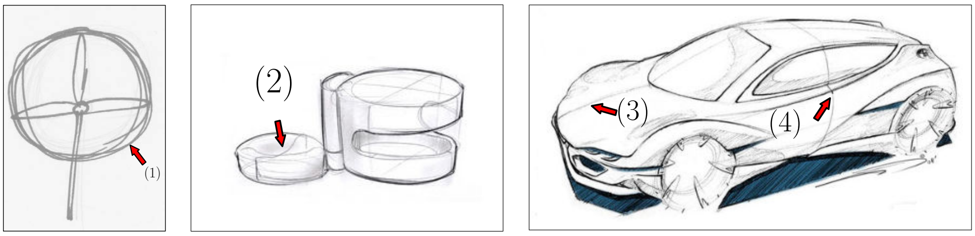
to give information about the object form and can be used to distinguished between flat and curved surfaces. When sketching, illumination differences may be portrayed by changes in tone. This may be done in two ways, namely by introducing constant or variable changes in tone intensity. Designers sketching in pen and ink can achieve this by using hatching techniques [Gup97]. Constant tone regions are obtained by using hatch strokes of equal spacing while variable tone changes can be achieved by varying the space between hatch strokes, the line weight of the strokes and by using cross-hatching [Gup97]. The direction of the hatch stroke is not chosen arbitrarily but selected such that the hatch stroke follows the principal direction of the surface [ZISS04]. Thus, as shown in Figure 2(a) hatch strokes give further evidence on the form of the surface.

Although shade and shadows have an important role in allowing observers to perceive the 3D structure of physical objects, when these are represented as sketches designers do not necessarily need to sketch exact representations of the illumination effects such that it is common for shade and shadows to be abstracted as lines [LMLH07]. Thus, shadows are at times represented by thick black lines or by a shadow boundary as shown in Figure 2(b). It is also possible to represent brightly lit or reflective surfaces with white or light-coloured lines. This implies that the interpretation of sketches requires the interpretation of coloured and non-coloured spaces. One may take this observation a step further to note that at times, supporting cues are sufficient to indicate the shape of the object without the need for specifying the object edges. An example of this can be seen in Figure 2(a) where the edge of the nut is missing although it is implied by neighboring edges.

Tone changes are not used exclusively to represent illumination effects. In fact as shown in Figure 2(c), designers also use tone changes in conjunction with texture to illustrate the different components and materials forming the object. While such cues do not necessarily provide additional information on the 3D structure of the object, they serve to enhance the aesthetic aspect of the sketch while providing useful information regarding the usage and functionality of the object. Other abstract concepts such as motion and speed among others can also be portrayed by introducing additional line strokes as shown in Figure 3(a) [Elb95].

Additional line strokes are also used by designers wishing to reinforce some aspect of the object form. These lines which are referred to as *accent* lines can be used to emphasize curvature or to generate ridge-line discontinuities in otherwise smooth surfaces as shown in Figure 3(b), 3(c) [CSP03]. In such cases, although planes and surfaces can be distinguished even if the accent lines are not present, their addition reinforces the interpretation of the surface. Additional lines may also be added when the designer needs to give the impression of an object's background without the need to specify the actual background [ES07]. Such lines, which are sometimes referred to as *table lines* are commonly used to indicate that the object is resting against something rather than hanging in mid-air.

Shadows and table lines appear to be prominent cues, featuring often in sketches. Shadows may be further subdivided into cast and attached shadows where cast shadows are formed when surfaces occlude each other from a light source while attached shadows are formed when light does not fall onto a surface of the object [MKK98]. Cast and attached shadows therefore provide different information about the object: cast shadows provide spatial information about objects in the sketch while attached shadows provide local information about the object edges. Table lines complement and support the spatial information portrayed by the cast shadows by providing information about the spatial relationship between the object and its back-



(a) (b) (c)

Figure 3: *Additional line strokes can be used to represent abstract concepts such as motion (1) but are also used to highlight features in the object surface (3) - (5).*

ground. Thus the semantics that these cues add to the sketched edges can be represented by edge labels which can be determined by applying edge labelling algorithms. Therefore this paper investigates the use of shadows and table lines to augment an edge labelling algorithm with cue semantics.

3. Genetic algorithm approach to line labelling

Machine interpretation of diagrams, specifically, the 3D construction of objects depicted in drawings can be achieved by either solving planar equations [LCLT08, RT02] obtained from the diagram or by optimizing some cost function related to the ideal geometry of an object [CGC99, LF92, LS96, PMC03]. While these methods determine the depth coordinates of salient points of the object, other techniques, namely line labelling techniques provide an initial interpretation of the drawing from which an initial inflation of the diagram can be obtained.

Huffman [Huf71] and Clowes [Clo71] proposed a labelling scheme for trihedral objects in which edges occur due to the intersection of two planes S_1 and S_2. These edges can be described as convex if the exterior angle between the two planes is less than π, concave if the exterior angle is greater than π and occluding if either one of S_1 or S_2 is not visible. Convex edges are assigned the label $+$, concave edges the label $-$ while occluding edges are assigned the label $\rightarrow$ with the direction of the arrow being such that the occluded or invisible plane occurs on the left-hand side of the line. This labelling scheme has been extended to include the labelling of more generic drawings such as tetrahedral objects [VM01], curved objects [Coo08] as well as drawings which capture illumination changes [Wal75, Coo01]. The labelling scheme has also been adapted to introduce new edge semantics. In particular, Waltz [Wal75] introduced a new edge label $\overset{\rightarrow}{-}$ to distinguish between true concave edges which occur when two planes of the same object intersect to form a concave edge and apparent concave edges which are formed when two separate objects are placed adjacent to each other such that an occluding edge of one of the objects touches the planar surface of the second object or background wall as shown in Figure 4. Such edges are referred to as *crack* edges.

In edge labelling literature, drawings are typically described in terms of junctions which in the case of trihedral objects, are categorised as W, Y, T and L junctions as shown in Figure 4. Junction dictionaries Γ describing legal labels for each junction are then used to determine the proper labels for a new drawing. Labelling algorithms typically assume neat, accurate drawings obtained either by means of edge detectors from 2D scenes or by means of vectorization of neat drawings. For such drawings, the junction dictionaries Γ are typically used as hard constraints, pruning out impossible labels for the edges at a new junction. The difficulty with such an approach is that ambiguous or misaligned junctions may have the most valid labels disallowed because the 2D drawn junction does not satisfy the hard constraints imposed by the labelling algorithm.

Furthermore, while algorithms described by Waltz [Wal75] and Cooper [Coo01] take into account the presence of illumination changes, they assume that these will be represented by their edge boundaries and that they are a true representation of illumination changes in the scene. Waltz and Cooper therefore introduce additional hard constraints with which they are able to label the object and shadow edges. In sketches however, designers typically abstract illumination changes and this abstraction is subjective to the designer, depending mostly on the mental image that the designers would like to impart to the observers. This suggests that in sketches, illumination representations are better described as suggestive cues allowing the observer to consolidate the interpretation of an edge rather than hard constraints which enforce a particular interpretation.

Myers and Hancock [MH00] propose an alternative approach to the line labelling which uses a genetic algorithm (GA) to obtain the best labelling scheme for the drawing through an evolutionary process which uses the dictionary Γ to determine the fitness of the selected labels. In this way, the GA uses the junction label dictionary as soft constraints and while penalizing labels that are not found in the dictionary, it does not disallow the label from being used. This approach is attractive for the problem of labelling sketches since ambiguous or incorrect drawings can be labeled with the best possible label scheme and the algorithm will list alternative solutions which are ranked in order of fitness. Thus, rather than classifying ambiguous drawings as impossible to label, the algorithm described by Myers and Hancock [MH00] will list the most likely labelling schemes.

Using this approach, the labelled drawing is described as a chromosome E consisting of N genes where N is the number of edges in the drawing. Each gene describes an edge label $\lambda_i \in \Lambda$ where Λ is the list of all possible edge labels that can be assigned to an edge, that is, $\Lambda = \{+,-,\rightarrow,\leftarrow,\overrightarrow{-},\overleftarrow{-}\}$. The chromosome is therefore defined by $E = \{\lambda_i, \cdots \lambda_N\}$. The objective of the GA is to evolve a population of chromosomes such that the genes of the chromosome are valid interpretations of the drawing edges. To do so, the GA must assess the fitness of the chromosome by comparing the genes to the legal labels defined in the junction dictionary Γ. Thus, the drawing is subdivided into a list of junctions $J_k, k = 1 \cdots K$ where K is the number of junctions in the drawing such that a subset $E(J_k) \in E$ that gives the edge labels at junction J_k can be obtained from the chromosome E. The Hamming distance between $E(J_k)$ and the legal labels for that junction, defined in Γ, determine the fitness of the junction and therefore, the summation of the Hamming distance for each junction $J_k, k = 1 \cdots K$ will determine the fitness of the chromosome [MH00].

When artistic cues are introduced to the drawing, they effectively constrain the interpretation of the relevant edge to a subset of the possible interpretations of that edge. By observing the semantics of cues used in drawings, it is possible to create a second dictionary that maps a cue to a constrained set of possible interpretations [BC12]. We refer to this dictionary as the *cue constraint filter* (CCF) since the role of this dictionary is to filter out edge interpretations that do not match the semantics of the cue acting on the edge. Thus, our cue-based GA (cGA) is initialised with a chromosome population in which edges having a cue acting on them are labelled with a label chosen from $\Lambda(n)$ which is a subset of the possible label set Λ and defined as $\Lambda(n) = \Lambda \cap CCF(C(n))$ where $CCF(C(n))$ represents the semantics attributed to the cues $C(n)$ which bear upon an edge n whose gene is represented by $g_n, 1 \leq n \geq N$.

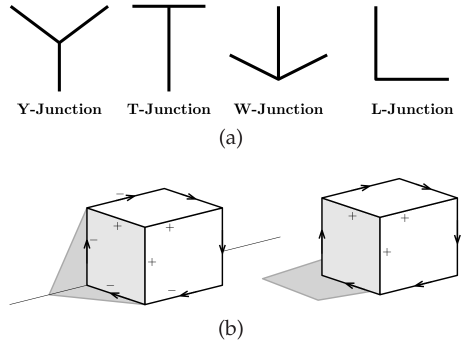


Figure 4: *(a) Y, T, W and L junctions that are found in trihedral objects. (b) An example of how shading cues and table lines determine the interpretation of a sketch and illustrating the difference between the edge labels.*

Thus, the initial population of the cGA is expected to be close to the intended interpretation of the sketch.

Despite the constraint imposed on the initial population, the mechanisms of cross-over and mutation allow the GA to change its chromosome and hence explore the search space. However, in so doing, the initial design intent information prompted by the artistic cues may be lost through the evolutionary process. For this reason, the CCF is used to augment the chromosome fitness by introducing a penalty function related to the cues in addition to the Hamming distance fitness associated with the legal labels defined in Γ. We define this penalty function as:

$$P_n = \begin{cases} \frac{1}{N} & \text{if } \Lambda(n) \neq \emptyset, \lambda_i \notin \Lambda(n) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

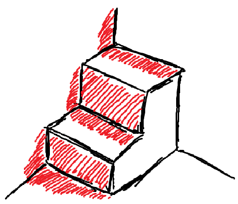
such that the chromosome fitness may be defined by:

$$F(E) = \alpha \left(\frac{1}{2N} \sum_{k=1}^{K} \min_{l=1,\cdots,|\Gamma|} H(E(J_k),\Gamma) \right) - (1-\alpha) \left(\sum_{n=1}^{N} P_n \right) \quad (2)$$

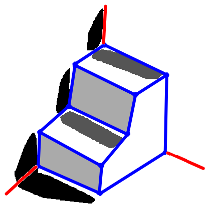
where H denotes the Hamming distance and α is a weight factor that determines the confidence in the cues. Thus, although the CCF imposes hard constraints on the initial population, it acts as a soft constraint, through $F(E)$, in the following generations.

4. Preparing the sketch

In this section we describe an initial attempt in adopting the cGA to the labelling of freehand sketches. We assume that the objects depicted in the sketch are trihedral objects which are drawn from a generic viewpoint, that is, slight perturbations of the viewpoint would not change the representation of the object. This latter assumption poses no particular restriction on the



(a)



(b)

Figure 5: *A sketch and its interpretation. In (b) light grey regions represent the attached shadow, darker grey regions the cast shadows located within the planes while the black regions the cast shadows not located within planes.*

artist since sketches are naturally drawn in this manner [Hof00]. We also assume that the objects are illuminated from a single point diffuse light source.

The separation of sketched shading cues from the object contour is not an easy task, particularly since as cues are subject to artist idiosyncracies. Hatch detection techniques such as [LMLK99, TBM09] among others assume that the hatched strokes are uniform in width and spacing. This however, is not necessarily the case with sketched hatched regions, adding to the difficulty in separating hatched regions from the object contours. Rather than severely limiting artist freedom or manually labelling all edges, we propose a simple user interface which requires that the drawing is sketched in two colours, reserving black for the contours and table lines and using any other contrasting colour to sketch the shading. In addition, we assume that the attached shadow is sketched such that it is spread across at least 80% of the plane to which it is attached. The sketch can be drawn using any preferred inking software or on plain paper.

To prepare the sketch for interpretation, k-means clustering is used to determine wether each non-white pixel belongs to the set of black strokes or to the set of coloured strokes, hence effectively distinguishing between the shading strokes and the black contour and table line strokes. The latter are subsequently thinned to a single width representation after which they are modelled by polylines as described in [SG80]. Since the objects are trihedral, planar objects, each polyline represents a single edge of the drawing and the connectivity of these endpoints is used to organise the drawing into junctions J_k by finding all other edges with which it is connected. Open-ended polylines are simultaneously identified as table-lines. The orientation of the edges forming J_k are then used to classify the junction as being one of $\{W, L, T, Y\}$.

The shading strokes are then split into individual shaded areas by fitting polygons around the shaded regions, using the contour edges and table lines obtained previously to refine the shaded areas such that the individual shaded regions are identified. It is then necessary to distinguish between cast and attached shadows. Since we require that the drawing is sketched from a generic viewpoint, cast shadows should not cover an entire plane. Thus, if the polygon fitting the shaded area occupies more than 80% of the plane, the shaded region is labelled as an attached shadow.

Once the cues are identified, the edges they bear upon are identified by using proximity of the contour edges to shade boundary in the case of cast and attached shadows or line endpoints in the case of the table lines. The cue information obtained from the sketch can then be compared to the CCF, thus identifying $\Lambda(n)$ for each edge in the drawing.

5. Evaluation methodology

To establish the performance of the cGA, it is necessary to monitor the evolutionary process and verify that the cGA does indeed converge to the solution that reflects design intent. Quantitative performance of the cGA can be obtained by measuring the population entropy, span and fitness throughout the evolutionary process [MH00]. The population entropy, defined as

$$S = -\sum_{i=1}^{|\Psi|} \rho_i \log \rho_i \tag{3}$$

where Ψ is the set of distinct chromosomes E in the population and ρ_i is the proportion of occurrences of the chromosome E_i in Ψ, measures the number of distinct chromosomes within the population. In a standard GA, the entropy is expected to be large since chromosome strings are intialised with random values spread across the search space. As the population evolves however, the chromosomes are expected to converge to a desired solution such that the entropy should decrease. The population span is defined as

$$H_t = \sum_{i=1}^{|\Psi|} \sum_{j=i+1}^{|\Psi|} H(E_i, E_j) \tag{4}$$

where $H(E_i, E_j)$ is the Hamming distance between chromosomes E_i and E_j. The span too is expected to be large initially, indicating that the random chromosomes are well spread in the search space. As the evolutionary process takes place and the chromosomes converge to the desired solution, the span is expected to decrease. On the other hand, population fitness is

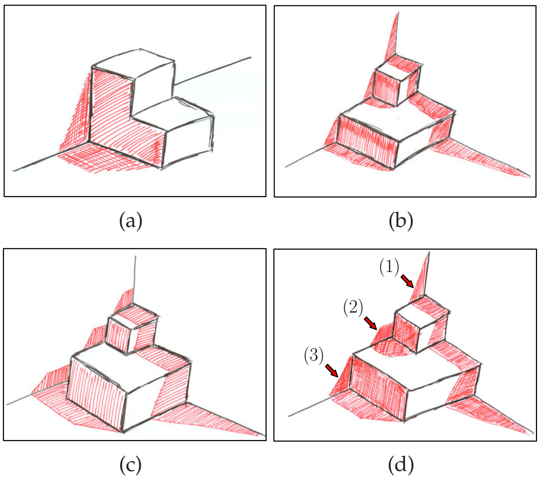


Figure 6: *Sample sketches on which the cGA is evaluated.*

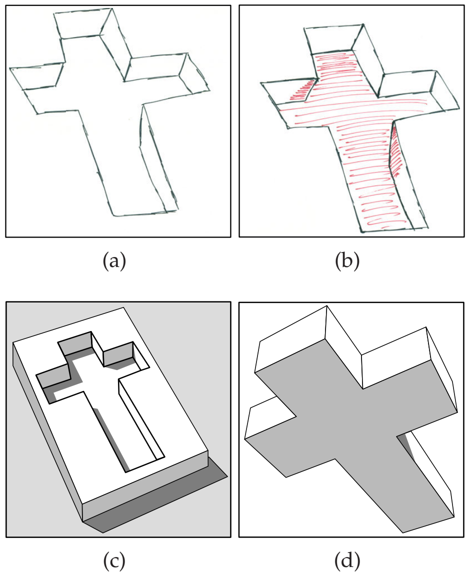


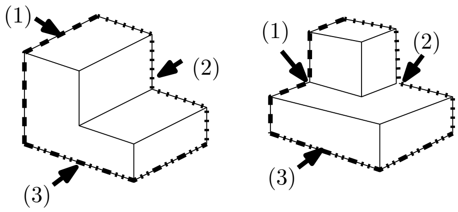
Figure 7: *An ambiguous sketch which can be interpreted as either a hollow cross as shown in (c) or a cross seen from below as shown in (d). The cues present in sketch (b) suggest that the sketch should be interpreted as shown in (d).*

expected to increase as the population evolves, indicating that the population is moving towards the expected solution.

In order to determine that the cGA does use the cues to evolve the population to the intended solution, the cGA was evaluated on test drawings such as those shown in Figure 6 and Figure 7 where the cues can alter the interpretation of the sketch. In the absence of cues, the edges forming the outer contour of the objects in the sketches of Figure 6 can have five possible interpretations as summarised in Figure 8. Edges shown by the dashed lines and labeled (1) form an edge chain which must have the same label so that the drawing has a valid label solution. Similarly edges grouped under edge chains (2) and (3) must also have the same label. This leads to the five possible interpretations given in Figure 8. The presence of cues however, primes the observer to select one solution as the intended interpretation of the sketch. Figure 6(a) should be labelled according to interpretation (iii), Figure 6(b) according to interpretation (v) and Figure 6(c) as (iii). This sketch has a missing cue and therefore allows us to observe the performance of the cGA when cues are missing from the drawing. The cues present in the sketch of Figure 6(d) are conflicting, with the shading labelled (1) and (3) indicating interpretation (v) while shading labelled (2) indicates interpretation (iii). It is logical to assume that the intended interpretation of this drawing is that represented in (v) since this interpretation is supported by the majority of the cues. This sketch therefore allows us to determine the performance of the cGA in the presence of cue inconsistencies.

The drawing in Figure 7(a) is an intentionally ambiguous drawing which can be interpreted as either a hollow cross or as a cross seen from below, although only the hollow interpretation has junctions which exhibit a valid geometry as described in the junction dictionary Γ. The interpretation with the cross seen from beneath can however be reinforced by the addition of shading cues as shown in Figure 7(b) such that it would be desirable for the cGA to obtain a line labelling that reflects this interpretation. Thus, this drawing allows us to observe the performance of the cGA under conditions where the intended interpretation has geometric inconsistencies.

The performance of the cGA can be compared with that of the cue-less GA in order to verify that the introduction of the cues in the cGA does improve the selection of the intended interpretation. For comparison purposes, the fitness of the final solution obtained by the cue-less GA was evaluated using the stricter fitness function used for the cGA. This will allow us to identify the fitness of the GA with respect to the design intent and hence obtain a fair comparison with the proposed cGA. In both cases, the genetic algorithms were implemented with proportionate fitness selection, a 1-point crossover with a rate of 0.9 and a mutation rate of 0.03 with a population size of 100. The population span, entropy and fitness of the cGA were evaluated over 50 trials of 500 generations. In addition, since the fitness function is dependent on the selection of the parameter α which determines the confidence in the cues present in the sketch, the cGA was evaluated using dif-



	Interpreation	Edge chain (1)	(2)	(3)
(i)	floating object	$\rightarrow$	$\rightarrow$	$\rightarrow$
(ii)	resting on table	$\rightarrow$	$\rightarrow$	$\underline{\rightarrow}$
(iii)	against left wall	$\underline{\rightarrow}$	$\rightarrow$	$\underline{\rightarrow}$
(iv)	against right wall	$\rightarrow$	$\underline{\rightarrow}$	$\underline{\rightarrow}$
(v)	in a corner	$\underline{\rightarrow}$	$\underline{\rightarrow}$	$\underline{\rightarrow}$

Figure 8: *Possible interpretations for sketches shown in Figure 6*

ferent values of α in order to determine the effect that this parameter has on the performance of the cGA.

6. Results and discussion

The selection of the value of α determines the confidence in the cues present in the drawing and has an impact on the population fitness and hence the evolutionary mechanism of the cGA. This can be observed in the graph shown in Figure 9 which gives the mean fitness and the mean best fitness of the population obtained for the sketch shown in Figure 6(a), evaluated for α values in the range $[0.1, 0.9]$. In this graph one may observe that there is a sharp drop in fitness for values of α greater than 0.7. This implies that at these values of α the evolutionary mechanism produces chromosomes that have a larger improvement in their fitness value when they are evolved to match the junction dictionary rather than the restricted label set defined by the CCF. This results in interpretations which while being geometrically correct, do not reflect design intent. Smaller values of α force the cGA to give greater importance to the CCF, forcing a stricter adherence to the interpretation suggested by the cues in the drawing. This can be observed in Figure 9 where for values of α less than 0.2, the mean fitness reaches the maximum fitness value indicating that all the population converges to the desired solution. Such confidence in the cues is however undesirable since cues are not necessarily drawn correctly as can be seen in Figure 6(d). Blind faith in the cues in this drawing would result in an impossible interpretation of the sketch which would have the edges labelled as (1) in Figure 8 being interpreted alternately as ‘against a left wall’ and ‘in front of a left wall’. This interpretation is represented as (vi)

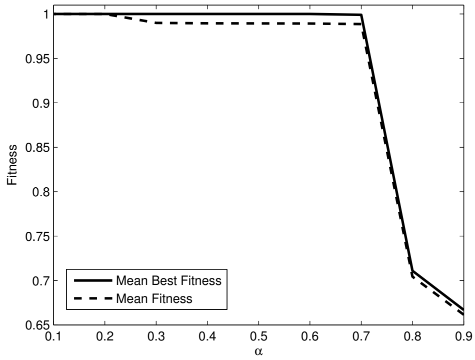


Figure 9: *Graph showing the change in fitness with α for the sketch in Fig, 6(a)*

Table 2: *Different interpretations obtained by the cGA for the sketch in Figure 6(d) using different values of α. The column highlighted indicates the most plausible interpretation for this sketch which contains a wrong cue.*

α	% of interpretation occurrences (i)	(ii)	(iii)	(iv)	**(v)**	(vi)
0.8	0	16	30	4	**46**	4
0.6	0	2	4	4	**68**	22
0.4	0	0	0	0	**46**	54

in Table 2 which gives the number of occurrences of each interpretations for different values of α.

In all cases, chromosomes with interpretation (v) were identified as being the most fit. With $\alpha = 0.8$ the cGA obtained the smallest number of geometrically incorrect interpretations, but this value of α also resulted in 50% of the interpretations disregarding the implied interpretation of other cues in the sketch. On the otherhand, with $\alpha = 0.4$ the cGA has a strong belief in the cues and as a result, the majority of the trials converged to a solution which is geometrically incorrect. Thus, although low values of α tend to give solutions which reflect the interpretation portrayed by the cues, midrange values of α may be more suitable if the sketch contains inaccurate cues. For the rest of this evaluation, the value of α is set at 0.6.

The span and entropy of one iterate of the cGA on Figure 6(b) are shown in Figure 10. One may note that the entropy is initially large but decreases and reaches a steady-state as the population converges to a solution. Local peaks in the entropy occur due to the evolutionary mechanisms of cross-over and mutation which introduce new clusters into the drawing. Peaks in the entropy are coupled with peaks in the population span, indicating that the clusters introduced through the evolutionary process are significantly diverse from other existing clusters. This is important in the evolutionary process since it allows the cGA to search for solutions other than those suggested by the CCF. This is important for the cGA in the case of errors or inconsistencies in the cues. The fitness plot given in 10(c) is the average maximum fitness of 50 iterations of the cGA on this same sketch. This shows that the restricted initial population of the cGA is placed in a strategic pace within the search space such that the initial population has a relatively high fitness value which is then further improved by the evolutionary process.

Table 3 compares the solutions obtained by the cGA with those of the cue-less GA for the sketches in Figures 6(a), 6(b) and 6(c). From this table, one may observe that while both the GA and cGA converge to a solution that is geometrically correct, the cGA is more consistent with the design intent and therefore achieves a mean best fitness value that is larger than that of the GA. These results show that the CCF is effective in guiding the GA towards the intended interpretation of the sketch. This has also been observed in the interpretation of Figure 7(b) where due to the presence of the cues, the cGA converged to the interpretation portrayed in Figure 7(d) although this interpretation has a fitness value of 0.866 due to the incorrect geometry of this interpretation. On the other hand the GA identified only the interpretation of Figure 7(c) as the interpretation of this sketch.

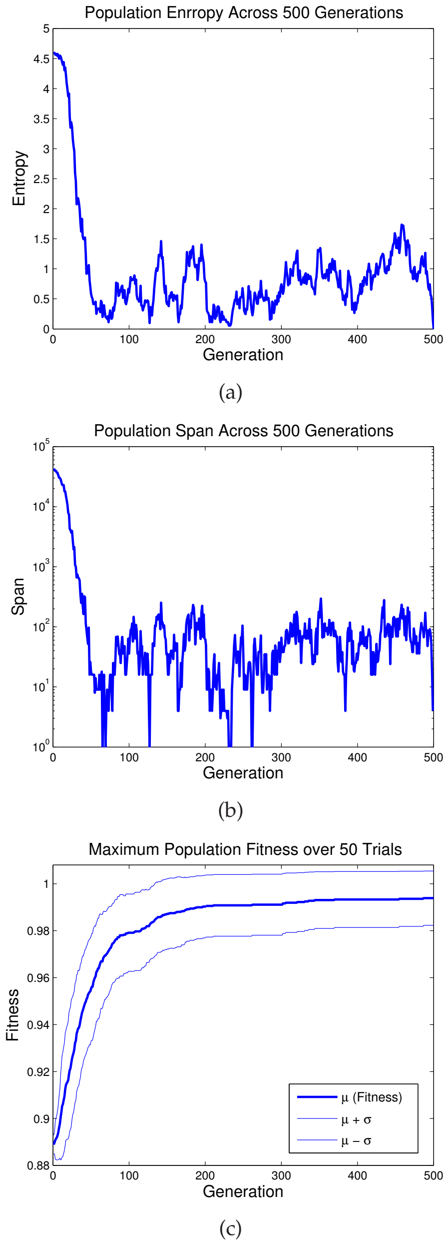


Figure 10: *Entropy, Span and Maximum Fitness curves for Figure 6(b).*

7. Conclusions

In this paper we show that artistic cues can be used to reduce the ambiguity in the interpretation of freehand sketches, obtaining an interpretation that matches design intent. This approach may be improved if the designer is allowed to sketch freely, using just one colour. This requires further investigation in pattern analysis to allow the distinction between the hatched regions and the sketch strokes. Further improvement could be achieved if the parameter α is determined from the confidence with which the sketch preparation step identifies and associates the cues with the edges in the sketch. Thus obtaining an α value for each edge in the sketch.

References

[BC12] BONNICI A., CAMILLERI K.: Genetic algorithm for line labelling of diagrams having drawing cues. In *Proceedings of the 7th International Conference on the Theory and Application of Diagrams* (2012). 5

[CGC99] COMPANY P., GOMIS J. M., CONTERO M.: An optimization-based algorithm to reconstruct 3d models from single line drawings. In *II Seminario Italo-Español, "Diseño y fabricabilidad de los productos industriales"* (March 1999), A. L., Giannini P. S., (Eds.), pp. 3–12. 4

[Clo71] CLOWES M. B.: On seeing things. *Artificial Intelligence 2*, 1 (1971), 76 – 116. 4

[Coo01] COOPER M.: The interpretation of line drawings with contrast failure and shadows. *International Journal on Computer Vision 43*, 2 (2001), 75–97. 1, 4

[Coo08] COOPER M.: A rich discrete labeling scheme for line drawings of curved objects. *IEEE Transactions on Pat-*

Table 3: *A comparison of the solutions obtained by the cGA and the GA for sketches 6(a), 6(b) and 6(c). Columns 3-7 list the number of times the GA and the cGA obtained interpretations (i) - (v) respectively together with the fitness of each interpretation. Column 8 then gives the mean fitness achieved by each algorithm. Bold values indicate the desired interpretation. For comparison purposes, the GA solution was re-evaluated using the stricter fitness function of the cGA.*

		% of interpretation occurrences (fitness of solution)					
	Diagram	(i)	(ii)	(iii)	(iv)	(v)	mean best fitness value
GA	6(a)	30 (0.877)	25 (0.938)	**20 (1)**	15 (0.785)	10 (0.877)	0.903
	6(b)	16 (0.778)	24 (0.822)	22 (0.911)	22 (0.911)	**16 (1)**	0.883
	6(c)	16 (0.778)	24 (0.911)	22 (0.822)	**22 (1)**	16 (0.911)	0.890
cGA	6(a)	0	0	**100 (1)**	0	0	1
	6(b)	0	0	30 (0.911)	0	**70 (1)**	0.973
	6(c)	0	24 (0.911)	0	**76 (1)**	0	0.979

tern Analysis and Machine Intelligence 30, 4 (April 2008), 741–745. 4

[CSP03] COSTA SOUSA M., PRUSINKIEWICZ P.: A few good lines: Suggestive drawing of 3d models. *Computer Graphics Forum 22*, 3 (2003), pp. 381–390. 2, 3

[Elb95] ELBER G.: Line illustrations ∈ computer graphics. *The Visual Computer 11*, 6 (1995), 290 – 296. 2, 3

[ES07] EISSEN K., STEUR R.: *Sketching. Drawing Techniques for Product Designers*. BIS Publishers, 2007. 3

[Gup97] GUPTILL A. L.: *Rendering in Pen and Ink*. Watson-Guptill, 1997. 3

[Hof00] HOFFMAN D.: *Visual Intelligence: How We Create What We See*. W. W. Norton & Company, 2000. 1, 6

[Huf71] HUFFMAN D. A.: Impossible objects as nonsense sentences. *Machine Intelligence 6* (1971), 295–323. 4

[JD09] JOHNSON G., DO E. Y.-L.: Games for sketch data collection. In *SBIM '09: Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling* (New York, NY, USA, 2009), ACM, pp. 117–123. 2

[LCLT08] LIU J., CAO L., LI Z., TANG X.: Plane-based optimization for 3d object reconstruction from single line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 2 (2008), 315–327. 4

[LF92] LECLERC Y. G., FISCHLER M. A.: An optimization-based approach to the interpretation of single line drawings as 3d wire frames. *International Jouranl on Computer Vision 9*, 2 (1992), 113–136. 4

[LMLH07] LEE Y., MARKOSIAN L., LEE S., HUGHES J. F.: Line drawings via abstracted shading. In *International Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2007), ACM, p. 18. 3

[LMLK99] LLADÓS J., MARTÍ E., LÓPEZ-KRAHE J.: A hough-based method for hatched pattern detection in maps and diagrams. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition* (Washington, DC, USA, 1999), ICDAR '99, IEEE Computer Society, pp. 479–. 6

[LS96] LIPSON H., SHPITALNI M.: Optimization-based reconstruction of a 3d object from a single freehand line drawing. *Computer-Aided Design 28*, 8 (1996), 651–683. 1, 4

[MH00] MYERS R., HANCOCK E. R.: Genetic algorithms for ambiguous labelling problems. *Pattern Recognition 33*, 4 (2000), 685 – 704. 4, 5, 6

[MKK98] MAMASSIAN P., KNILL D. C., KERSTEN D.: The perception of cast shadows. *Trends in Cognitive Sciences 2*, 8 (1998), 288–295. 3

[OSSJ09] OLSEN L., SAMAVATI F., SOUSA M., JORGE J.: Sketch-based modeling: A survey. *Computers & Graphics 33* (2009), 85–103. 1

[Pip07] PIPES A.: *Drawing for Designers*. Laurence King Publishing, 2007. 1

[PMC03] PIQUER A., MARTIN R. R., COMPANY P.: Using skewed mirror symmetry for optimization-based 3d line-drawing recognition. In *5th IAPR International Workshop on Graphics Recognition* (2003), pp. 182–193. 4

[RT02] ROS L., THOMAS F.: Overcoming superstrictness in line drawing interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 4 (April 2002), 456–466. 4

[SG80] SKLANSKY J., GONZALEZ V.: Fast polygonal approximation of digitized curves. *Pattern Recognition 12*, 5 (1980), 327 – 331. 6

[TBM09] THIERJUNG R., BRIELER F., MINAS M.: Online recognition of hatched and filled regions in hand-drawings. In *Proc. IUI Workshop on Sketch Recognition, Sanibel Island, Florida, February 8, 2009* (2009). 6

[VM01] VARLEY P. A., MARTIN R. R.: The junction catalogue for labelling line drawings of polyhedra with tetrahedral vertices. *International Journal of Shape Modeling 7*, 1 (2001), 23–44. 4

[Wal75] WALTZ D.: *The Psychology of Computer Vision*. McGraw-Hill, 1975, ch. 2: Understanding line drawings of scenes with shadows, pp. 19–91. 4

[ZISS04] ZANDER J., ISENBERG T., SCHLECHTWEG S., STROTHOTTE T.: High quality hatching. *Computer Graphics Forum 23*, 3 (2004), 421–430. 3