Artificial Intelligence for Engineering Design, Analysis and Manufacturing

cambridge.org/aie

Sketch-based interaction and modeling: where do we stand?

Alexandra Bonnici[1], Alican Akman[2], Gabriel Calleja[1], Kenneth P. Camilleri[1], Patrick Fehling[3], Alfredo Ferreira[4], Florian Hermuth[3], Johann Habakuk Israel[3], Tom Landwehr[3], Juncheng Liu[5], Natasha M. J. Padfield[1], T. Metin Sezgin[2] and Paul L. Rosin[6]

[1]Faculty of Engineering, University of Malta, Room 418, Msida MSD2080, Europe; [2]Koç University, Istanbul, Turkey; [3]Hochschule für Technik und Wirtschaft, Berlin, Germany; [4]Instituto Superior Técnico, Lisbon, Portugal; [5]Peking University, Beijing, China and [6]Cardiff University, Cardiff, Wales, UK

Review Article

Cite this article: Bonnici A *et al* (2019). Sketch-based interaction and modeling: where do we stand? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 1–19. https://doi.org/10.1017/S0890060419000349

Received: 30 November 2018
Revised: 7 August 2019
Accepted: 25 October 2019

Key words:
Sketch-based interfaces; sketch interactions; modeling; augmented and virtual reality; sketch-based queries; non-photorealistic rendering

Author for correspondence:
Alexandra Bonnici, E-mail: alexandra.bonnici@um.edu.mt

Abstract

Sketching is a natural and intuitive communication tool used for expressing concepts or ideas which are difficult to communicate through text or speech alone. Sketching is therefore used for a variety of purposes, from the expression of ideas on two-dimensional (2D) physical media, to object creation, manipulation, or deformation in three-dimensional (3D) immersive environments. This variety in sketching activities brings about a range of technologies which, while having similar scope, namely that of recording and interpreting the sketch gesture to effect some interaction, adopt different interpretation approaches according to the environment in which the sketch is drawn. In fields such as product design, sketches are drawn at various stages of the design process, and therefore, designers would benefit from sketch interpretation technologies which support these differing interactions. However, research typically focuses on one aspect of sketch interpretation and modeling such that literature on available technologies is fragmented and dispersed. In this paper, we bring together the relevant literature describing technologies which can support the product design industry, namely technologies which support the interpretation of sketches drawn on 2D media, sketch-based search interactions, as well as sketch gestures drawn in 3D media. This paper, therefore, gives a holistic view of the algorithmic support that can be provided in the design process. In so doing, we highlight the research gaps and future research directions required to provide full sketch-based interaction support.

Introduction

Sketching is a natural and intuitive means of communication for expressing a concept or an idea. A sketch may serve several purposes: it can be used as a support tool for problem-solving, it might record something that a person sees, it can be a way of storytelling as a part of human interaction, or it can be used for developing ideas at any stage of a design process. Thus, sketching is seen as a method for creativity and problem-solving and is considered to be central to the design-oriented disciplines of architecture, engineering, and visual communication (Craft and Cairns, 2009). In order to assess the effectiveness of sketching, researchers conducted experiments using applications in areas, such as industrial design (Schütze *et al*., 2003; Lugt, 2002), route planning (Heiser *et al*., 2004), and interface design (Landay and Myers, 1995). These studies demonstrate that participants who are allowed to sketch freely during the experiment are able to design products which have better functionality while experiencing fewer difficulties in the design process (Schütze *et al*., 2003). Moreover, when compared to note taking, the act of sketching offers better support to individual re-interpretative cycles of idea generation and enhances individual and group access to earlier ideas (Lugt, 2002). In route design, sketching is shown to enhance collaboration, with users who are allowed to use sketches designing more efficient routes in less time, demonstrating the ability of sketches to focus attention and provide easy communication between groups (Heiser *et al*., 2004). Moreover, in interface design, sketching has been shown to aid the evaluation and formation of ideas, allowing designers to focus on the larger conceptual issues rather than trivial issues such as fonts or alignment of objects (Landay and Myers, 1995).

The intuitive and communicative nature of sketches has brought the act of sketching to the attention of human–computer interface designers who focus on developing intuitive interfaces. Sketch-based interfaces have the potential to combine the processing power of computers with the benefits of the creative and unrestricted nature of sketches. However, realizing this potential requires combining efforts from several research areas, including computer graphics, machine learning, and sketch recognition. Sketch recognition has many challenges that arise from the computational difficulties of processing the output of the highly individual and personal task of sketching, requiring algorithms that can overcome the ambiguity and variability of the sketch. An effective sketch recognition method should be able to recognize freehand drawings, created on any surface and with any material. Achieving high recognition rates that meet these constraints remains a challenge.

This paper builds on the discussions which took place during a three-day workshop held at the University of Malta in March 2018. This workshop brought together researchers with experience in sketch interpretation, sketch-based retrieval, sketch interactions in virtual and augmented reality (VR/AR) interfaces, and non-photorealistic rendering (NPR). Although these research areas are often discussed independently, they complement each other: natural interfaces in augmented reality systems allow for the exploration of concepts in collaborative design. Interpretation of drawings made in physical or digital ink also focuses on natural interfaces, albeit in two-dimensional (2D) digital or paper drawings. Thus, seeking common grounds between these two research areas allow for the integration of different sketching modalities in conjunction with augmented reality rendering. To such a synergy, research in three-dimensional (3D) object search makes it possible to integrate library searching of object parts within the augmented reality environment. The object simplification through NPR is then particularly useful in this context to search for specific parts. This paper presents the position taken by the participants after this workshop. In the paper, we take a broad view and look into the interpretation problem in diverse contexts, for example, in the context of 3D modeling, sketch-based retrieval, multimodal interaction, VR and AR interfaces. Conclusions reached in this paper are a result of the discussions which took place during the workshop and reflect not only the authors' opinions but also the insights brought into the workshop by practicing manufacturing engineers and architectural designers who were also present at the workshop.

The rest of the paper is divided as follows: the section "State of the art in sketch interpretation and modeling" provides a review of the state of the art in sketch interpretation and sketch-based modeling algorithms, the section "Future directions" discusses open challenges and future directions that should be addressed to improve the practicality of these systems, while the section "Conclusion" concludes the paper.

State of the art in sketch interpretation and modeling

Machine interpretation of drawings dates back to as early as the 1960s with the development of algorithms able to interpret blueprints and cadastral maps to automate the digitization process of such drawings (Ablameyko and Pridmore, 2000), branching quickly into the interpretation of drawings as 3D objects (Clowes, 1971; Huffman, 1971). Research in sketch interpretation remains active through attempts to relax drawing constraints as well as the development of different technologies which changed the way people draw. Figure 1 illustrates the different sketch interactions which will be discussed in this paper. The section "Interpretation of offline sketches" describes the processing steps to obtain 3D models from paper-based sketches. The section "Interactive sketches" describes the interactive interfaces which require digital sketching. The section "Sketch-based shape retrieval" describes sketch-based retrieval approaches which can compare the sketch directly to some 3D object database or to a sketched rendering of the object. Finally, the section "Beyond the single-user, single-sketch applications" describes sketching interactions in VR and AR which can be used to either manipulate a premade 3D object or create a fresh 3D object within the VR/AR environment.

Interpretation of offline sketches

In its most primitive form, a sketch captures fleeting ideas (Eissen and Steur, 2007). The sketch may, therefore, be incomplete and inaccurate, but the ability to explain abstract concepts through drawings makes the sketch a powerful means of communication (Olsen *et al.*, 2008). Notwithstanding the strengths of pen-and-paper sketching, the sketch serves only as an initial working document. Once a concept is sufficiently developed, initial sketches are redrawn using computer-aided design (CAD) tools to obtain blueprints for prototyping (Cook and Agah, 2009) or to benefit from VR or AR interactions with the product. Despite the effectiveness and ability of CAD tools to handle complex objects, these tools have a steep learning curve for novice users and even experienced designers spend a considerable amount of time and energy using these CAD tools. Ideally, the conversion from paper-based sketches to a working CAD model is achieved without requiring any redrawing of the sketch. The machine interpretation of paper-based drawings may be loosely divided into three steps, namely distinguishing ink marks from the background through binarization; representing the ink strokes in vector form; and obtaining shape information from the drawing to change the flat drawing into a 3D working model.

Image binarization

Off-the-shelf binarization algorithms, such as Otsu's or Chow and Kaneko's algorithms (Szeliski, 2010), provide a suitable foreground to background separation when drawings are drawn on plain paper and scanned. However, problems arise with the introduction of textured paper, such as ruled or graph paper, bleed-through from previous drawings, as illustrated in Figure 2a and even variable illumination, as illustrated in Figure 2b. Thus, binarization algorithms need to be robust to these gray-level artifacts, leading to more robust binarization algorithms such as Lins *et al.* (2017), among others.

Vectorization

Once the ink strokes are distinguished from the image foreground, vectorization is applied to allow the ink strokes to be redrawn under the CAD environment (Tombre *et al.*, 2000). The focus here lies in the accurate representation of the topology of the ink strokes, paying particular attention to preserve an accurate representation of junction points (Katz and Pizer, 2004). Skeletonization algorithms, which remove pixels contributing to the width of the ink strokes while retaining the pixels which contribute to the medial-axis of strokes, are a natural first step toward vectorization (Tombre *et al.*, 2000). However, skeletonization produces spurious line segments, especially if the ink strokes are not smooth. Thus, skeletonization algorithms rely heavily on beautification and line fitting of the skeletal lines (Hilaire and Tombre, 2006). Alternatively, rather than attempt to correct the spurs created through skeletonization, the medial-axis may be obtained through matching pairs of opposite contours (Ramel *et al.*, 1998) or horizontal and vertical run lengths (Keysers and Breuel, 2006). All of these algorithms require visiting each pixel in the image to determine whether it forms part of the medial-axis. Line strokes can, however, be approximated as

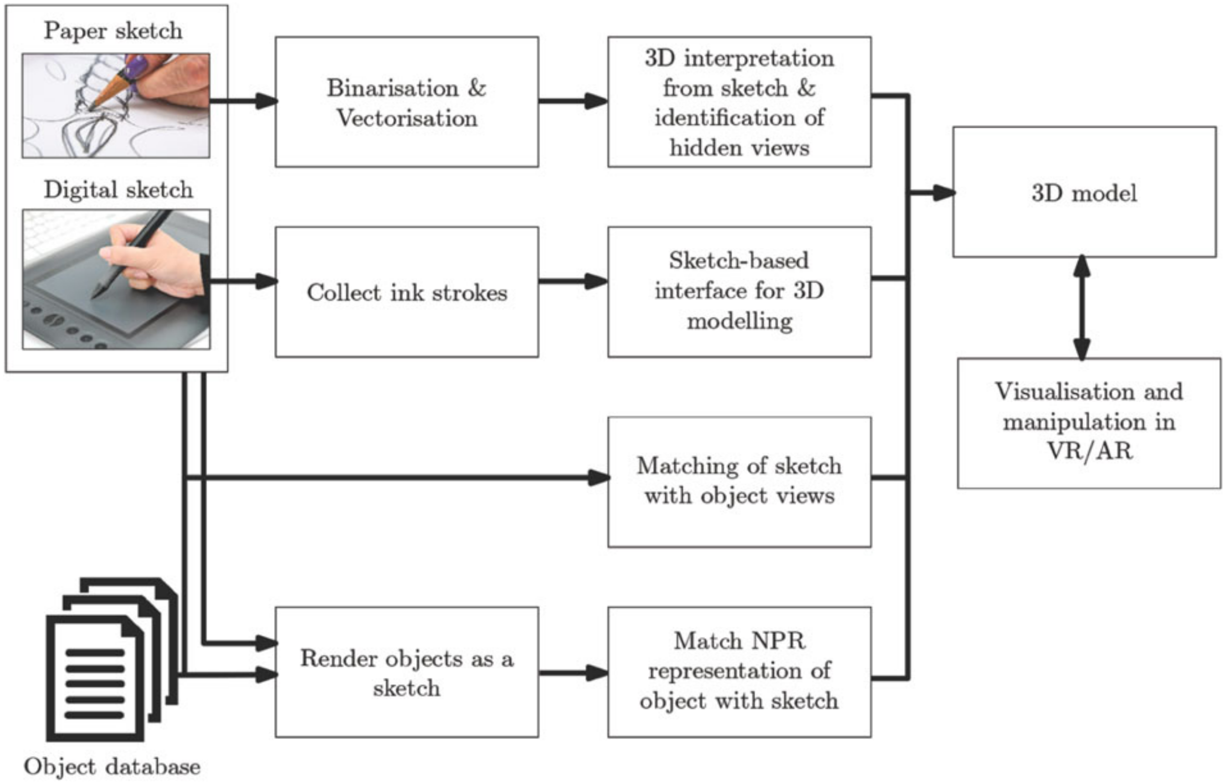

Fig. 1. An overview of the different sketch interaction modes discussed in this paper.

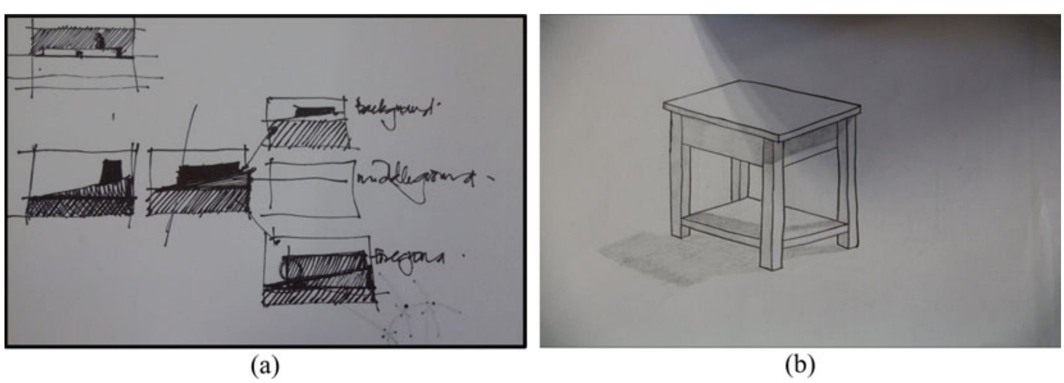

Fig. 2. (a) A pen-based sketch with bleed-through (drawing provided by Stephen C. Spiteri). (b) A pencil sketch showing variable illumination.

piecewise linear segments, and thus, it is possible to reduce the computational costs for locating the medial-axis by adopting a sampling approach (Dori and Wenyin, 1999; Song *et al.*, 2002). These sampling approaches then rely on heuristics to propagate the sampler through the stroke and attempt to propagate the line for its entirety, beyond the junction point.

Junction points, however, have an essential role in the interpretation of the drawing, and thus, if the vectorization does not find the junction locations directly, these are often estimated from the intersection points of lines (Ramel *et al.*, 1998). This approach, while suitable for neat, machine-generated line drawings, is not suitable for human sketches which are typically drawn sloppily with poorly located junctions (Ros and Thomas, 2002) as illustrated in Figure 3. Moreover, these algorithms typically assume that the drawings consist predominantly of straight lines and circular arcs. Problems arise when this assumption is relaxed to include a larger variety of smooth curves, which allows for drawings with more natural surfaces, as illustrated in Figure 4. Recent vectorization algorithms shifted the focus from the location of lines to the localization of junction points, borrowing from computer vision approaches of finding corners in natural images but adapting this to sketched drawings. Notably, Chen *et al.* (2015) use a polar curve to determine the number of branches at a potential junction point, hence establishing the junction order as well as locating the junction position. Noris *et al.* (2013), Pham *et al.* (2014), Favreau *et al.* (2016), and Bessmeltsev and Solomon (2019) characterize the topology of junctions typically found in sketches, describing the different possible points of contact

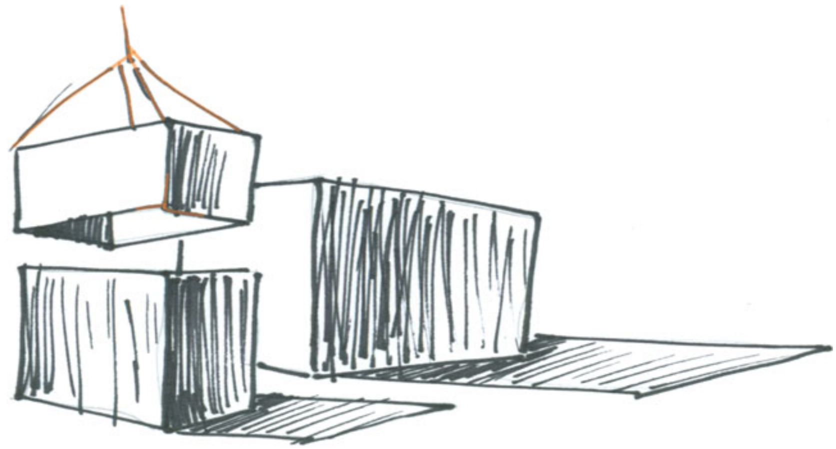

Fig. 3. Lines do not necessarily intersect accurately at a junction point.

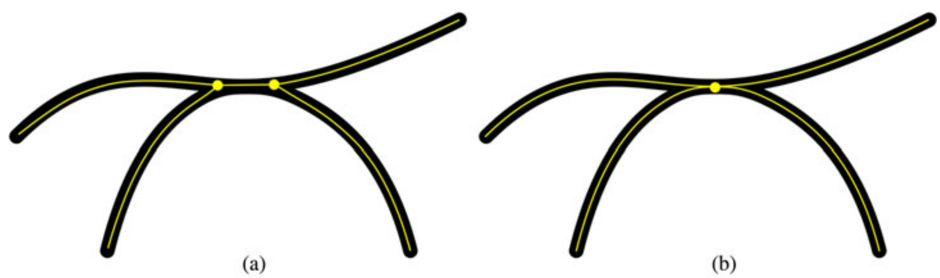

Fig. 4. The two smooth curves are badly represented by two junction points in (a) rather than the single tangential point of intersection as in (b).

between the central lines of two strokes at every junction, while Bonnici *et al.* (2018) use Gabor-like filters to first roughly localize junctions, and then refine the junction position and topology by focusing only on the image area around the junction.

Table 1 provides an overview of the vectorization algorithms discussed and their performance. We note that the evaluation of vectorization algorithms does not follow a standardized procedure, and there are four metrics described in the literature. The edit cost index (ECI) which is the weighted average of the number of false alarms, the number of misses, the number of fragmented ground-truth vectors, and the number of individual ground-truth vectors grouped into a single vector (Chhabra and Phillips, 1998). This measure, therefore, assesses the number of corrections to the vector data that the user must perform to correct any algorithmic errors. Dori and Wenyin (1999) introduce the vector recovery index (VRI) which gives a measure of the similarity between the extracted vectors and the ground-truth vectors, taking into account the overlap between vectors, fragmentation and merging of vectors. Noris *et al.* (2013) later introduce two alternative metrics. The centerline error (CE) measures the shift in the detected centerline position from ground-truth centerline as a ratio of the line thickness, while the salient point error (SPE) measures the distance between the detected junction points and the ground-truth junction points as a percentage of the line width.

Moving away from traditional preprocessing required for vectorization techniques, Simo-Serra *et al.* (2016) employ a fully convolutional neural network (CNN) to change a rough sketch into a simplified, single-line drawing. The CNN is designed to have three parts. First, an encoded spatially compresses the image. The second step then extracts the essential lines from the image, while the third step acts as a decoder and converts the simplified representation into a gray scale image of the same resolution as the input. The CNN is trained using pairs of rough and simplified sketches, with the rough sketches being generated through an inverse reconstruction process, that is, given a simplified sketched drawing, the artist is asked to make a rough version of the sketch by drawing over it. Data augmentation is then used to introduce tonal variety, image blur and noise, thus increasing the drawing pairs. Similarly, Li *et al.* (2017) apply CNNs to extract structural lines from manga images. Here, a deeper network structure is used to overcome the patterned regions of the manga. The use of CNNs can, therefore, provide the necessary preprocessing to allow for the extraction of line vectors from rough sketches exhibiting over-sketching, as well as textures typical of shading strokes.

Interpretation

Once vectorized, the sketch can be rewritten in a format which is compatible with CAD-based software such as 3DMax[1] among many others. These drawings remain, however, flat 2D drawings and obtaining the desired sketch-to-3D interpretation requires further drawing interpretation. The problem of assigning depth to a drawing is not a trivial task due to the inherent ambiguity in the drawing (Lipson and Shpitalni, 2007; Liu *et al.*, 2011). Edge labeling algorithms, such as those described in Clowes (1971), Huffman (1971), Waltz (1975), and Cooper (2008) among others, determine the general geometry of the edge, that is, whether an edge is concave, convex, or occluding. These algorithms define a junction as the intersection of three or four edges, creating a catalog of all possible junction geometries. The catalog of junctions is used as a look-up table to recover the 3D structure from the drawing. Although this approach is effective, its main

[1]https://www.autodesk.eu/products/3ds-max/overview

Table 1. Comparison of vectorization algorithms listing a brief description of the algorithms and their performance

Algorithm	Brief description	Performance measure
Ramel *et al.* (1998)	Straight line, quadrilateral and Bezier curve fitting on edge contours of drawings. Suitable for technical and engineering drawings	Visual inspection using the author's own image database
Dori and Wenyin (1999)	Propagates a rectangular segment along lines in a binary image to extract line and circular arc fragments. Suitable for technical and engineering drawings	Average *VRI* of 0.88 on the author's own images
Song *et al.* (2002)	Propagates a rectangular segment for lines and three rectangular segments for circular arcs, progressively simplifying the drawing. Suitable for technical and engineering drawings	Average *VRI* of 0.95 and an average *ECI* of 0.16
Hilaire and Tombre (2006)	Fuzzy segmentation of the skeleton into lines and arcs, followed by skeleton simplification and unification of parts. Suitable for technical and engineering drawings	An average *ECI* of 0.68 and an average *VRI* of 0.74 on sample images from the *GREC 2003*[a] contest images
Keysers and Breuel (2006)	Performs run-length encoding directly on the binary image and fits geometric primitives (lines and arcs) to resulting codes, using a branch-and-bound technique to recursively subdivide the parameter space. Suitable for technical and engineering drawings	A *VRI* of 0.76 on the *GREC 2003* contest images
Noris *et al.* (2013)	Analyses the drawing topology to overcome junction ambiguities. Handles freehand sketches	Evaluated on the author's own drawings. An average *CE* of 4.15% of line thickness and an *SPE* smaller than the linewidth
Chen *et al.* (2015)	Performs contour tracking on the image to detect line junctions. The position of the junction is then determined through intensity polar mapping. The bending degrees of contour paths are used to determine the junction topological structure. Handles freehand sketches	Visual inspection using the author's own image database
Pham *et al.* (2014)	Detect zones of high curvature on the image skeleton to detect junction zones. The skeleton topology around this zone is then used to optimize the position of the junction. Handles rough sketches	A *repeatability score* of 75.25% on rotation, scale and noise variations on GREC 2011 drawings
Favreau *et al.* (2016)	Image skeleton is over-segmented and represented as a hypergraph. Segments are then merged according to simplicity and fidelity criteria. Junction locations determined through curve connectivity constraints. Handles rough sketches	An average displacement between ground truth and extracted vectors of 0.6 pixels for a line width of 24 pixels
Bessmeltsev and Solomon (2019)	Uses frame-field processing to disambiguate junctions, using graph structure to preserve the drawing topology. Handles freehand sketches	Visual comparison using the images from Favreau *et al.* (2016) and Noris *et al.* (2013)

[a]http://www.cvc.uab.es/grec2003/

drawback lies in the intensive computation to search and manage the junction catalog. Moreover, specifying the geometry alone is not sufficient for the formation of the 3D shape since there may be numerous 3D inflations of the sketch which satisfy this geometry. Thus, optimization-based methods, such as those described in Lipson and Shpitalni (2007) and Liu *et al.* (2011), use shape regularities, such as orthogonality and parallel edges, to obtain a 3D inflation which closely matches the human interpretation of the drawing as illustrated in Figure 5. Alternatively, the initial inflation can make use of perspective or projective geometries, for example, by locating vanishing points to estimate the projection center, then using camera calibration techniques to estimate the 3D geometry (Mitani *et al.*, 2002).

The problem remains in deducing the hidden, unsketched part of the drawing. Algorithms, such as that described in Ros and Thomas (2002), obtain the full 3D structure by solving planar equations of the object surfaces and assume that a wireframe drawing of the object is available. However, when people sketch, they typically draw only the visible part of the object such that the wireframe drawing is not always readily available. Moreover, our visual understanding of sketches allows us to infer the hidden parts of the drawing without too much effort (Cao *et al.*, 2008).

Identification of hidden sketch topology typically starts from the geometric information held within the visible, sketched parts. In general, several plausible connections between the existing, visible vertices in the drawing are created to obtain a

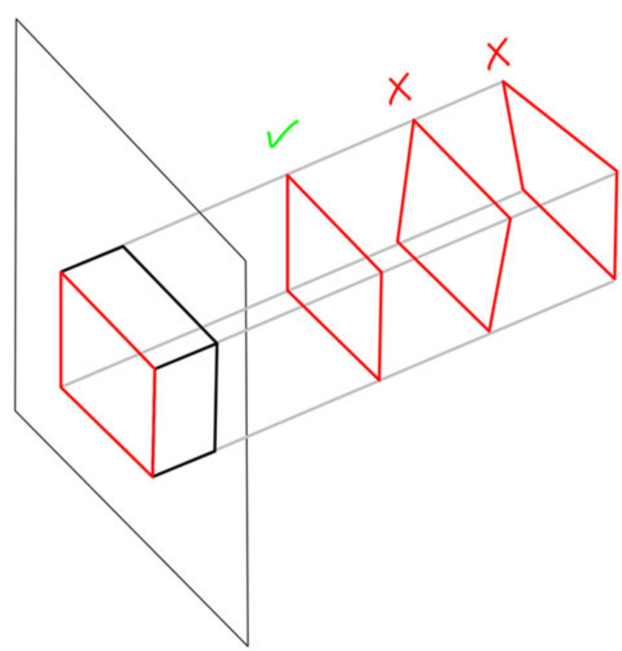

Fig. 5. A 2D drawing may have several 3D inflations. Optimization algorithms based on heuristic regularities, such as orthogonality and parallel edges, may be used to prune out unlikely interpretations.

reasonable, initial wireframe representation of the drawing. This initial representation is then modified by breaking links, introducing new vertex nodes to merge two existing edge branches, or introducing new edge branches to link two otherwise disconnected vertices (Cao *et al.*, 2008; Varley, 2009). These modifications are carried out in such a way that the final hidden topology satisfies some heuristics, mainly based on human perception principles, such as the similarity between the hidden faces and visible faces (Cao *et al.*, 2008), retaining collinear and parallel relationships, and minimizing the number of vertices in the topology (Kyratzi and Sapidis, 2009). An exhaustive exploration of all the possibilities with which the visible vertices can be combined to form the hidden topology remains a problem. Kyratzi and Sapidis (2009) resolve this problem by adopting graph-theoretical ideas, allowing for multiple hypotheses of the hidden topology to exist in the branches of the tree structure.

The main limitation in the interpretation of paper-based sketched drawings remains that of the accuracy of the drawing. Misrepresentation of a junction point will result in a bad match between the sketched junction and the cataloged junctions which in turn results in incorrect geometry labels. This error will then propagate to the sketch inflation and estimation of the hidden viewpoints. Human interpretation of a 3D shape from sketches can, however, tolerate considerable variations from the true geometric form. Moreover, the geometric constraints used in the creation of depth from drawings may not fully represent the human ability of understanding the shape from sketches. Learning-based approaches may, therefore, be more suited to this task. Lun *et al.* (2017) consider such an approach, using deep networks to translate line drawings into 2D images representing the surface depth and normal from one or more viewpoints. A 3D point cloud is generated from these predictions, following which a polygon mesh is formed. Although the architecture proposed by Lun *et al.* (2017) can be trained to provide a reconstruction from a single viewpoint, the network architecture does not obtain sufficient information from this single viewpoint to reconstruct the 3D shape accurately. Lun *et al.* (2017) propose to allow users to introduce viewpoints interactively, bridging the gap between offline interpretation techniques and interactive sketching.

The interpretation of offline sketches, therefore, relies on the ability of the vectorization algorithms to extract the lines and junctions that form the drawing topology as well as the interpretation algorithms that convert the 2D drawing into its 3D geometry. Perhaps the most difficult task of the vectorization algorithms lies in the identification of the line strokes and junction locations when presented with rough sketches consisting of overtracing and scribbling. While the most recent literature, particularly the works proposed by Pham *et al.* (2014) and Favreau *et al.* (2016), allows for some degree of roughness in the drawing, vectorization algorithms need to be robust to a greater degree of scribbling and incomplete drawings if these algorithms are to be useful for an early-stage design.

The interpretation algorithms which add depth to the 2D drawing also need to become more robust to drawing errors which can be introduced due to the quick manner with which initial sketches are drawn. While learning approaches, such as that described in Lun *et al.* (2017), can relax the need for strict adherence to correct geometry, a compromise between learning wrong interpretations of a junction geometry and the possibility of handling badly drawn junctions needs to be reached. The algorithms must, therefore, be able to learn the broader context in which the junction is being interpreted. Human artists resolve ambiguities through the use of artistic cues such as shadows and line weights. Thus, learning-based algorithms could also include the interpretation of these cues when gathering evidence on the geometric form of the sketched object.

Interactive sketches

The availability and increasing popularity of digital tablets brought about a shift in the sketching modality from the traditional pen-and-paper to interactive sketches drawn using digital ink. Sketch-based interfaces, such as Sketch (Zeleznik *et al.*, 2006), Cali (Fonseca *et al.*, 2002), NaturaSketch (Olsen *et al.*, 2011), Teddy (Igarashi *et al.*, 1999), Fibermesh (Nealen *et al.*, 2007), and DigitalClay (Schweikardt and Gross, 2000) among many others, make use of additional inked gestures to allow users to inflate or mold the 2D drawings into a 3D shape.

Sketch-based interfaces often require that the user creates sketches using some particular language. For example, in Teddy (Igarashi *et al.*, 1999), the user draws a simple 2D silhouette of the object from which the 3D shape is constructed through the operation of blobby inflation. The algorithm first extracts the chordal axis of the triangulated mesh of a given silhouette. Then, an elevating process is carried out to inflate the 2D shape into 3D space, which is mirrored by the other side of the shape. The system demonstrates a simple but effective interface of sketch-based modeling. However, it can only handle simple and bulbous shapes, and hence cannot be easily generalized to other shape modeling such as shapes with sharp features.

While sketch-based interfaces overcome some of the difficulties in the interpretation of the sketch, they introduce a sketching language which distracts from the natural spontaneity of freehand sketching. Moreover, the interfaces are often designed such that the user progressively refines the 3D shape (Masry and Lipson, 2007; Xu *et al.*, 2014), which can be time-consuming.

Delanoy *et al.* (2018) learn to reconstruct 3D shapes from one or more drawings, using a deep CNN that predicts occupancy of a voxel grid from a line drawing. The CNN provides an initial 3D reconstruction as soon as the user draws a single view of the desired object. This single view will be updated once a new viewpoint is introduced, using an *updater* CNN to fuse together information from different viewpoints without requiring stroke correspondences between the drawings. The CNN is trained on primitive shapes, such as cubes and cylinders, which are combined through geometric additions and subtractions to create objects which have flat orthogonal faces but which can also support holes or protruding convex surfaces. A total of 20,000 such objects were created, using 50 of these shapes for testing, while the rest were used for training. Li *et al.* (2018) also adopt a CNN to infer depth from the 2D sketch of the 3D surface. This approach, however, aims at rendering general freeform shapes, focusing on geometric principles and optional user input to overcome drawing ambiguities. The CNN is used to predict depth/normal maps using flow field regression and a confidence map, which gives the ambiguity at each point of the input sketch. As with Delanoy *et al.* (2018), the user first draws a single viewpoint of the object which is rendered as a 3D object. The user can then either further modify the surface by drawing curves over the surface, or providing depth values at sparse sample points, or reuse the frontal sketch to draw the back view of the object. Training data for CNN was generated by applying NPR to 3D shapes to attain sketch-like silhouette curves.

Table 2. Comparison of interactive sketch-based algorithms

Algorithm	Brief description	Evaluation
Teddy (Igarashi *et al.*, 1999)	Creates a closed polygonal mesh model from the initial stroke which the user can then modify through the use of extrusions, cutting and smoothing operations each of which changes the underlying polygonal structure such that it fits the new strokes. The tool supports the formation of blob-like freeform objects	Informally evaluated among computer graphics researchers and students who reported becoming fluent with the tool within 10 min of tutorial and guided practice
DigitalClay (Schweikardt and Gross, 2000)	The user draws a sketch representation of the object using an isometric viewpoint. The user strokes are "beautified", that is, lines straightened and adjacent strokes connected to each other to form corners. This initial sketch can be modified by the user by either dragging the corner points or drawing over the edges. Huffman-Clowes labeling is then applied to the drawing before creating the 3D representation of the object	n/a
Masry and Lipson (2007)	The user draws a wireframe representation of the object which is reconstructed assuming an underlying orthogonal axis system. Hidden lines are not shown in the rendered reconstruction, and the user can rotate the resulting object and modify it accordingly. This approach supports planar objects with straight or curved contours	Visual comparison by the authors of objects created through this approach. Objects consist of initial sketches with up to 50 strokes which are processed in interactive time
Sketch (Zeleznik *et al.*, 2006)	A gesture-based user interface to create 3D primitives such as cones, cylinders, spheres, and prisms. Other gestures can be used to modify these primitives through extrusions, rotation and scaling deformations. Shadows can be sketched to show spatial displacement	n/a
Fibremesh (Nealen *et al.*, 2007)	Similar to previous works, the user generates an initial 3D model by sketching a contour which is then inflated. The user can then modify this inflation. However, the stroke modifications are applied directly to the 3D surface and are added as a collection of 3D curves. The interface supports mainly the creation of blob-like, freeform objects. However, curve deformations and surface optimizations allow for the modeling of sharp features on these surfaces	Informal evaluation with first-time, novice users as well as one professional 2D animation artist. The users are reported to have found the system easy-to-use and that it supports the transfer of skills acquired in 2D sketching to 3D modeling
NaturaSketch (Olsen *et al.*, 2008)	A sketch can be drawn over existing images to aid the drawing of objects in proportion. A planar mesh is constructed over the sketched strokes, and this is inflated using a functional vertical displacement to create a 3D surface. Further annotations can be made on the generated surface to introduce bumps, holes, extrusions, and embossing deformations to the object	Evaluated through an informal study with eight participants, five of whom are graphics researchers, one was a 2D pencil artist and the other two did not have experience with drawing or modeling. It was observed that participants drew better when tracing on an image
True2Form (Xu *et al.*, 2014)	Sketches of objects are imported into the user interface and the user may annotate these sketches with different colored strokes to indicate properties such as orthogonalities, parallelism, symmetry, and curve planarity. These properties are then used as constraints in the reconstruction framework. The algorithm can handle freeform objects	The authors provide a visual comparison between the models generated by their system and those generated by similar user interfaces
Delanoy *et al.* (2018)	Use a CNN to provide an initial 3D reconstruction which is updated when the user draws another viewpoint. Can be used to model man-made objects with flat, orthogonal faces but through the addition and subtraction of primitives also supports holes and convex protrusions	Evaluated by two expert users and six other participants with limited drawing/3D modeling skills. Users were asked to model one of two shapes. Participants were quick to learn how to use the system but found that the resulting 3D models lacked detail
Li *et al.* (2018)	Use a CNN to predict depth/normal maps using flow field regression. The user is allowed to draw additional lines on the sketch to refine or add details to the surface. Freeform smooth shapes, with rolling guided normal filtering to allow for sharp features on the smooth surfaces	Five novice users with little knowledge in sketching and 3D modeling were asked to do three modeling tasks in which they recreated target shapes

Table 2 compares some of the interactive sketch-based interfaces described in the literature. We observe that interactive systems which perform an initial blob-like inflation of the drawing, seem to require less effort to create the first 3D model of the object. However, these systems then require effort to deform the initial shape to the desired object form. On the other hand, language-based systems may require more effort to create the initial model, but then, this would require fewer deformations to achieve the desired shape. Deep learning systems appear to provide a good middle ground, using viewpoints which are a natural way of reducing ambiguities. By removing the need to align the viewpoints, these systems reduce the burden on the user. However, these systems produce models which appear to be lacking in object detail. Hence, more work is required to capture the desired level of surface detail with such systems.

Sketch-based shape retrieval

The interpretation methods discussed thus far attempt to create a new 3D model based on the sketched ink strokes. An alternative approach to generating the 3D model linked to the sketch is to assume that a model already exists in some database and that the sketch may be used to retrieve the best fitting model. Sketch-based shape retrieval engines have been used to improve human–computer interaction in interactive computer applications and have been studied since the Princeton Shape Benchmark (Shilane *et al.*, 2004). While sketches are generally simpler than images, they sufficiently describe 3D shapes and are easily produced by hand. Sketch-based retrieval engines for 3D models are important in large datasets where we require fast and accurate methods.

In the approach described by Shilane *et al.* (2004), the user draws the side, front, and top views of the 3D object to retrieve the 3D object whose shape agrees most closely to the given views. Retrieval-based modeling algorithms then consist of three steps, namely view selection and rendering, feature extraction and shape representation, and metric learning and matching (Chen *et al.*, 2003; Pu and Ramani, 2005; Yoon *et al.*, 2010). To improve the quality of the retrieval, efforts are made for more effective descriptors of both sketches and shapes. For instance, in Chen *et al.* (2003), light field descriptors are extracted to represent 3D shapes.

Complex objects can then be modeled by retrieving and assembling the object in a part-wise manner (Chen *et al.*, 2003), while complex scenes comprised of different objects can be modeled by retrieving each object individually (Eitz *et al.*, 2012). However, retrieval-based methods require very large collections of shapes. Moreover, despite the size of the dataset, the likelihood of finding an identical match between a 3D shape and its sketched counterpart is very small. This is because sketch-based retrieval algorithms typically assume that the sketched drawing will match one of the selected viewpoint representations of the object in the database. However, there can be no guarantee that the user's sketch will match the selected object viewpoint. Nor is there a guarantee that the sketching style will correspond to the database object representation. Thus, shape retrieval algorithms also focus on improving the matching accuracy between the sketched query and the shape database, for example, in Wang *et al.* (2015), CNNs are used to learn cross-domain similarities between the sketch query and the 3D object at the image level by projecting 3D shapes into 2D images, thus avoiding the need to specify the object viewpoint. In Zhu *et al.* (2016), a cross-domain neural networks approach is proposed to learn the cross-domain mapping between 2D sketch features and 3D shape features directly. A different method, deep correlated metric learning (DCML), is proposed in Dai *et al.* (2018). The proposed DCML exploits two different deep neural networks for each domain to map features into feature space with a joint loss. They aim to increase the discrimination of features within each domain as well as the correlation between different domains, thus minimizing the discrepancy across the sketch and the shape domain. The proposed method in this study performs significantly better than the proposed method in Wang *et al.* (2015).

One approach to implementing database queries is to learn alternative representations of 3D shapes to improve retrieval performance. Xie *et al.* (2016) propose to learn the barycenters of 2D projections of 3D shapes for significant improvement in sketch-based 3D shape retrieval. A different approach is to convert the database contents into a sketch-like form since this would make subsequent query matching more straightforward. Thus, lines making up strokes should be extracted from 2D images. The same approach can be deployed for 3D models by first generating multiple 2D views, from which the lines are extracted, or else the lines can be directly extracted from the geometry of the 3D model. For example, in Eitz *et al.* (2012), the main idea is to generate a set of 2D sketch-like drawings from the 3D objects in the database, thus, performing matching in 2D rather than direct matching between the 2D sketch and the 3D shape.

2D image-based line detection

Extracting lines from images has been a well-studied topic in computer vision for more than 20 years. In particular, there are a number of common applications in areas such as medical imaging (e.g., blood vessel extraction from retinal images) and remote sensing (road network extraction from aerial images), and these have spawned a variety of line detection methods such as methods based on eigenvalues and eigenvectors of the Hessian matrix (Steger, 1998); the zero-, first-, and second-order Gaussian derivatives (Isikdogan *et al.*, 2015); and 2D Gabor wavelets among others (Soares *et al.*, 2006). Alternatively, general features such as local intensity features can be used in conjunction with classifiers (e.g., neural network classifiers) to predict the existence of lines (Marin *et al.*, 2011). Line detection can also be applied to NPR which aims at resynthesizing images and 3D models in new styles, which include (but are not limited to) traditional artistic styles. Thus, NPR provides the means to convert the 3D model database contents into a sketch-like form.

One effective NPR approach was described by Kang *et al.* (2007), who adapted and improved a standard approach to line detection, which performs convolution with a Laplacian kernel or a difference-of-Gaussians (DoG). As with some of the methods described above, Kang *et al.* (2007) estimate the local image direction and apply the DoG filter in the perpendicular direction. The convolution kernel is deformed to align with the local edge flow, which produces more coherent lines than traditional DoG filtering.

Another NPR technique related to line detection is the rendering of pencil drawings, in which methods aim to capture both the structure and tone of pencil strokes. The former is more relevant to sketch retrieval, and the approach described in Lu *et al.* (2012) generates a sketchy set of lines while trying to avoid false responses due to clutter and texture in the image. They first perform convolution using kernels as a set of eight line segments in the horizontal, vertical, and diagonal directions. These line segments are set to the image height or width. The goal of this initial convolution is to classify each pixel into one of the eight directions (according to which direction produces the maximum response), thereby producing eight response maps. The second stage of convolution is applied, using the eight line kernels on the eight response maps. The elongated kernels link pixels into extended lines, filling gaps, and slightly lengthening the lines present in the input image, producing a coherent and sketchy effect.

With the advent of deep learning, CNN approaches have also been applied in recent years to image-based line detection for the same applications and tend to out-perform traditional methods. For instance, Xu *et al.* (2018) extracted roads from remotely sensed images using a segmentation model that was designed based on the UNET architecture (Ronneberger *et al.*, 2015) in which the contracting part was implemented using DenseNet (Huang *et al.*, 2017). In addition, local attention units were included to extract and combine features from different scales. Leopold *et al.* (2019) described an approach to retinal vessel segmentation that was also based on the UNET architecture. Across several datasets, the model's performance did not always achieve the state of the art. However, it was argued that this was due to the downsizing of the input images that was performed in order to improve computational efficiency. Gao *et al.* (2018) tackled pencil drawing using a vectorized CNN model (Ren and Xu, 2015) and given the difficulty in obtaining suitable ground truth they used a modified version of Lu *et al.* (2012) to generate the training set. In order to improve the quality of the training data, they manually adjusted the parameters of Lu *et al.'s* algorithm to obtain the best result. This enabled them to build a final deep model that was able to produce more stable results than Lu *et al.* (2012).

As alluded to above, an issue in line detection is coping with noisy data. Many line detection methods also include a postprocessing step for improving the quality of the raw line detection. For instance, Marin *et al.* (2011) apply postprocessing in order to fill pixel gaps in detected blood vessels and remove isolated false positives. Isikdogan *et al.* (2015) and Steger (1998) use the hysteresis thresholding approach that is popular in edge detection: two-line response thresholds are applied, and those pixels above the high threshold are retained as lines, while those pixels below the low threshold are discarded. Pixels with intermediate line responses between the thresholds are only retained if they are connected to pixels that were determined to form lines (i.e., above the high threshold).

Although state-of-the-art image-based line detection methods are fairly effective and robust, they are limited by the lack of RGB images to directly capture the object's underlying geometric information. Another limitation is the lack of semantic knowledge needed to disambiguate conflicting information or handle missing information. These factors make the current methods prone to detecting lines arising from the clutter in the images, whereas humans are better able to focus on the salient contours. Moreover, the output of typical image-based line detection methods can be somewhat fragmented and noisy.

3D model-based line detection

If lines are extracted from 3D models, then these lines can directly reflect the geometry of the object. In comparison, lines extracted from images are determined by the image's intensity variations, which can be affected by extraneous factors, such as illumination, and perspective distortion, meaning that significant lines may easily be missed, and spurious lines introduced.

A straightforward approach to locate lines on the surface of a 3D model is to find locations with extremal principal curvature in the principal direction – such loci are often called ridges and valleys. The curvature of a surface is an intrinsic property, and thus, the ridge and valley lines are view independent. While this might seem advantageous, DeCarlo *et al.* (2003) argued (in the context of NPR) that view-dependent lines better convey smooth surfaces and proposed an alternative that they termed *suggestive contours*. These are locations at which the surface is almost in contour from the original viewpoint and can be considered to be locations of true contours in close viewpoints. More precisely, the suggestive contours are locations at which the dot product of the unit surface normal and the view vector is a positive local minimum rather than zero.

Related work by Judd *et al.* (2007) on *apparent ridges* also modified the definition of ridges to make them view-dependent. They defined a view-dependent measure of curvature based on how much the surface bends from the viewpoint. Thus, it takes into consideration both the curvature of the object and the foreshortening due to surface orientation. Apparent ridges are then defined as locations with maximal view-dependent curvature in the principal view-dependent curvature direction.

This earlier work was systematically evaluated by Cole *et al.* (2008), based on a dataset that they created which contains 208 line drawings of 12 3D models, with two viewpoints and two lighting conditions for each model, obtained from 29 artists. Using precision and recall measures, they quantitatively compared the artists' drawings with computer-generated (CG) drawings, namely image intensity edges (Canny, 1986), ridges and valleys, suggestive contours and apparent ridges. They showed that no CG method was consistently better than the others, but that instead, different objects were best rendered using different CG methods. For instance, the mechanical models were best rendered using ridges and edges, while the cloth and bone models were best rendered using occluding contours and suggestive contours. Cole *et al.* (2008) experimented with combining CG methods, and found for example that folds in the cloth model could be identified by the presence of both suggestive contours and apparent ridges. They also found that the artists were consistent in their lines, and in a later user study showed that people interpret certain shapes almost as well from a line drawing as from a shaded image (Cole *et al.*, 2009), which confirms the hypothesis that a sketch-based interface should be an effective means of accessing 3D model information.

In contrast to image-based line detection, there has been substantially less take-up of deep learning for 3D models due to the irregular structure of 3D data. Thus, a common approach is to process 3D as multiview 2D data. For instance, Ye *et al.* (2019) take this approach. They aim to generate line drawings from 3D objects but apply their model to 2D images rendered from the 3D object. Their general adversarial network (GAN) incorporates long short-term memory (LSTM) to enable it to generate lines as sequences of 2D coordinates. Nguyen-Phuoc *et al.* (2018) propose a CNN-based differentiable rendering system for 3D voxel grids which is able to learn various kinds of shader including contour shading. While voxel grids provide the regular structure missing from meshes, in practice their large memory requirements limit their resolution, and so Nguyen-Phuoc *et al.* (2018) resize their 3D data to a $64 \times 64 \times 64$ voxel grid.

Displaying the search results

Equally important in the sketch-based retrieval approach is the way the matching results are presented to the user for the user to make full benefit of search. Traditionally, search results are displayed as thumbnails (Shilane *et al.*, 2004) and applications, such as Google's 3D Warehouse[2], allow the user to select and modify the viewpoint of the object. These display strategies, however, do not take into account the advantages of human–computer interaction paradigms and devices. Adopting VR/AR environments for the exploration of search results have the advantage of allowing far more content to be displayed to the user by making full use of the 3D space to organize the content, allowing the user to examine search results with respect to three different criteria simultaneously (Munehiro and Huang, 2001). The challenge here is to determine how to arrange the query result in the open

[2] https://poly.google.com/

3D space such that the organization remains meaningful to the user as the user navigates in the 3D space. While the 3D axis has been used for such purposes, with each axis defining a search criterion, the display problem is a more complex problem and requires more attention. Also, challenging is establishing the way the users interact with the search objects in the immersive environment. While gestures seem like the most natural interaction modality, the interpretation of unintended gestures may lead to undesirable states (Norman, 2010).

Beyond the single-user, single-sketch applications

The applications discussed thus far focus on single-user, single-object, sketch-to-3D applications. While this remains a significant research challenge, sketch communication is not limited to single-user applications, nor does it have to be focused on individual objects. Sketches may be used in communication with multiple parties and may capture not only the physical form of the object but also the interaction of the sketched object with other objects in its environment or the functionality of the object. The interpretation of the sketch, therefore, goes beyond the interpretation of the ink strokes but should include other means of communication, such as speech or eye-gaze, which occur while sketching. The collaborative aspect of sketching may be extended from the physical world to the VR or AR domain, where improved tools make virtual sketching more accessible. VR and AR opens sketching applications to sketching directly in the 3D sketching domain and to applications where collaborators may be present together in the virtual world. The following sections discuss these aspects of sketching interfaces in greater depth.

Multimodal sketch-based Interaction

When people sketch, particularly when sketching is taking place in a collaborative environment, other, natural and intuitive methods of communication come into play. Thus, combining sketch interpretation with different sources of information obtained during the act of sketching increases the richness of the data available for understanding and interpreting the sketch to improve the user-interface experience. Hence, the need for multimodal sketch-based interactions.

Informal speech is one of the leading interactions in multimodal sketch-based systems since speech is a natural method of communication and can provide additional information beyond that captured in the sketch. The research questions that arise are two-fold: how will the user using such a system want to interact with the system and how will the system analyze the conversation that has arisen? Experiments have been carried out to find answers to these questions by analyzing the nature of speech–sketch multimodal interaction. These studies investigate general tendencies of people, such as the timing of the sketch (Adler and Davis, 2007), and the corresponding conversation interaction to design effective sketch–speech-based systems (Oviatt *et al*., 2000).

During sketching, people exhibit subtle eye-gaze patterns, which in some cases, can be used to infer important information about user activity. Studies demonstrate that people perform distinguishing eye-gaze movements during different sketch activities (Çığ and Sezgin, 2015). Thus, the natural information coming from eye-gaze movements can be used to identify particular sketch tasks. These observations lead researchers to take eye-gaze information into account when creating multimodal sketch-based interaction. For example, in Çığ and Sezgin (2015), eye-gaze information is used for early recognition of pen-based interactions. They collect data from 10 participants (6 males and 4 females) over five different tasks to train an SVM-based system for task prediction with an 88% success rate. They demonstrate that eye-gaze movement that naturally accompanies pen-based user interaction can be used for real-time activity prediction.

An important aspect which eye-gaze and speech prediction systems need to address is the ability of the systems to recover from prediction errors. While error recovery is important in all steps of the sketch interpretation, it becomes more critical with eye-gaze and speech predictions since the user's natural reaction to errors would be to change their behavior in an attempt to force the system to correct the error. Such a change in behavior could, in turn, further reduce the prediction performance.

While eye-gaze and speech provide information about the sketch, haptic feedback is a different mode of interaction which provides information to the user, conveying the natural feeling of interaction to the user. Haptic feedback changes the sketch interaction in VR or AR applications, providing a realistic substitute for the interaction with physical surfaces (Strasnick *et al*., 2018). In this study, they propose three prototypes a novel VR controller, Haptic Links, that support the haptic rendering of a variety of two-handed objects and interactions. They conduct a user evaluation with 12 participants (ages 25–49, 1 female) that shows users can perceive many two-handed objects or interactions as more realistic with Haptic Links than with typical VR controllers. Such a feedback is of particular use when the virtual environment plays a significant role in sketch interaction. Such tasks include sketching or drawing on a virtual object or writing on a board, where haptic feedback enhances the user experience through the physical feelings of the virtual surface. Systems which include haptic feedback use principles of kinematics and mechanics to exert physical forces on the user. For example, in Massie and Salisbury (1994), a force vector is exerted on the user's fingertip to allow the user to interact with and feel a variety of virtual objects including controlling remote manipulators, while in Iwata (1993), a pen-shaped gripper is used for direct manipulation of a freeform surface.

Augmented and virtual reality

The qualities of sketching as an easy and efficient method to create visual representations have also had an impact in the field of VR and AR. Virtual and augmented media are inherently 3D spatial media, and thus, sketching in VR and AR involves usually the creation of 3D visual representations. Such systems typically allow users to draw and immediately perceive strokes and planes in 3D space. Users create strokes by using input devices, such as controllers or pens, which are also tracked by the VR system. Users can easily perceive the drawings from different angles by just moving their head and body. Table 3 provides a comparative summary of the VR and AR interactions discussed here under.

Early immersive sketching systems were developed by Keefe *et al.* (2001), who created a sketching environment for artists within a cave automatic virtual environment (CAVE), Fiorentino *et al.* (2002), who tried to introduce 3D sketching in industrial styling processes, or Schkolne *et al.* (2001), who suggested to use bare hands for the creation of rough sketches. The *Front Design Sketch Furniture Performance Design*[3] project demonstrated an AR-alike application of freehand 3D sketching for the design of furniture, including printing of the results using

[3]http://www.frontdesign.se/sketch-furniture-performance-design-project

Table 3. Overview of immersive sketching studies

Project/ authors	Environment	Aim and metrics	Participants	Key findings
Cave painting Keefe *et al.* (2001)	VR CAVE	Uses informal observations to evaluate the feasibility of immersive modeling	Two art classes, painters, art students of varied backgrounds	Demonstrates the feasibility of immersive modeling to create pieces of art
Drawing on air Keefe *et al.* (2007)	Desktop VR	Evaluation of haptic support for immersive sketching by recording the positional error and the drawing time	12 experienced designers and artists	Haptic support allows for drawings with roughly half the positional and rotational errors in comparison to freehand 3D sketching
Immersive sketching Israel *et al.* (2009)	VR CAVE	Comparing user behavior during 2D and 3D sketching by observing the sketch size and details, the usage time, and user speed	24 furniture designers and interior architects	3D sketches were almost twice as large, users moved 33% faster, the sketches contained twice as much detail, and usage times were 50% higher in the 3D condition as in the 2D condition
Learnability Wiese *et al.* (2010)	VR HMD	Measure the line accuracy, uniformity, shape deviation, workload time, line matching, and corrective movements while sketching	25 students, 12 designers, 4 architects, 1 visual artist, and 4 art teachers	Line accuracy, uniformity of objects, shape deviation increased after 30 minutes
Sketching on surfaces Arora *et al.* (2017)	VR HMD	Evaluation of possible accuracy benefits from sketching on virtual and physical surfaces compared to no support (free mode). Measures the depth deviation and smoothness of curvature	12 participants without immersive sketching experience	Physical drawing surfaces improved accuracy by 22% and virtual surfaces by 17%
Multiplanes Barrera Machuca *et al.* (2017)	VR HMD	Measures the acceptance of real-time "beautification" of strokes through the ease of interaction, perceived speed and accuracy, and the overall opinion	8 undergraduate students (2 with prior immersive sketching experience)	In average medium ratings (4.5 on a 7-point Likert scale), 6 participants were excited about the features
Physical guides Wacker *et al.* (2018)	AR HMD	Comparing the sketching performance of virtual and physical objects of the same size. Measures the deviation in 3D in the x, y, and z directions	16 participants (11 with no prior AR or immersive drawing experience, 4 with AR but no immersive drawing experience, 1 with immersive drawing but without AR experience)	Tracing physical objects 48% more accurate than tracing virtual objects with no haptic support

rapid prototyping technologies. Among the most recent immersive sketching systems are commercially available tools such as Tilt Brush[4] and Gravity-Sketch[5] which provide modeling functionalities, including the creation of strokes, surfaces, volumetric meshes, or primitives. Such tools provide advanced 3D user interfaces, menus, and widgets and can be run with most of the latest VR headsets and 3D controllers.

The VR market has seen a major technology shift in the past years particularly, with the emergence of affordable high-resolution head-mounted displays (HMDs) in the consumer markets. Industrial VR solutions make more and more use of HMDs which today significantly outnumber projection-based solutions. This shift is also visible in the field of immersive sketching. Where earlier works such as those described in Fiorentino *et al.* (2002), Keefe *et al.* (2001), Israel *et al.* (2009), and Wiese *et al.* (2010) among others, mainly used projection-based solutions, recent research systems such as those described in Arora *et al.* (2017) and Barrera Machuca *et al.* (2017) and commercial systems such as Tilt Brush and Gravity-Sketch, typically employ HMDs. HMDs offer the advantages of lower costs, lower space requirements, and increased mobility in comparison to projection-based systems. However, HMDs typically block the view of the physical environment, whereas in projection-based systems, users can see each other, even though usually only one user can perceive the 3D scene from the right perspective (Drascic and Milgram, 1996).

A considerable number of studies has investigated the characteristics of immersive freehand sketching. In their seminal paper "CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience", Keefe *et al.* (2001) were the first to show that immersive sketching within a CAVE can foster creative drawing and sculpting processes among artists. They reported their observations of participants from two art classes, painters, and art students of varied backgrounds. They noted that skilled artists, young artists, and novice users were able to create "meaningful piece[s] of art" (*op. cit.*, p. 92) with their system. In another study among 12 experienced designers and artists, Keefe *et al.* (2007) investigated the effects of two bimanual 3D drawings techniques with haptic support. They found that these techniques allow for drawings with roughly half the positional and rotational errors compared to freehand 3D sketching techniques. Israel *et al.* (2009) compared 2D and 3D sketching processes and the resulting sketches. In a study among 24 furniture designers and interior architects, they found that 3D sketches were almost twice as large

[4]https://www.tiltbrush.com/
[5]www.gravitysketch.com/vr/

as 2D sketches, users moved 33% faster in the 3D condition than in the 2D condition, the resulting 3D sketches contained twice as many details than 2D sketches, and usage times were 50% higher in the 3D condition. Furthermore, users reported that it felt more "natural" to draw three-dimensionally in a 3D environment. The 3D environment seemed to support the creation of 3D representations in one-to-one scale and to foster the interaction with sketches from the moment of their creation, which could, in turn, stimulate creative development processes. In an effort to investigate the effects of visual and physical support during immersive sketching, Arora *et al.* (2017) discovered in an observational study among five design experts that they prefer to switch back and forth between controlled and free modes. In another study among 12 participants without professional drawing experience, Arora *et al.* (2017) used depth deviation and smoothness of curvature as a measure of accuracy and show that a physical drawing surface helped to improve the accuracy of a sketch by 22% over their free mode counterpart. Virtual surfaces, which are easy to implement, were surprisingly close with a 17% improvement. The use of visual guides, such as grids and scaffolding curves, improved the drawing accuracy by 17% and 57%, respectively. However, the drawings were less esthetically pleasing than the free mode sketches, especially with the use of scaffolding curves. A system developed by Barrera Machuca *et al.* (2017) followed another approach. Here, 3D strokes were projected onto 2D planes and corrected or "beautified" in real time. In a preliminary evaluation among eight undergraduate students, of which 25% had prior immersive sketching experience, users appreciated this informal and unobtrusive interaction technique and were satisfied with the quality of the resulting sketches.

The question of how fast users can adapt to immersive sketching was subject to a learnability study with 25 design, arts, and architecture students by Wiese *et al.* (2010). In the study, Wiese *et al.* (2010) measured immersive sketching abilities during three test trials occurring within 30 min of each other and in which users had to draw four basic geometries. Wiese *et al.* (2010) report improvements of approximately 10% in line accuracy, 8% in shape uniformity, and 9% in shape deviation. These results underline the hypothesis that immersive sketching skills can improve over time, even after short periods of learning.

With the growing popularity of AR, some AR-based 3D sketching approaches recently surfaced. In AR, the user can perceive their physical environment, seamlessly augmented with virtual information and objects. Typical AR frameworks either use the hardware of mobile device, for example, Apple ARKit[6], Google ARCore[7], and Vuforia[8] or HMDs, for example, Microsoft HoloLens[9]. Both frameworks have the potential for drawing and sketching applications. Smartphone-based solutions typically use the motion, environmental and position sensors as well as the device's camera to determine its position in space. The user can either draw directly on the screen or by moving the screen.

Among the AR-based sketching systems, SketchAR[10] helps users to increase their drawing skills. The application uses the phone's camera to capture the physical environment. When the system detects physical paper in the image, the user may overlay a template, such as the sketch of a face as shown in Figure 6, onto the physical paper. The user can then use physical pens to trace the template on the physical sheet of paper while controlling the result on the smartphone display. CreateAR[11], another AR-based sketching applications, allows users to create and place sketches at particular geo-locations, making them accessible for other users (Skwarek, 2013). Similar applications are also available for Microsoft's HoloLens; most applications let the user draw by pressing the thumb against the forefinger, creating strokes when the user moves their hand. In a study among 16 participants (4 female, 12 male, 11 with no prior AR nor immersive drawing experience, 4 with AR but no immersive drawing experience, 1 with immersive drawing but without AR experience), Wacker *et al.* (2018) compared the sketching performance on virtual and physical objects of the same size in AR. They found that tracing physical objects can be performed 48% more accurate than on virtual objects with no haptic support.

Interesting research questions remain in the field of learnability, especially in the AR/VR context. Future mid- and long-term studies could investigate to which degree users can develop freehand sketching skills and if they can even reach the accuracy of traditional sketching on paper. Physical and virtual scaffolding curves, grids and planes have been shown, to a limited extent, to improve the drawing accuracy. However, in some cases, these structures result in overloaded user interfaces and reduce the esthetic quality of the resulting sketches. The search for supporting structures which do not compromise the creativity and fluidity of the sketching process will thus remain an essential branch of research in the field of immersive sketching. AR-related research has also shown that haptic feedback also improves accuracy when tracing virtual objects. This raises the question of whether haptic feedback may introduce the structure introduced by grid lines or scaffolding curves without the same hindering effect on the esthetics of the resulting sketches.

Future directions

While there are many breakthroughs in the literature in the area of sketch-based interpretations and interactions, these are not reflected in the tools available in the industry, particularly in the design industry where there still exists a gulf between 2D sketching and 3D modeling for rapid prototyping and 3D printing. Examining the problems faced in industrial applications lead us to identify the following questions and challenges.

Media breaks in the product design workflow

The different nature of the sketches and drawings used at each stage in the design process calls for different software/hardware support throughout the design process (Tversky and Suwa, 2009). For instance, sketch-based modeling which does not require precise dimensions is ideal for the development of 3D models from initial sketches. However, precise dimensions are required at later, detailed design stage, and thus, the sketch-based interface should allow for their introduction. Moreover, while novel AR and VR environments are useful to visualize and interact with the virtual prototypes, the more traditional CAD tools may be more suited for detailed design. One must also take into consideration the human factor: people may be more comfortable and proficient using the tools they are familiar with.

[6]https://developer.apple.com/arkit/
[7]https://developers.google.com/ar/discover/
[8]https://www.vuforia.com/
[9]https://www.microsoft.com/en-ca/hololens
[10]https://sketchar.tech/
[11]https://www.createar.co/

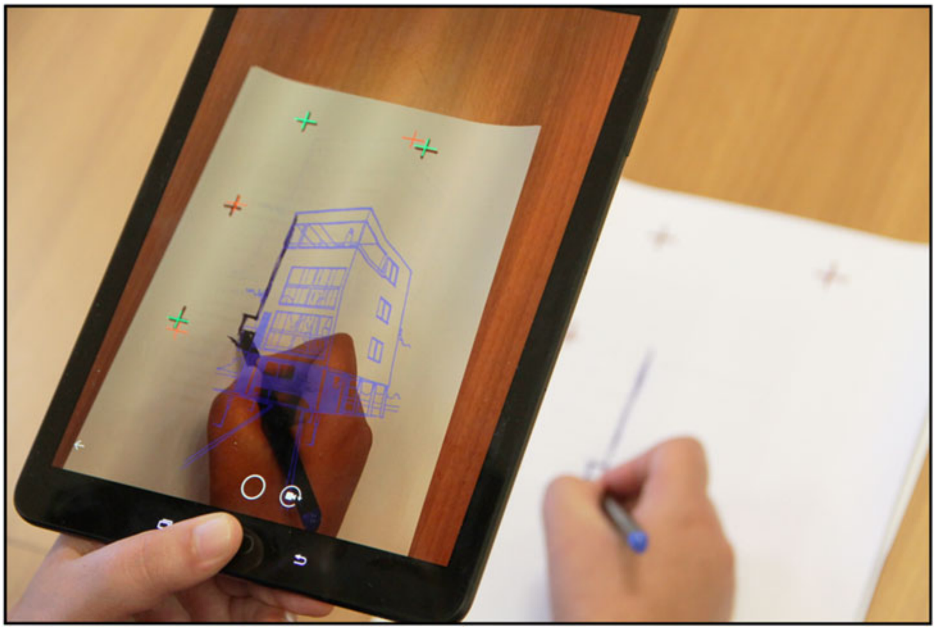

Fig. 6. Sketching with SketchAR: the application overlays a template over the image of the physical paper which the user can then trace onto the paper using physical pens.

The current sketch-based interfaces and sketch-based modeling tools described in the literature do not take these factors into account. Thus, while there is support for sketching systems on 2D media, sketching in AR and VR environments as well as sketch-based queries, these systems are not interoperable, resulting in media breaks which limit the practical use of these systems. What is required is a system which allows for different sketch interpretation systems to integrate seamlessly with each other such that there is no interruption of the workflow. Early work described in Bonnici *et al.* (2015) transitions from a paper-based sketch to a 3D model in a virtual environment, providing a glimpse that seamless transitions between media are possible. Full interoperability will require an investigation into a file interchange format to facilitate the transition of sketch and model data between different applications.

Thinking sketches

There is some considerable difference between sketches drawn at an individual level and those drawn during group brainstorming sessions. Recording multimodal interactions become necessary in group sketching to capture fully the thought process, especially since gestures can be considered as a second-layer sketch. Through the concept of reflection in action, the fluid, mental representation of the concept is objectified and externally represented, refining the concept through gestures.

However, recording and using gestures raises further challenges. Gestures are subconscious actions, unlike sketching, which is a conscious action. Capturing all unconscious actions during sketching, while interesting will overload the interpretation system with information, giving rise to the need to filter out natural gestures, such as habitual arranging of one's hair, which are not related to the act of sketching. Such filtering requires identifying gestures which are commonly used across different cultures and which can be interpreted in the same manner across the board, raising the question of whether it is possible to find such common gestures which have been naturally adopted across different cultures, or if the interpretation system can adapt to the personalization of gestures. However, before a system that records all gestures is brought into place, it is worth investigating whether such a system would bring about a change in the group interaction since full recording may be seen as inhibiting and imposing on the "free-will" of the group participants.

Support for off-site collaborative sketches

Internationalization has brought about a greater need for off-site collaboration in the design process. Technology has made it possible to share media in the form of text documents, sketches, computer-aided models, or physical artifacts which facilitates this collaboration. However, despite the advances in telepresence systems, one of the main bottlenecks, reducing the effectiveness of communication in collaborative work, remains the lack of mechanisms for communicating nonverbal cues such as the locus of attention on the shared media at any given instance in time (D'Angelo and Gergle, 2018). In small groups of two or three, the participants, predominantly the speaker, issues deictic gestures (naturally by hand or finger pointing) to communicate the locus of attention and context. Previous work on communication of distant dyads shows that speech and deictic gestures collectively carry complementary information that can be used to infer regions of interest in 2D shared media (Monk and Gale, 2002; Kirk *et al.*, 2007; Cherubini *et al.*, 2008; Eisenstein *et al.*, 2008). For larger groups, and in particular in remote collaboration, the inability to issue deictic gestures severely limits the quality of communication and makes it difficult to create common ground for communication. Real-time eye-gaze visualizations can, therefore, support the collaboration process by providing the means for communicating shared visual space, thus improving the coordination between collaborating parties (D'Angelo and Gergle, 2018). However, the real-time eye-gaze display is often considered to be distracting because of the low signal-to-noise ratio of the eye-gaze data which is a result of the constant movement of the eyes (Schlösser, 2018). This calls for the need of further investigations into the visualization of shared spaces from eye-gaze information. D'Angelo and Gergle (2018) apply eye-gaze visualizations to search tasks, illustrating shared spaces using two approaches, first by displaying heat-maps which are retained for as long as a pair of collaborators fixate on an area and second through path visualizations. The findings of this study show that while eye-gaze information does facilitate the search task, the used visualization methods received mixed responses.

In collaborative sketching applications, inferred regions of interest could be used to create loci of attention, highlighting the object parts that are under study. Combined with speech information streamed from participants, such a part highlighting, or VR/AR-based augmentation is expected to aid the communication process. Thus, further research is required on the joint fusion of eye-gaze information and speech information such that this can be visualized effectively without introducing unnecessary distractions which are counter-productive to the design process. It is also worth noting that D'Angelo and Gergle consider pairs of collaborators. Thus, the visualization of the co-resolution of the different point of regards in larger group settings also needs to be investigated.

Data tracking: sketch information indexing through the workflow

The different workflows in the design process give rise to different sketches, often by different designers working at different phases in the project. Thus, another important aspect of the design process is the ability to trace through the different sketches, for example, to identify when a specific design decision was taken. The concept of product lifecycle management (PLM) in product design is a management system which holds all information about the product, as it is produced throughout all phases of the product's life cycle. This information is made available to everyone in an organization, from the managerial and technical levels to key suppliers and customers (Sudarsan *et al*., 2005). Smart and intelligent products are becoming readily available due to the widespread availability of related technologies such as RFIDs, small-sized sensors, and sensor networks (Kiritsis, 2011). Thus, although sketching interfaces consider the interaction between the designer and the artifact being designed, it is important to look beyond this level of interaction and consider all stakeholders of the artifact.

Such PLM systems typically include product information in textual or organizational chart formats, providing information different key actors along the product's life cycle on aspects such as recyclability and reuse of the different parts of the product, modes of use and more (Kiritsis, 2011). Expanding this information with the sketches, drawings and 3D modeling information carried out during the design process will, therefore, extend the information contained in the PLM. Consumers would be able to trace back to the design decisions of particular features on the object, while designers would be able to understand how consumers are using the product and could exploit this information, for example, to improve quality goals.

The challenge, therefore, lies in providing the means to establish an indexing and navigation system of the product design history, providing a storyboard of the design process from ideation stages to the final end-product.

Data collection for a reliable evaluation test cases

Related to all of the above is the need to create common evaluation test cases upon which research groups may evaluate their algorithms. Notably challenging is the need to collect and annotate data of people interacting naturally with an intelligent system when such a system is not yet available.

From Table 1, we may observe that the GREC dataset has been used for vectorization algorithms which are more suited for neat, structured drawings such as circuit diagrams or engineering drawings. However, a similar database for freeform objects is not readily available, leading Noris *et al.* (2013), Favreau *et al.* (2016), and Bessmeltsev and Solomon (2019) to use their own drawings. Thus, a dataset useful for the evaluation of vectorization algorithms should depict a broader range of 3D objects. Moreover, these should be available not only as neat drawings but also as rough sketches. Simo-Serra *et al.* (2016) note the difficulty in aligning ground-truth drawings with rough sketches, and employ an inverted dataset reconstruction approach, asking artists to draw rough sketches over the predefined ground-truth sketches. This approach, however, restricts the drawing freedom of the artist. A reverse approach was adopted in Bonnici *et al.* (2018), whereby the artist was allowed to sketch freely, following which, the sketch was scanned and traced over using a new layer in a paint-like application. In this application, however, artists were given drawings depicting 3D objects to copy. While this approach serves the purposes for sourcing of data for vectorization algorithms, a more holistic approach could, for example, require that the artists draw rough sketches of physical 3D objects. In this manner, besides collecting sketches for the vectorization process, a dataset of different sketch representations of 3D objects is also collected. If these sketches are also coupled with depth scans or 3D models of the physical object, the same sketches can also be used to assess 3D object reconstruction and retrieval algorithms.

Conclusion

In this paper, we have presented a review of the state of the art in sketch-based modeling and interpretation algorithms, looking at techniques related to the interpretation of sketches drawn on 2D media, sketch-based retrieval systems, as well as sketch interactions in AR and VR environments.

We discuss how current systems are focused on solving the specific problems related to the particular interpretation problem, however, few systems address the overarching sketch interpretation problem which provides continuity across different sketching media and sketching interactions to support the entire design process.

At an algorithmic level, we note that offline drawing interpretation systems would benefit from robust extraction of junctions and vectors from drawings which exhibit a greater degree of roughness and scribbling than that typically used to evaluate the algorithms described in the literature. At the same time, the interpretation of the 2D sketch as a 3D object requires the added flexibility to accept drawing errors and provide the user with plausible interpretations of the inaccurate sketch. This flexibility is necessary for offline interpretation algorithms to be applicable for early-stage design drawings, which are drawn rapidly and with not much attention to geometric accuracy. We have also discussed online interpretation algorithms, noting that interactive systems typically compromise between fast creation of initial objects and overall deformation and editing time required to achieve the desired object shape. Language-based interactive systems generally require the user to dedicate more effort in reducing ambiguities at the sketch-level but require knowledge of more drawing rules, whereas blob-like inflation systems can provide the initial inflation quickly but require more effort to adjust the object. Combining online and offline interpretation techniques could, therefore, provide solutions whereby the user can obtain the initial 3D representation of the object using offline interpretation techniques, such that the initial model has more detail than the blob-like inflation systems, but can be achieved through natural sketching without the need to introduce any language-based rules. The interactive techniques can then be used to modify the shape of the object to correct for 3D geometry errors due to drawing ambiguities or errors.

We discuss that, while sketch interpretation algorithms are typically concerned only with the generation of the 3D object generated, these interpretation systems can be used in conjunction with sketch retrieval algorithms to help designers retrieve similar objects from online resources and catalogs. Research in this area needs to take into consideration two important aspects. The first is how to match sketchy drawings with objects. Here, NPR techniques can be used to facilitate the matching problems, although these algorithms would benefit from further research in line extraction in order to obtain line drawing representations of objects that are less susceptible to image noise and closer to the human-perceived lines. The second problem to investigate is the visualization of the search results so as not to overwhelm the user with many results. Here, VR and AR systems may provide users with better search space exploration tools.

VR and AR systems offer a platform for sketching in their own right, introducing a variety of user interactions and multiuser collaborations which would, otherwise, not be possible from traditional sketching interfaces. While user studies show that AR/VR sketching systems are viable options, medium- and long-term evaluation studies on the use of these systems are, thus far, not available. Such studies are necessary to help shape future immersive interactive systems.

Overall, although sketch interpretation systems are more intuitive and pleasant to use than the traditional CAD-based systems, these systems are still not widely used among practitioners in the design industry. One factor affecting this is the time and effort required by the user to take the initial sketch into its desired 3D form. For these systems to have any impact among practitioners, the time and effort required to create a model using automated sketch interpretation techniques should be less than that required to redraw the object using traditional CAD systems. Another factor that should be taken into consideration is the interoperability of the interpretation system. Design companies invest resources for systems and tools, and thus allowing for the seamless integration of novel tools with existing work flow practices, designers are more likely to embrace the new technologies being researched and developed.

Future sketch-based interfaces should also support collaborative design, including interactions between all stakeholders of the product design. We believe that addressing the challenges presented in this paper will allow for the development of new sketch interpretation systems that take a more holistic approach to the design problem and will, therefore, be of more practical use to practicing designers.

References

Ablameyko S and Pridmore T (2000) The line drawing interpretation problem. In Ablameyko S and Pridmore T (eds), *Machine Interpretation of Line Drawing Images: Technical Drawings, Maps and Diagrams*. London: Springer, pp. 1–19.

Adler A and Davis R (2007) Speech and sketching: an empirical study of multimodal interaction. *Proceedings of the 4th Eurographics Workshop on Sketch-Based Interfaces and Modelling*, Riverside, California, August 02–03, 2007. New York: ACM, pp. 83–90.

Arora R, Kazi RH, Anderson F, Grossman T, Singh K and Fitzmaurice G (2017) Experimental evaluation of sketching on surfaces in VR. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver, Colorado, USA, May 06–11, 2017. New York: ACM, pp. 5643–5654.

Barrera Machuca MD, Asente P, Lu J, Kim B and Stuerzlinger W (2017) Multiplanes: assisted freehand VR drawing. *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, Québec City, QC, Canada, October 22–25, 2017. New York: ACM, pp. 1–3.

Bessmeltsev M and Solomon J (2019) Vectorization of line drawings via polyvector fields. *ACM Transactions on Graphics* **38**, 9.1–9.12

Bonnici A, Israel JH, Muscat AM, Camilleri D, Camilleri K and Rothenburg U (2015) Investigating user response to a hybrid sketch-based interface for creating 3D virtual models in an immersive environment. *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications*, Berlin, Germany, March 11–14, 2015. Berlin: SCITEPRESS, pp. 470–477.

Bonnici A, Bugeja D and Azzopardi G (2018) Vectorisation of sketches with shadows and shading using COSFIRE filters. *Proceedings of the ACM Symposium on Document Engineering*, Halifax, NS, Canada, August 28–31, 2018, pp. 1–10.

Canny J (1986) A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**, 679–698.

Cao L, Liu J and Tang X (2008) What the back of the object looks like: 3D reconstruction from line drawings without hidden lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**, 507–517.

Chen DY, Tian XP, Shen YT and Ouhyoung M (2003) On visual similarity based 3D model retrieval. *Computer Graphics Forum* **22**, 223–232

Chen JZ, Lei Q, Miao YW and Peng QS (2015) Vectorization of line drawing image based on junction analysis. *Science China Information Sciences* **58**, 1.

Cherubini M, Nüssli MA and Dillenbourg P (2008) Deixis and gaze in collaborative work at a distance (over a shared map): a computational model to detect misunderstandings. *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, Savannah, Georgia, March 26–28, 2008. New York: ACM.

Chhabra AK and Phillips IT (1998) *The Second International Graphics Recognition Contest Raster to Vector Conversion: A Report*. Graphics Recognition Algorithms and Systems, pp. 390–410.

Çığ Ç and Sezgin TM (2015) Gaze-based prediction of pen-based virtual interaction tasks. *International Journal of Human-Computer Studies* **73**, 91–106.

Clowes MB (1971) On seeing things. *Artificial Intelligence* **2**, 76–116.

Cole F, Golovinskiy A, Limpaecher A, Barros HS, Finkelstein A, Funkhouser T and Rusinkiewicz S (2008) Where do people draw lines? *ACM Transactions on Graphics* **27**, 88:1–88:11.

Cole F, Sanik K, DeCarlo D, Finkelstein A, Funkhouser T, Rusinkiewicz S and Singh M (2009) How well do line drawings depict shape? *ACM Transactions on Graphics* **28**, 1–9.

Cook MT and Agah A (2009) A survey of sketch-based 3D modelling techniques. *Interacting with Computers* **21**, 201–211.

Cooper M (2008) A rich discrete labelling scheme for line drawings of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**, 741–745.

Craft B and Cairns P (2009) Sketching sketching: outlines of a collaborative design method. *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology (BCS-HCI '09)*, Cambridge, United Kingdom, September 01–05, 2009. Swinton, UK: British Computer Society, pp. 65–72.

Dai G, Xie J and Fang Y (2018) Deep Correlated Holistic Metric Learning for Sketch-Based 3D Shape Retrieval. *IEEE Transactions on Image Processing* **27** (7), 3374–3386

D'Angelo S and Gergle D (2018) An eye for design: gaze visualizations for remote collaborative work. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*, Montreal QC, Canada, April 21–26, 2018. New York, NY, USA: ACM, Paper 349, p. 12, doi:10.1145/3173574.3173923.

DeCarlo D, Finkelstein A, Rusinkiewicz S and Santella A (2003) Suggestive contours for conveying shape. *ACM Transactions on Graphics* **22**, 848–855.

Delanoy J, Aubry M, Isola P, Efros AA and Bousseau A (2018) 3D sketching using multi-view deep volumetric prediction. *Proc. ACM Comput. Graph. Interact. Tech. 1, 1, Article 21 (July 2018)*, p. 22. doi:10.1145/3203197.

Dori D and Wenyin L (1999) Sparse pixel vectorization: an algorithm and its performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. **21**, No. 3, pp. 202–215.

Drascic D and Milgram P (1996) Perceptual issues in augmented reality. *SPIE Stereoscopic Displays and Virtual Reality Systems III,* San Jose, CA, United States, Vol. **2653**, pp. 123–135.

Eisenstein J, Barzilay R and Davis R (2008) Gesture salience as a hidden variable for coreference resolution and keyframe extraction. *Journal of Artificial Intelligence Research* **31**, 353–398.

Eissen K and Steur R (2007) *Sketching. Drawing Techniques for Product Designers*. Netherlands: BIS Publishers.

Eitz M, Richter R, Boubekeur T, Hildebrand K and Alexa M (2012) Sketch-based shape retrieval. *ACM Transactions on Graphics* **31**, 1–10.

Favreau J-D, Lafarge F and Bousseau A (2016) Fidelity vs. simplicity: a global approach to line drawing vectorization. *ACM Transactions on Graphics* **35**, 120:1–120:10.

Fiorentino M, De Amicis R, Stork A and Monno G (2002) Surface design in virtual reality as industrial application. *Proceedings of the DESIGN Conference*, May 14–17, 2002, Cavtat, Dubrovnik, Croatia, pp. 477–482.

Fonseca MJ, Pimentel C and Jorge JA (2002) CALI: an online scribble recognizer for calligraphic interfaces. *Sketch Understanding, Papers from the 2002 AAAI Spring Symposium*, 25–27 March 2002, Stanford, California.

Gao C, Tang M, Liang X, Su Z and Zou C (2018) PencilArt: a chromatic penciling style generation framework. *Computer Graphics Forum* **37**, 395–409.

Heiser J, Tversky B and Silverman M (2004) Sketches for and from collaboration. In Gero JS, Tversky B and Knight T (eds), *Visual and Spatial Reasoning in Design III*, NSW, Australia: Key Centre of Design Computing and Cognition, University of Sydney, pp. 69–78.

Hilaire X and Tombre K (2006) Robust and accurate vectorization of line drawings. *IEEE Transactions on Pattern Analysis and Machine Interpretation* **28**, 890–904.

Huang G, Liu Z, Van Der Maaten L and Weinberger KQ (2017) Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 21–26 July, Hawai, pp. 4700–4708.

Huffman D (1971) Impossible objects as nonsense sentences. *Machine Intelligence* **6**, 295–324.

Igarashi T, Matsuoka S and Tanaka H (1999) Teddy: a sketching interface for 3D freeform design. *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, August 8–13, 1999, Los Angeles, California, USA. ACM Press/Addison-Wesley Publishing Co, pp. 409–416.

Isikdogan F, Bovik A and Passalacqua P (2015) Automatic channel network extraction from remotely sensed images by singularity analysis. *IEEE Geoscience and Remote Sensing Letters* **12**, 2218–2221.

Israel JH, Wiese E, Mateescu M, Zöllner C and Stark R (2009) Investigating three-dimensional sketching for early conceptual design – results from expert discussions and user studies. *Computers & Graphics* **33**, 462–473.

Iwata H (1993) Pen-based haptic virtual environment. *Proceedings of the IEEE Virtual Reality Annual International Symposium*, September 18–22, 1993, Seatle, USA, pp. 287–292.

Judd T, Durand F and Adelson E (2007) Apparent ridges for line drawing. *ACM Transactions on Graphics* **26**, 19.1–19.7

Kang H, Lee S and Chui CK (2007) Coherent line drawing. *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering*, San Diego, California, August 04–05, 2007. New York: ACM, pp. 43–50.

Katz RA and Pizer SM (2004) Untangling the Blum medial axis transform. *International Journal of Computer Vision* **55**, 139–153.

Keefe DF, Feliz DA, Moscovich T, Laidlaw DH and LaViola Jr JJ (2001) Cavepainting: a fully immersive 3D artistic medium and interactive experience. *Proceedings of the 2001 Symposium on Interactive 3D Graphics*. New York: ACM, pp. 85–93.

Keefe D, Zeleznik R and Laidlaw D (2007) Drawing on air: input techniques for controlled 3D line illustration. *IEEE Transactions on Visualization and Computer Graphics* **13**, 1067–1081.

Keysers D and Breuel T (2006) Optimal line and arc detection on run-length representations. *Proceedings of the Graphics Recognition Workshop*, Hong Kong, China, August 25–26, 2005. LNCS, Springer.

Kiritsis D (2011) Closed-loop PLM for intelligent products in the era of the Internet of things. *Computer-Aided Design* **43**, 479–501.

Kirk D, Rodden T and Stanton Fraser D (2007) Turn it this way: grounding collaborative action with remote gestures. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, San Jose, California, USA, April 28–May 03, 2007. New York: ACM, pp. 1039–1048.

Kyratzi S and Sapidis N (2009) Extracting a polyhedron from a single-view sketch: topological construction of a wireframe sketch with minimal hidden elements. *Computers & Graphics* **33**, 270–279.

Landay JA and Myers BA (1995) Interactive sketching for the early stages of user interface design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Denver, Colorado, USA, May 07–11, 1995. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co, pp. 43–50.

Leopold HA, Orchard J, Zelek JS and Lakshminarayanan V (2019) PixelBNN: augmenting the pixelCNN with batch normalization and the presentation of a fast architecture for retinal vessel segmentation. *Journal of Imaging* **5**, 26.

Li C, Liu X and Wong T (2017) Deep extraction of manga structural lines. *ACM Transactions on Graphics* **36**, Article 117.

Li C, Pan H, Liu Y, Tong X, Sheffer A and Wang W (2018) Robust flow-guided neural prediction for sketch-based freeform surface modeling. *ACM Transactions on Graphics* **37**, Article 238. doi:10.1145/3272127.3275051.

Lins RD, de Almeida MM, Bernardino RB, Jesus D and Oliveira JM (2017) Assessing binarization techniques for document images. *Proceedings of the 2017 ACM Symposium on Document Engineering*, 4th–7th September 2017, Valletta, Malta. Malta: ACM, pp. 183–192.

Lipson H and Shpitalni M (2007) Correlation-based reconstruction of a 3D object from a single freehand sketch. *ACM SIGGRAPH 2007 Courses*, San Diego, California, August 05–09, 2007. New York: ACM, p. 44.

Liu J, Chen Y and Tang X (2011) Decomposition of complex line drawings with hidden lines for 3D planar-faced manifold object reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**, 3–15.

Lu C, Xu L and Jia J (2012) Combining sketch and tone for pencil drawing production. *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, Annecy, France, June 04–06, 2012. Germany: Eurographics Association, pp. 65–73.

Lugt RVD (2002) Functions of sketching in design idea generation meetings. *Proceedings of the 4th Conference on Creativity & Cognition*. Loughborough, UK: ACM, pp. 72–79.

Lun Z, Gadelha M, Kalogerakis E, Maji S and Wang R (2017) 3D shape reconstruction from sketches via multi-view convolutional networks. *2017 International Conference on 3D Vision (3DV)*, 10–12 October 2017, Qingdao, China. IEEE, pp. 67–77.

Marin D, Aquino A, Gegundez-Arias ME and Bravo JM (2011) A new supervised method for blood vessel segmentation in retinal images by using grey-level and moment invariant-based features. *IEEE Transactions on Medical Imaging* **30**, 146–158.

Masry M and Lipson H (2007) A sketch-based interface for iterative design and analysis of 3D objects. *ACM SIGGRAPH 2007 Courses*, San Diego, California, August 05–09, 2007. ACM, p. 31.

Massie TH and Salisbury JK (1994) The PHANToM haptic interface: a device for probing virtual objects. *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, November 1994, Chicago, USA, Vol. **55**(1), pp. 295–300.

Mitani J, Suzuki H and Kimura F (2002) 3D sketch: sketch-based model reconstruction and rendering. In Cugini U and W ozny M (eds), *From*

Geometric Modelling to Shape Modelling, Kluwer International Federation For Information Processing Series, Vol. **208**. Norwell, MA: Kluwer Academic Publishers, pp. 85–98.

Monk AF and Gale C (2002) A look is worth a thousand words: full gaze awareness in video-mediated conversation. *Discourse Processes* **33**, 257–278.

Munehiro N and Huang TS (2001) An interactive 3D visualization for content-based image retrieval. *Proceedings of the IEEE International Conference on Multimedia*, 22–25 August 2001, Tokyo, Japan, Japan, pp. 22–25.

Nealen A, Igarashi T, Sorkine O and Alexa M (2007) Fibermesh: designing free-form surfaces with 3D curves. *ACM Transactions on Graphics* **26**, 41.1–41.9

Nguyen-Phuoc TH, Li C, Balaban S and Yang Y (2018) RenderNet: a deep convolutional network for differentiable rendering from 3D shapes. *Advances in Neural Information Processing Systems*, Montréal, Canada, December 03–08, 2018, pp. 7891–7901.

Noris G, Hornung A, Sumner RW, Simmons M and Gross M (2013) Topology-driven vectorization of clean line drawings. *ACM Transactions on Graphics* **32**, 1–11.

Norman DA (2010) Natural user interfaces are not natural. *Interactions* **17**, 6–10.

Olsen L, Samavati F, Sousa MC and Jorge J (2008) *A Taxonomy of Modelling Techniques Using Sketch-Based Interfaces*. Eurographics state of the art reports, 1 (1.4), 1.

Olsen L, Samavati F and Jorge J (2011) NaturaSketch: modeling from images and natural sketches. *IEEE Transactions on Computer Graphics and Applications* **31**, 24–34.

Oviatt S, Cohen P, Wu L, Vergo J, Duncan L, Suhm B, Bers J, Holzman T, Winograd T, Landay J, Larson J and Ferro D (2000) Designing the user interface for multimodal speech and pen-based gesture applications: state-of-the-art systems and future research directions. *Human-Computer Interaction* **15**, 263–322.

Pham T-A, Delalandre M, Barrat S and Ramel J-Y (2014) Accurate junction detection and characterization in line-drawing images. *Pattern Recognition* **47**, 282–295.

Pu J and Ramani K (2005) A 3D model retrieval method using 2D freehand sketches. *Computational Science, ICCS 2005*, Atlanta, GA, May 22–25, 2005. Springer Berlin Heidelberg, pp. 343–346.

Ramel JY, Vincent N and Emptoz H (1998) A coarse vectorization as an initial representation for the understanding of line drawing images. In Blostein D and Kwon Y-B (eds), *Lecture Notes in Computer Science, Graphics Recognition – Algorithms and Applications*, Vol. **1389**. Berlin, Germany: Springer-Verlag, pp. 48–57.

Ren JS and Xu L (2015) On vectorization of deep convolutional neural networks for vision tasks. *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, January 25–30, 2015, Austin Texas, USA.

Ronneberger O, Fischer P and Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In Navab N, Hornegger J, Wells W and Frangi A (eds), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015.* Lecture Notes in Computer Science, Vol. **9351**. Cham: Springer, pp. 234–241.

Ros L and Thomas F (2002) Overcoming superstrictness in line drawing interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 456–466.

Schkolne S, Pruett M and Schröder P (2001) Surface drawing: creating organic 3D shapes with the hand and tangible tools. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, March 31–April 5 in Seattle, Washington, USA. New York: ACM, pp. 261–268.

Schlösser C (2018) Towards concise gaze sharing. *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications (ETRA '18)*, Warsaw, Poland, June 14–17, 2018. New York, NY, USA: ACM, Article 78, p. 3, doi:10.1145/3204493.3207416.

Schütze M, Sachse P and Römer A (2003) Support value of sketching in the design process. *Research in Engineering Design* **14**, 89–97.

Schweikardt E and Gross MD (2000) Digital clay: deriving digital models from freehand sketches. *Automation in Construction* **9**, 107–115.

Shilane P, Min P, Kazhdan M and Funkhouser T (2004) The Princeton shape benchmark. *Proceedings on Shape Modelling Applications*, 7–9 June 2004, Genova, Italy, pp. 167–178.

Simo-Serra E, Iizuka S, Sasaki K and Ishikawa H (2016) Learning to simplify: fully convolutional networks for rough sketch cleanup. *ACM Transactions on Graphics* **35**, Article 121.

Skwarek M (2013) CreatAR: Augmented reality app. *ACM SIGGRAPH 2013 Mobile*, Anaheim, California, July 21–25, 2013, pp. 21:1–21:1.

Soares JVB, Leandro JJG, Cesar RM, Jelinek HF and Cree MJ (2006) Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. *IEEE Transactions on Medical Imaging* **25**, 1214–1222.

Song J, Su F, Chen J, Tai C and Cai S (2002) An object-oriented progressive-simplification-based vectorization system for engineering drawings: model, algorithm, and performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 1048–1060.

Steger C (1998) An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 113–125.

Strasnick E, Holz C, Ofek E, Sinclair M and Benko H (2018) Haptic links: Bimanual haptics for virtual reality using variable stiffness actuation. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Montreal QC, Canada, April 21–26, 2018, Vol. **644**. New York: ACM, pp. 1–12.

Sudarsan R, Fenves SJ, Sriram RD and Wang F (2005) A product information modeling framework for product lifecycle management. *Computer-Aided Design* **37**, 1399–1411.

Szeliski R (2010) *Computer Vision: Algorithms and Applications*. London: Springer-Verlag.

Tombre K, Ah-Soon C, Dosch P, Masini G and Tabbone S (2000) Stable and robust vectorization: how to make the right choices. In Chhabra AK and Dori D (eds), *Graphics Recognition Recent Advances. GREC 1999.* Lecture Notes in Computer Science, Vol. **1941**. Berlin, Heidelberg: Springer, pp. 3–18.

Tversky B and Suwa M (2009) Thinking with sketches. In Markman AB and Wood KL (eds), *Tools for Innovation: The Science behind the Practical Methods That Drive New Ideas*. New York, NY, USA: Oxford University Press, pp. 75–84.

Varley P (2009) The use of neighbourhood matching in constructing hidden object topology. *Proceedings of the World Congress on Engineering*, July 1–3, 2009, London, U.K., Vol. **1**.

Wacker P, Voelker S, Wagner A and Borchers J (2018) Physical guides: an analysis of 3D sketching performance on physical objects in augmented reality. *Proceedings of the Symposium on Spatial User Interaction - SUI '18*, Berlin, Germany, October 13–14, 2018. New York, NY, USA: ACM Press, pp. 25–35, doi:10.1145/3267782.3267788.

Waltz D (1975) Understanding line drawings of scenes with shadows. In Winston PH (ed.), *The Psychology of Computer Vision*. McGraw-Hill, pp. 19–91, Chap. 2.

Wang F, Kang L and Li Y (2015) Sketch-based 3D shape retrieval using convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 7–12 June 2015, Boston, MA, USA.

Wiese E, Israel JH, Meyer A and Bongartz S (2010) Investigating the learnability of immersive free-hand sketching. *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, Annecy, France, June 07–10, 2010. Eurographics Association, pp. 135–142.

Xie J, Dai G, Zhu F and Fang Y (2017) Learning barycentric representations of 3D shapes for sketch-based 3D shape retrieval. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 3615–3623.

Xu B, Chang W, Sheffer A, Bousseau A, McCrae J and Singh K (2014) True2form: 3D curve networks from 2D sketches via selective regularization. *ACM Transactions on Graphics* **33**, 131.1–131.13

Xu Y, Xie Z, Feng Y and Chen Z (2018) Road extraction from high-resolution remote sensing imagery using deep learning. *Remote Sensing* **10**, 1461–1476

Ye M and Zhou S (2019) DeepShapeSketch: Generating hand drawing sketches from 3D objects. International Joint Conference on Neural Networks, pg 1-8, 14-19 July 2019, Budapest, Hungary.

Yoon SM, Scherer M, Schreck T and Kuijper A (2010) Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. *Proceedings of the 18th ACM International Conference on Multimedia*, Firenze, Italy, October 25–29, 2010. New York: ACM, pp. 193–200.

Zeleznik RC, Herndon KP and Hughes JF (2006) SKETCH: An interface for sketching 3D scenes. *ACM SIGGRAPH 2006 Courses*, Boston, Massachusetts, July 30–August 03, 2006. New York: ACM.

Zhu Fan, Xie Jin and Fang Yi (2016) Learning cross-domain neural networks for sketch-based 3D shape retrieval. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16),* Phoenix, Arizona, USA, February 12–17, 2016, pp. 3683–3689.

Alexandra Bonnici graduated as an Electrical and Electronics Engineer from the University of Malta and obtained her MPhil and PhD also from the University of Malta, working in the area of sketch recognition and interpretation. Her research interests include the application of image and signal processing as well as machine learning algorithms to areas of sketch interpretation and music analysis. Alexandra is a member of the IEEE, the ACM, and the Eurographics.

Alican Akman received his BSc degree in Electrical and Electronics Engineering with a double major in Physics from Bogazici University in 2017, both with high honors. He was also an exchange student in the Electrical and Computer Engineering Department at the University of Illinois at Urbana-Champaign in 2016. He is currently a research and teaching assistant at Koç University IUI Laboratory as an MSc student as well as a data scientist at Turkcell. His research interests include explainable artificial intelligence, machine learning, big data analytics, and computer vision. His latest research is focused on developing machine learning methods for novel applications such as animation generation and 3D model sketching.

Gabriel Calleja received the BEng (Hons) in Electrical and Electronics Engineering and MSc in Signals, Systems and Control degrees from the University of Malta in 2017 and 2018, respectively. He is currently working as a Systems and Network Engineer at ICT Solutions Ltd Malta designing and maintaining various networks and systems that meet client's needs.

Kenneth P. Camilleri graduated in Electrical Engineering from the University of Malta in 1991 and received the MSc in Signal Processing & Machine Intelligence and PhD in Image Processing & Pattern Recognition in 1994 and 1999, respectively, from the University of Surrey, Guildford, UK. He is currently the Head of Department of Systems and Control Engineering in the Faculty of Engineering and Director of the Centre for Biomedical Cybernetics at the University of Malta. Professor Camilleri has a long track record in research in the area of computer vision and in the analysis of brain signals, publishing over 120 research papers in peer-reviewed international journals and conferences. His present research work, together with his PhD students and research colleagues spans the areas of computer vision for eye-gaze tracking, various applications of thermal image analysis (thermography), sketch and drawing understanding, scene understanding augmented by natural language, ultrasound signal and image processing, electroencephalography (EEG) signal processing, brain–computer interfacing, multi-sensorial gait and motion analysis, stroke patient rehabilitation monitoring and neurorehabilitation.

Patrick Fehling is a student of Applied Computer Sciences at the HTW Berlin and participates in the research project Virtual Environment for Teamwork and ad-hoc Collaboration between Companies and heterogeneous User Groups (VENTUS). His work focuses on the Leapmotion controller and novel 3D user input techniques as alternatives to traditional handheld interaction devices. Besides, he works for BOC Information Technologies Consulting GmbH as a student employee in the development team of Business Process Modeling and Enterprise Architecture Management tools.

Alfredo Ferreira is an Assistant Professor at the Instituto Superior Técnico, University of Lisbon. He received his PhD (2009), MSc (2005), and BS (2002) degrees in Information Systems and Computer Science from the Technical University of Lisbon. He is simultaneously a researcher of the Visualization and Intelligent Multimodal Interfaces Group at INESC-ID Lisboa. He works on multimodal and natural interfaces, virtual and augmented reality, and 3D object retrieval. He has been involved in several national and European (SmartSketches, EuroTooling21, MAXIMUS) projects focusing on these topics. Alfredo Ferreira is also a member of ACM, IEEE, and Eurographics.

Florian Hermuth is a Master student of Applied Computer Science at the University of Applied Sciences HTW Berlin. He also works as a research assistant for the VENTUS research project, together with Prof. J.H. Isreal. Florian received a Bachelor of Science in Applied Computer Science at the University of Applied Sciences HTW Berlin, with a focus on multimedia. During his studies, Florian worked on different projects about virtual sketching in 2D and 3D.

Johann Habakuk Israel is a Professor of Applied Computer Sciences at the HTW Berlin and a member of the special interest group Be-greifbare Interaktion (tangible interaction) of the German Informatics Society. His main research areas are human–computer interaction, virtual reality, 3D user interfaces, immersive sketching and modeling, tangible interaction, and transdisciplinary research.

Tom Landwehr is a Bachelor student of Applied Computer Sciences at the HTW Berlin and was an intern with the research project Virtual Environment for Teamwork and ad-hoc Collaboration between Companies and heterogeneous User Groups (VENTUS) focusing on components of the user interface, as well as their implementation in a virtual reality environment.

Juncheng Liu received a Bachelor degree in Software Engineering from the Dalian University of Technology, China, in 2013. He is currently a PhD candidate at the Institute of Computer Science and Technology, Peking University, China. His main research interests include 3D processing, pattern recognition, and machine learning.

Natasha Mary Jane Padfield graduated from the University of Malta with a Master's degree in Signals, Systems and Control in 2018. Her main area of interest is signal processing, and she is currently reading for a PhD at the University of Strathclyde, with a research focus on brain controlled human–computer interfaces.

T. Metin Sezgin received a B.S. in Computer Science with summa-cum laude with Honors distinctions from Syracuse University in 1999, and the MS and PhD degrees from the Massachusetts Institute of Technology, in 2001 and 2006, respectively. He subsequently joined the University of Cambridge as a postdoctoral research associate, and held a visiting researcher position with Harvard University in 2010. He is currently an associate professor at Koç University, Istanbul, and directs the Intelligent User Interfaces (IUI) Laboratory. His research interests include intelligent human–computer interfaces and HCI applications of machine learning. He is particularly interested in applications of these technologies in building intelligent pen-based interfaces.

Paul L. Rosin is Professor at the School of Computer Science & Informatics, Cardiff University. Previous posts include lecturer at the Department of Information Systems and Computing, Brunel University London, UK, a research scientist at the Institute for Remote Sensing Applications, Joint Research Centre, Ispra, Italy, and lecturer at the Curtin University of Technology, Perth, Australia. His research interests are wide and include: low-level image processing, performance evaluation, shape analysis, facial analysis, medical image analysis, 3D mesh processing, cellular automata, and non-photorealistic rendering. Much of this work has been performed in multidisciplinary collaborations. In the area of non-photorealistic rendering, he has published many papers over the last decade, and he co-organized the "Vision meets Graphics" workshop in 2015 and 2017, the International Symposium on Computational Aesthetics in Graphics, Visualization in 2014 and 2015, and the workshop on Photographic Aesthetics and Non-Photorealistic Rendering in 2013. He was co-editor of the book "Image and Video-Based Artistic Stylisation" published by Springer in 2013. In the area of 3D object retrieval, in addition to his published papers, he co-organized the SHREC competition for "Canonical Forms for Non-Rigid 3D Shape Retrieval", as well as the SHREC competition for "Shape Retrieval of Non-Rigid 3D Human Models".