EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling
Ergun Akleman (Editor)

EXPRESSIVE 2015

# A Combined Junction-Cue Dictionary for Labelling Sketch Drawings with Artistic Shadows and Table-line Cues

A. Bonnici and K. P. Camilleri

Department of Systems and Control Engineering
Faculty of Engineering, University of Malta

**Abstract**

*The interpretation of user sketches generates research interest in the product design community since the computer interpretation of sketches may reduce the design-to-market time while giving the designer greater flexibility and control of the design process. This paper describes how cues, namely shadows and table lines used to express structural form in the drawing, may be used in a line-labelling algorithm to obtain a drawing interpretation that matches some design intent. To this extent, this paper describes canonical forms of the cues from which a combined junction and cue dictionary is created and used within a genetic algorithm framework to label the drawing. This paper also describes how such cues may be identified from the sketch.*

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

## 1. Introduction

Interpretation of 3D object representations from 2D sketches is not a trivial task since there can be an infinite number of 3D object geometries that project onto the 2D drawing [LS96]. While human observers easily ignore nonsensical interpretations [Hof00], such decisions are not trivially transferable onto a machine. Moreover, the interpretation of hand drawn sketches is made more complex because artists introduce additional embellishment artefacts to the sketch to make it more realistic [Pip07]. Trends in the interpretation of sketches have focused on sketch-based interfaces (SBIs) which instruct artists to draw in a specific manner, or adjust the sketch incrementally, reducing the potential of misinterpretation of the sketched strokes [OSSJ09]. While SBIs have interfaces which are less rigid than the interfaces associated with commercial computer aided design tools such as CATIA or AutoCAD among others, they are nevertheless different from the complete drawing freedom which designers enjoy when sketching freely. Thus it is desirable to investigate sketch interpretation algorithms that would allow for the interpretation of free-hand sketches. This however, is a research problem of considerable difficulty and for the scope of this paper, we will focus on the interpretation of sketches with embellishment artefacts that reflect the structural form of the object. Notably, in [BC13b] we note that practicing designers use shading and table-line cues to modulate the structural interpretation of the sketch, narrowing down the number of possible interpretations of the sketched object to one which reflects the designer's intent. In [BC13b], we show that these cues may be used as constraint filters within a line labelling framework reducing the geometry interpretation of the edge to a subset of all possible edge interpretations. While this generally results in a drawing interpretation that matches the design intent portrayed by the cues, this labelling algorithm has its limitations namely (a) the cue constraints are obtained by observing the way a cue modulates the edge it bears upon and thus constrains only that particular edge, whereas, as shown in Figure 1, the cue has an effect beyond that it directly bears upon; (b) the cues must be manually selected, labelled and associated with the edge they bear upon.

In order to address these issues, we first identify canonical forms of the cues which are used to build a combined junction-cue dictionary which extends the effect of the cue to all edges at the junction. We further use this canonical cue form to identify the cues that are present in the drawing. The rest of the paper is organised as follows: Section 2 presents the related work on line labelling algorithms, Section 4 presents a complete set of canonical shadow and table line cues, Section 5 shows how these canonical cues can



DOI: 10.2312/exp.20151185

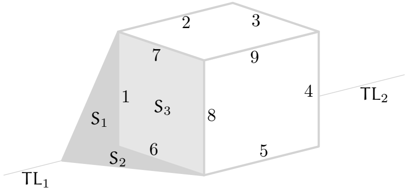



**Figure 1:** *Cues modulate the interpretation of neighbouring edges. Here, the table line $TL_1$ acts on edges 1 and 6, which are interpreted to be in physical contact with some background object or ground plane, enforcing the same interpretation for the neighbouring edges 2 and 5, extending the effect of the cue beyond the edges it bears upon. The same applies to the shadow cues $S_1$ and $S_2$ and the table line $TL_2$.*

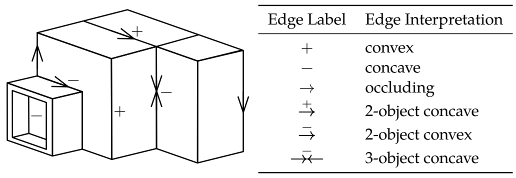



**Figure 2:** *A partially labelled drawing.*

be used to create a combined junction-cue dictionary which may be used within a genetic algorithm framework to label the drawings, Section 6 presents the results obtained while Section 7 concludes the paper.

## 2. Related Work

Line labelling algorithms are used to determine the geometry of an edge formed by the intersection of two planes $P_1$ and $P_2$ corresponding to distinct surfaces of an object. In single, trihedral objects, these are limited to *concave* edges, formed when the exterior angle between $P_1$ and $P_2$ is less than $\pi$; *convex* edges, formed when this is less than $\pi$; and *occluding* edges, formed when either one of $P_1$ and $P_2$ is not visible to the observer [Huf71, Clo71]. This labelling syntax has been extended to include a wider label vocabulary required by the different edge geometries associated with tetrahedral objects [VM01], objects with curved surfaces [Coo08b, Mal87] and scenes with illumination changes [Coo01, Wal75] among others. In particular, [Wal75] introduces three new edge labels, namely the *two-object concave*, *two-object convex* and *three-object concave* edge labels to distinguish between edges formed when two or three separable objects are in contact such that their representation in the drawing shares a common edge or plane as shown in Figure 2.

Drawings are typically labelled according to a pre-defined junction dictionary which consists of an exhaustive list of all possible legal edge labels associated with each junction geometry [Coo08a]. Labelling the drawing therefore consists of finding the set of consistent edge labels that result in a legal interpretation of the entire drawing. This can be achieved through Waltz filtering [Wal75] and other variants on Waltz filtering as discussed in [Mal87] and [Kir90] among others. Of particular interest is the optimisation approach used in [MH00] where a genetic algorithm is used to determine the optimal set of edge labels, using the junction dictionary as a measure of fitness of the edge interpretations and hence using the dictionary as soft constraints in contrast to its use as hard constraints on the edge labels in [Mal87,Coo08b,Kir90] among others. This has the advantage of identifying the best fitting labelling solution even if a completely legal one does not exist due to drawing ambiguities which arise from the rough nature of the user sketches.

Such an advantage is significant in the interpretation of drawings with cues since these are not necessarily drawn with accuracy. To this extent, the genetic algorithm approach was used in [BC13b] to label line drawings containing shadow and table line cues. Here, a second cue dictionary is created and this is used to constrain the possible interpretations of the edges which have cues bearing upon them. The fitness function is extended such that this measures the correspondence between the assigned labels and the constraints in both the junction and cue dictionaries. While this approach results in drawing interpretations that generally match the design intent as portrayed by the cues, the cue dictionary only constraints edges that have cues bearing upon them, whereas cues generally extend beyond this. Thus a better approach for the labelling of drawings with cues would be one which seeks to resolve the local effect of the cues, brought about by the use of the separate junction and cue dictionaries and fitness evaluation.

## 3. A complete set of canonical cue profiles

In order to use the cues with the line labelling algorithm to constrain the interpretation of an edge, it is necessary to map each instance of a cue onto some constrained interpretation, creating a cue-interpretation dictionary. Creating an exhaustive dictionary of all specific cues would be impractical due to the infinite possible cue shapes. However, the modulation of the edge interpretation is based primarily on the generic shape of the cue rather than on its specific shape, so much so that sketched cues are often drawn with a lower degree of accuracy in comparison to scenes created through computer graphics [Fou10]. Thus, the cue dictionary can categorise the cues according to their generic cue shape and an exhaustive list of such generic cue shapes becomes more tractable. To this extent, it is necessary to establish the light source positions that would cause the generic shadow shapes and table line cues. In [BC13b], we note that, assuming a single light source, there are seven possible light source placements which result in shading cues with different generic shape. We further note that there are ten different foreground-background relations that give rise to different

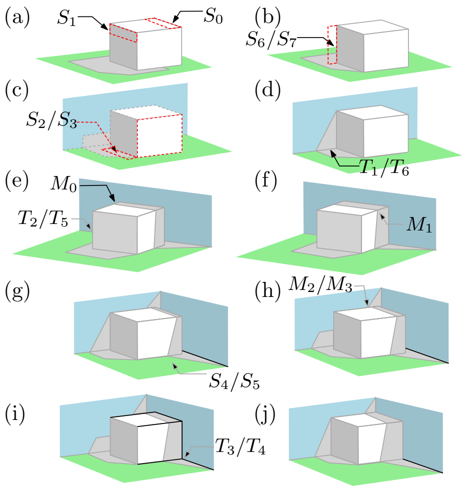



**Figure 3:** *Vertical and horizontal background planes may be placed in different positions resulting in different edge interpretations and cue combinations.*

cues corresponding to the different edge interpretations as shown in Figure 3. Thus to generate an exhaustive list of all shadow and table line cues, 3D primitives are placed under these ten foreground-background relations and light source placements, noting the different shadows and table line positions due to the different primitive geometries. These 3D primitives contain examples of all possible junction configurations and interpretations and are generated by dividing the 3D space into octants, treating different combinations of octants as foreground object [Wal75]. In the discussion that follows, we represent the shading and table-line cues that result from a single placed on the right of the object, although this can be trivially repeated for all other light source placements. Note that throughout the discussion, we reasonably assume that the drawing is drawn from a generic view point, that is, all edges and junction points are real and not due to some accidental alignment of the 3D object.

### 3.1. Cue profiles for shadows

The general form of the shadow sufficient for the purpose of distinguishing between different edge interpretations may be determined from the grey-level profile around the edge. Thus, we model shadow cues by the average grey-level of a rectangular strip along the edge as shown in Figure 3 and defined by:

$$S(d) = \frac{1}{L}\sum_{l=0}^{L-1} I(\mathbf{x}_d^l) \tag{1}$$

where $L$ is the width of the rectangular strip, $\mathbf{x}_d^l = \mathbf{x}_J + dR_\theta + lR_{\theta+\frac{\pi}{2}}$, $\mathbf{x}_J$ is the position of a reference junction point of which the edge is a member, $d$ is the displacement from the reference point along the edge, normalised such that $d = 0,\cdots 1$ where $d = 0$ corresponds to the position on the edge at the reference junction $\mathbf{x}_J$ while $d = 1$ corresponds to the other end of the edge, furthest away from $\mathbf{x}_J$, $R_\phi = [\cos\phi \sin\phi]'$ and $\theta$ is the orientation of the edge. Using this shadow profile model, we identify nine different canonical shapes for the shadow cues as detailed hereunder.

**Edge has no shadow cue** When an edge in the drawing has no shadow cue acting upon it, as in Figure 3(a), the shadow profile can be trivially described as $S_0(d) = 0$ for all $d = 0,\cdots,1$.

**Shadow cue along the edge** Edges that are completely enclosed in shadows, as in Figure 3(a), may be trivially described as $S_1(d) = g_s$ for all $d = 0,\cdots 1$ where $g_s$ is the grey-level corresponding to the shaded regions in the image.

**Cast shadows at concave edges** When two planes intersect to form a concave edge light rays falling on one of the planes may be partially blocked by the other, casting a shadow that tapers towards one of the junctions edges as shown in Figure 3(c). Depending on the reference junction selected, this shadow can have two canonical forms $S_2(d)$ and its reflection $S_3(d)$ as shown in Figure 4.

**Cast shadows at occluding edges** Depending on the geometry and spatial arrangement of objects in the scene, cast shadows at occluding edges may have different canonical forms, resulting in shadow profiles $S_4(d)$ - $S_9(d)$ as shown in Figure 4, examples of which can be observed in Figures 3(b) and (g).

As shown in Figure 3, shadows may occur on both side of an edge. We choose to model both shadows independently, while introducing a parameter $M_q$, $q = 0,\cdots,3$ to model the relation between inflection points $M_{left}$ and $M_{right}$ on the left and right hand side of the edge respectively. Thus, $q = 0$ when none or only one of the shadows on either side of the edge have an inflection point, for example Figure 3(e); $q = 1$ when $|M_{left} - M_{right}| \leq t$, for example in Figure 3(f); $q = 2$ when $M_{left} > M_{right} + t$; and $q = 3$ when $M_{left} < M_{right} - t$ as shown in Figure 3(h), where $t$ a threshold on the inflection point position, required to allow for some flexibility due to the rough nature of the sketch and sketched shadows.

### 3.2. Modelling the table lines

Table lines may be completely described by their position on the edge with respect to the reference junction point. We use two positional descriptors, namely the location of the table line along the length of the edge as well as the spatial location of the table line with respect to the edge. Denote by $\mathbf{x}_T$ the intersection point between the table line and the edge. Then, for edges of length $D$, the table line location on the edge may be categorised as (a) emerging from the reference junction if $\mathbf{x}_T - \mathbf{x}_J \leq Dt_{t_1}$, example in Figure 3(i); (b) emerging from the middle of the edge if $Dt_{t_1} < \mathbf{x}_T - \mathbf{x}_J \leq Dt_{t_2}$, for

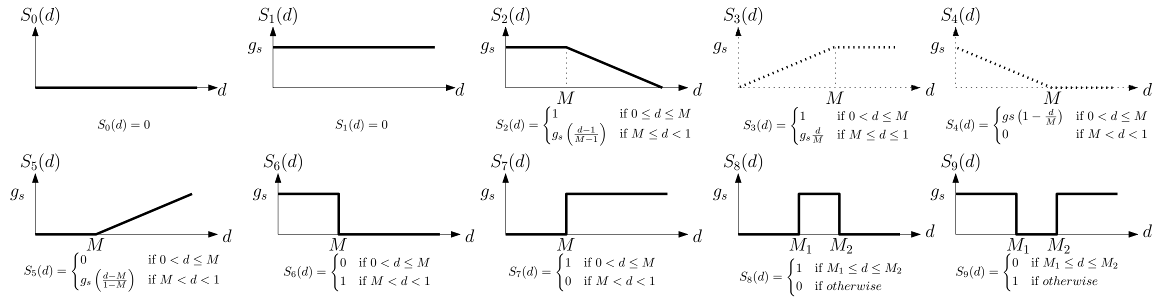



**Figure 4:** *A representation of the shadow profiles expected at drawing junctions.*

example Figure 3(e); and (c) emerging from the other junction point if $\mathbf{x}_T - \mathbf{x}_J \geq Dt_{t_2}$, where $t_{t_1}$ and $t_{t_2}$ are two thresholds whose value is in the range $[0,1]$ and serve to introduce a tolerance value to the table line location to compensate for the rough nature of the drawing. The spatial location of the table line with respect to the edge may be sufficiently described as being on the left or right hand side of the edge. Thus, the table lines may be represented by a parameter $T_r$, $r = 0, \cdots 6$ where $T_0$ represents an edge with no table line cue, while $T_r, r = 1 \cdots 6$, represents the six combinations of table line positions.

### 3.3. Symbolic representation of the cues at an edge

The cues present at an edge may therefore be represented by the tuple $(p_{right}, p_{left}, q, r)$, where $p_{right}$ and $p_{left}$ represent the shadow profile model on the right and left hand side of the edge. We define the set $c$ as the set consisting of the $Q$ possible combinations of the cue tuple, that is, $c = \{(p_{right}, p_{left}, q, r)_1, \cdots, (p_{right}, p_{left}, q, r)_Q\}$.

### 3.4. Identifying the cues from the sketched drawing

The drawing may be preprocessed using a vectorisation algorithm such as [BC13a] which identifies the drawing edges from the shadow strokes and table-line strokes and organises these edges into their respective junctions. For each edge $e_n$, the shadow profiles may be determined from the drawing as described in Section 3.1 while the table line location may be determined from the intersection of the table line with the drawing edges. Thus, a cue representation $\hat{S}_{right}, \hat{S}_{left}, \hat{M}, \hat{T}$ is obtained from the drawing such that the cue tuple associated with the edge $e_n$ may be obtained from:

$$p_{right,left} = \arg\left\{\min_{p=0,\cdots,9}\left\{S_p - \hat{S}_{right,left}\right\}\right\} \quad (2)$$

$$q = \arg\left\{\min_{q=0,\cdots,3}\left\{M_q - \hat{M}\right\}\right\} \quad (3)$$

$$r = \arg\left\{\min_{q=0,\cdots,6}\left\{T_r - \hat{T}\right\}\right\} \quad (4)$$

## 4. A combined junction-cue dictionary

To create the combined junction-cue dictionary, drawing primitives containing examples of all junction geometries are observed under different foreground-background relations and light source positions as explained in Section 3. Trihedral drawings consist of four distinct junction geometries, namely `W`, `Y`, `T`, `L`, which can be rotated and combined to form the complete set of junction geometries [VM01, Coo08a]. We represent these geometries by an indexed set $I_G = \{1, \cdots, K_f\}$, where $K_f$ is the total number of different junctions, such that the specific junction geometry may be identified by $g \in I_G$. Each specific junction geometry is formed by $K_g$ edges such that edges at the junction can be indexed by the index set $I_j = \{1, \cdots, K_g\}$. For consistency, $I_j$ is an ordered set and is ordered such that the edges are listed in a clockwise manner, starting from an orientation of $0°$ with the horizontal axis.

The specific junction geometry, when observed under different foreground-background relations will have $K_c^g$ unique cues indexed by the index set $I = \{1, \cdots K_c^g\}$. Each of these $i \in I$ cues is comprised of the cue tuples at the individual edges forming the junction such that the cues at the junction may be defined by $C_g(i) = \{\kappa_j\}_{j \in I_j}$, where $\kappa_j \in c$ as shown in Figure 5. The junction-cue dictionary can therefore be defined as the indexed family of sets $\Gamma = \{\{\gamma_{g,i}\}_{i \in I}\}_{g \in I_G}$, where $\gamma_{g,i}$ is the set of all possible edge label interpretations at a junction with geometry $g$ given the cue indexed by $i$.

### 4.1. Constraining the edge interpretations

In the absence of any geometry or cue constraints, an edge $e_n$, $n = 1, \cdots N$ may be labelled with the edge label $\lambda_n \in \Lambda$ where $\Lambda = \{+, -, \overset{-}{\rightarrow}, \overset{-}{\leftarrow}, \overset{+}{\rightarrow}, \overset{+}{\leftarrow}, \overset{+}{\rightarrow\!\!\leftarrow}\}$ is the full set of edge labels associated with edges that form a trihedral object. Specifically, an edge may be labelled with the edge label $\omega_{\lambda_n}$, that is $\lambda_n = \omega_{\lambda_n} \in \Lambda$. Through the vectorisation preprocessing step, the drawing is arranged into $k = 1, \cdots, K$ junctions, each having a junction geometry label $g_k \in I_G$. Specifically, a junction will have the junction geometry label $\omega_{g_k}$, that is, $g_k = \omega_{g_k} \in I_G$. The unconstrained edge labels at the junction can therefore be expressed as $\{\lambda_n\}_{n \in I_j}$

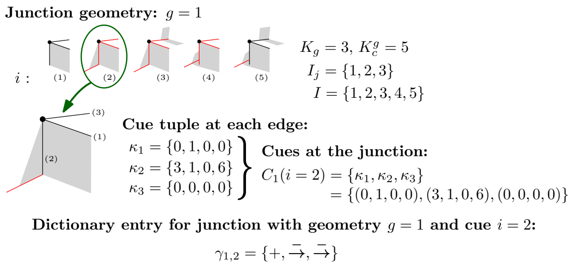



**Figure 5:** *An example of a dictionary entry for a W junction. For clarity, the red line denotes the table line cue.*

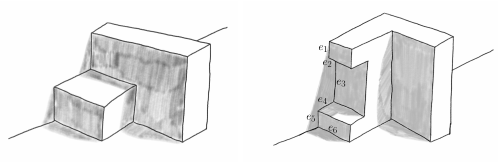
and specifically, $\{\omega_{\lambda_n}\}_{n \in I_j}$. A drawing edge may have a cue $\phi_n \in c$ acting upon it, with the specific cue $\omega_{\phi_n}$, that is, $\phi_n = \omega_{\phi_n} \in c$. The cues at the junction may therefore be represented by $\beta_j = \{\phi_n\}_{n \in I_j}$ and specifically, $\omega_{\beta_j} = \{\omega_{\phi_n}\}_{n \in I_j}$. These cues can be used to constrain the interpretation of the edge such that $\{\lambda_k\}_{k \in I_j} = \{\{\gamma_{\rho_j, \nu_j}\}_{\nu_j \in I}\}_{\rho_j \in I_G}$, where $\{\nu_j | \beta_j = C_{\rho_j}(\nu_j)\}$

## 5. Line labelling using a genetic algorithm

To cast the labelling problem under the genetic algorithm framework, we need to define the chromosome as well as the fitness mechanism to allow the genetic algorithm to select the most suitable edge labels $\lambda_n$ for each edge in the drawing. We represent the drawing edges by the chromosome $E$, consisting of $N$ genes, where $N$ is the number of edges in the drawing. Each of the genes represents an edge label $\lambda_n$ such that the chromosome is defined by $E = \{\lambda_1, \cdots, \lambda_N\}$, with the particular chromosome of the population being $E_i = \{\lambda_1^i = \omega_{\lambda_1^i}, \cdots \lambda_1^N = \omega_{\lambda_i^N} | \omega_{\lambda_n^i} \in \Lambda\}$. The goal of the genetic algorithm is to evolve the population of chromosomes such that the genes of the chromosome are the best fitting interpretations of the edges. The genetic algorithm mechanism of cross-over and mutation allow for the genetic algorithm to generate new chromosomes, exploring the search-space, while selecting the fittest of the chromosomes to form the evolving population [ES03]. To this extent, a suitable fitness function that allows the selection of chromosomes whose genes satisfy the junction and cue constraints present in the drawing must be selected.

### 5.1. The fitness function

Let $A = \{\{\gamma_{\rho_j, \nu_j}\}_{\nu_j \in I}\}_{\rho_j \in I_G}$ be the set of edge labels associated with the junction geometry $\rho_j$ and cues $\nu_j$ as determined by the junction-cue dictionary, and $B_j^i = \{\lambda_k^i\}_{k \in I_j}$ be the edge labels obtained from the chromosome $E_i$ for junction $j$. If the chromosome is to satisfy the geometry and cue constraints, then $B_j^i \in A$ such that the fitness of the chromosome may be determined by the sum of the differences between the edge labels assigned to the chromosomes and



(a) Drawing 1 (b) Drawing 2

**Figure 6:** *Two drawings with localised cues*

those specified by the junction-cue dictionary. Thus, the fitness function can be defined as:

$$F(E_i) = \frac{1}{2N} \sum_{j=1}^{J} \min_{m=1,\cdots|A_j|} \mathrm{H}(A(m), B_j^i) \tag{5}$$

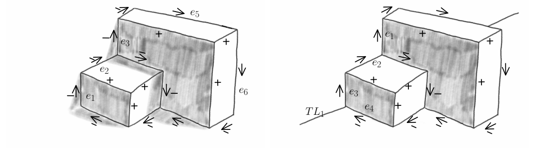
where $\mathrm{H}(A(m), B_j^i)$ is the Hamming distance between $A(m)$ and $B_j^i$.

## 6. Results

In order to determine the advantage that the combined junction-cue dictionary has over using two separate cue and junction dictionaries to perform the line labelling task, the labelling algorithm was applied to drawings used in [BC13b], namely, those shown in Figure 6. These drawings were chosen specifically because the performance of the cue-constrained line labelling algorithm (*cGA*) described in [BC13b] was sub-optimal, with the *cGA* reaching the desired solution in 90% and 42% of the 50 trials over which the *cGA* was performed.

For comparison purposes, in our implementation, the genetic algorithm was implemented using the same evolution mechanisms used in [BC13b], that is, a mutation rate of 0.03 and a cross-over rate of 0.9, using the half-uniform crossover method, while stochastic universal sampling was used to provide the selection mechanism. The population size used was that of a 100 chromosomes while the population was allowed to evolve for at most 80 generations.

In both cases, the shadows and table lines which were originally used to constrain only the edges they bear upon, result in under-constrained edges such that while reducing the number of plausible interpretations to a smaller subset of all possible interpretations, the *cGA* was not able to determine the intended interpretation, as defined by the cues present in the drawing, in all of the trials. In contrast, using the combined junction-cue dictionary, the line labelling algorithm now attains the intended interpretation in all trials. This is particularly relevant in Drawing 2, which had a low performance under the *cGA*. In this drawing, the shadows that bear upon edges $e_1$, $e_2$, $e_3$ and $e_4$ are alone, insufficient to constrain the interpretation of these edges. However, when



(a) Drawing 3 (b) Drawing 4

**Figure 7:** *Two drawings with missing cues*

the shadow cues are considered in conjunction with the table line cue acting on $e_5$ and $e_6$, as well as the junction geometry, the edge $e_4$ can now be constrained to a single interpretation, and this in turn places constrains on the interpretations of edges $e_1$, $e_2$ and $e_3$, hence the effect of the table line is propagated onto all edges it affects. This is desired because the effect of the cues are in general, not localised to single edges.

The algorithm was also evaluated on two versions of Drawing 2 which have missing cues as shown in Figure 7. Drawing 3 may be interpreted as having no background plane albeit with badly sketched shadows, or as a drawing with a background plane and correctly drawn shadows but with missing table-line cues. On the other hand, Drawing 4 has the table line representation but the drawing has missing shadows such that the shadow cues are inconsistent with the light source placement. As expected, in both cases, the population does not reach the optimal fitness value of one since the cues present in the drawings are an incomplete match to those in the dictionary. In both drawings, the performance of the genetic algorithm is consistent and the solutions reached are illustrated in Figure 7. In the case of Drawing 3, the shadow cue at $e_3$ closely matches with the shadow profile $S_3(d)$ such that the genetic algorithm favours the interpretation of the missing table line cue, anchoring the interpretation of edges $e_1, e_2$ and $e_3$ to that of a separable concave edge interpretation and a fitness value of 0.98, the highest fitness value possible for this interpretation, was obtained in all trials. This drawing highlights a further difference between the combined junction-cue dictionary and the separate dictionaries, namely the additional context provided by the junction geometry which allows for the inclusion of edge interpretations in the absence of any cues in the dictionary. The effect of this may be observed with the interpretation of edges $e_5$ and $e_6$ in Drawing 3 which, given the junction geometry and light source location, should not have any shadow cue acting upon them. The lack of shadow cues in the drawing therefore supports the interpretation that these edges are occluding edges. In the individual dictionaries, edges with no cues bearing upon them do not have the required context which allows us to include such interpretations in the dictionary and without such constraints, the interpretation of these edges falls back onto the junction dictionary alone. This would support the $\rightarrow$ and $\overset{-}{\rightarrow}$ interpretations equally. In fact, the *cGA* is reported to support the occluding edge interpretation in only 48% of the trials, in contrast with the 100% support this interpretation achieves with the combined junction-cue dictionary. Thus, as suggested in [Coo01], the combined junction-cue dictionary allows the absence of cues to be included as a cue.

Drawing 4 exhibits a degree of inconsistency between the shadow cues and the table-line cues since the absence of shading cues in the upper part of the object support an interpretation of an object that is not touching any background while the table-line acting upon edges $e_4$ and $e_3$ implies otherwise. Using the combined junction-cue dictionary, the genetic algorithm again obtained the maximum fitness value possible for the drawing, namely that of 0.80 and the interpretation obtained by the genetic algorithm is consistent over all trails, and corresponds to the interpretation supported by the upper part of the object. Although this differs from the interpretation preferred by the *cGA*, the interpretation obtained by the combined junction-cue dictionary is expected since the lack of shadows at edges $e_1, e_2$ and $e_3$ strongly support the interpretation that the object is not touching against any background. Although the table-line at edge $e_4$ supports the interpretation that the object should indeed be touching the background, this is the only cue supporting this interpretation such that this interpretation has an overall lower fitness that the alternative interpretation that the object is not touching the background. Note that in this case, the shadow cues that are present in the drawing constrain the interior edges at the two `Y` junctions to be convex edges but do not provide further support for the edge labels at the exterior edges of the object.

## 7. Conclusion

In this paper we describe a canonical representation of shadow and table line cues that may be present in the drawing. We show that these cues may be used in a combined junction-cue dictionary which improves the performance of the line labelling algorithm. The combined junction-cue dictionary offers the necessary context to allow for absent cues, due to particular geometry and foreground-background relations to be treated as cues, which was not possible with separate junction and cue dictionaries.

The results obtained in Drawing 4 lead to an interesting observation. Despite the cue inconsistency, the table line $TL_1$ seems to lend more support to the interpretation which has the object touching some background plane rather than the alternative interpretation that there is no such background plane, notwithstanding the fact that there is more evidence to support the alternative interpretation that the object is not touching the background. This leads to an interesting research question on whether human attend to cues in a different manner and whether this can be encoded within the dictionary.

## References


[BC13a] BONNICI A., CAMILLERI K.: A circle-based vectorization algorithm for drawings with shadows. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modelling* (2013), pp. 69–77. 4

[BC13b] BONNICI A., CAMILLERI K.: A constrained genetic algorithm for line labelling of line drawings with shadows and table-lines. *Computers & Graphics 37*, 5 (2013), 302 – 315. 1, 2, 5

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

[Coo01] COOPER M.: The interpretation of line drawings with contrast failure and shadows. *International Journal on Computer Vision 43*, 2 (2001), 75–97. 2, 6

[Coo08a] COOPER M.: *Line Drawing Interpretation*. Springer-Verlag, 2008. 2, 4

[Coo08b] COOPER M.: A rich discrete labeling scheme for line drawings of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 4 (April 2008), 741–745. 2

[ES03] EIBEN A. E., SMITH J. E.: *Introduction to Evolutionary Computing*. Springer, 2003. 5

[Fou10] FOURQUET E.: Learning about shadows from artists. In *Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging* (2010), Jepp P., Deussen O., (Eds.), pp. 107–114. 2

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

[Huf71] HUFFMAN D. A.: Impossible objects as nonsense sentences. *Machine Intelligence 6* (1971), 295–323. 2

[Kir90] KIROUSIS L. M.: Effectively labeling planar projections of polyhedra. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 2 (1990), 123–130. 2

[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

[Mal87] MALIK J.: Interpreting line drawings of curved objects. *Journal of Computer Vision 1* (1987), 73–103. 2

[MH00] MYERS R., HANCOCK E. R.: Genetic algorithms for ambiguous labelling problems. *Pattern Recognition 33*, 4 (2000), 685 – 704. 2

[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

[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. 2, 4

[Wal75] WALTZ D.: *The Psychology of Computer Vision*. McGraw-Hill, 1975, ch. 2: Understanding line drawings of scenes with shadows, pp. 19–91. 2, 3