



# Understanding and improving the relation between interaction simplicity and expressivity for producing illustrations

Axel Antoine

## ► To cite this version:

Axel Antoine. Understanding and improving the relation between interaction simplicity and expressivity for producing illustrations. 2018, pp.1-6. hal-01900071

HAL Id: hal-01900071

<https://hal.science/hal-01900071>

Submitted on 22 Oct 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Understanding and improving the relation between interaction simplicity and expressivity for producing illustrations

**Axel Antoine**

Université de Lille, Inria, UMR  
9189 - CRISTAL, Lille, France  
axel.antoine@univ-lille.fr

**Résumé**

Cette thèse s'intéresse à la relation entre expressivité et simplicité dans un contexte de production d'illustrations. Plus précisément, elle se concentre sur le design de systèmes et techniques permettant aux utilisateurs novices de facilement et rapidement produire des illustrations avec un haut niveau d'expression tout en permettant aux utilisateurs experts d'augmenter leur performance.

**Mots Clés**

expressivité; illustrations; workflows; performance

**Abstract**

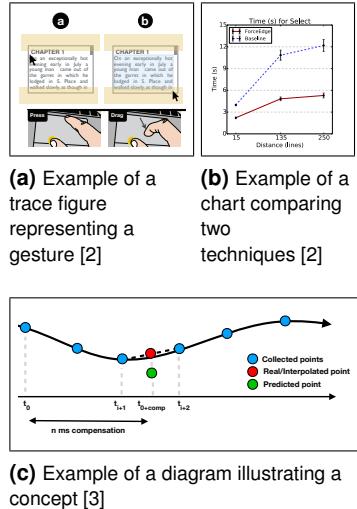
This thesis is interested in the relationship between expressivity and simplicity in the context of illustrations production. More precisely, it focuses on the design of systems and workflows allowing novice users to produce illustrations in an easier and faster way with a high level of expressivity while still allowing experts users to boost their production performance.

**Author Keywords**

expressivity; illustrations; workflows; performance

**ACM Classification Keywords**

[H.5.2]: User Interfaces: Input devices and strategies



**Figure 1: Examples of illustrations used in papers**

## Topic of the Ph.D.

In the HCI community, illustrations are often used as visual supports to describe novel interaction techniques or systems, both statically (through trace figures, diagrams, slides), dynamically (animated or interactive figures, videos), or for data visualization through charts. Figure 1 shows three examples of illustrations: a trace figure, a chart and a diagram. If some types of illustrations can be produced directly from the real world (e.g. photos, videos) using, for example, a camera, some others need to be produced manually. This can take a significant amount of time depending on user skills with graphics editing softwares, especially for trace figures.

The main objective of this thesis is to study the relationship between expressivity and simplicity in the context of illustration production. More precisely, it focuses on the design of systems and workflows allowing novice users to produce illustrations in an easier and faster way with a high level of expressivity and, while still allowing experts users to build on top of the produced illustrations.

## Related work on the trace figures production

As a first step of this thesis, we were interested in the process of trace figures production in the HCI community.

Trace figure illustrations are defined as graphical illustrations of the most essential features of a scene using only contours/outlines of objects, people and the represented environment. We studied existing systems designed to produce trace figures and, from the set we chose, we classified them in different categories, depending on what their associated workflow is based on.

*Workflows using sensors.* To facilitate the production of choreographic movements illustrations, Chi et al. designed Demodraw [5]. Their system allows the user to define a

succession of body poses using a Microsoft Kinect. Then their system generates a 2D render of the 3D avatar manipulated with the sensor but only proposes to produce body gestures (objects such as devices can not be produced with DemoDraw) and in a non vector-based output.

*Workflows from sketches.* In a similar paradigm of fast prototyping illustrations, Kim et al. designed SketchStudio [8], a tool to define animated scenarios by controlling 2D elements in a spatial and temporal space. These scenarios are composed of 2D sketches manually drawn and predefined avatars navigating between and interacting with the 2D sketches. Unfortunately, this tool still heavily relies on sketches, thus requiring users to have high sketching skills, making the tool limited to *expert* users with such sketching skills. Moreover, similarly to DemoDraw, it produces full-body illustrations only, making it unadapted to non-full-body illustrations (typically, a hand interacting with a smartphone). Finally, the non-use of a complete 3D space is perceived as a limitation from some participants who had difficulties with perspective perception.

*Workflows using other medias.* Another technique to produce a trace figure is to automatically compute vector-based version of a photography. The vector-based version is computed using automatic vector graphics algorithms (e.g. Vectorizer in Adobe Illustrator) but involves the vectorization of the whole image. In fact, automatic algorithms detect similar color ranges in the image to create vector-based fills. This results in a tedious post-process to delete all the unwanted created fills to obtain the essential part of the trace figure. Edge detection algorithms may be useful in that case to only extract the outlines of the trace figures and avoid the manual post-process step [4], but this technique results in a pixel-based trace figures hard to edit afterwards.

**Taxonomy of trace figures****Categories**

- Demonstration of gestures
- Overview of system setup or assembly
- Interaction sequences
- Design space
- Other

**Main Characteristics**

- Devices and objects
- Environment
- Colors
- Annotations
- Notion of time

**Progression and results**

In this section, we present the taxonomy of trace figures we made and the current workflow generally used to produce these trace figures and the associated limitations. Then, we present Esquisse, an alternative to this workflow that can be used by anyone to produce trace figures.

*Taxonomy of trace figures*

We investigated all the illustrations used in the last 3 years proceedings from the ACM UIST conference. We found a use of 124 trace figures out of 222 accepted papers (which represents 29% of the paper) from which we identified several categories and characteristics (details in the sidebar).

*Current practices for producing trace figures*

We contacted 12 HCI researchers that we identified as using trace figures in their research papers. They seem to adopt similar workflows (with minor variations), composed of the 5 steps described as follows:

**Step 1:** Staging the situation to illustrate in the real world and take a photo of it.

**Step 2:** Transfer the photo to a computer and open it as a background layer in an graphics editing software.

**Step 3:** Sketch traces on top of the photo following the contours of the different elements of the photo.

**Step 4:** Adding details such as arrows, text or any other effect wanted to illustrate the discourse.

**Step 5:** Export the trace figure to include it in the support.

While almost all the researchers found this process relatively easy to apply, they also agreed on its longness and tediousness. Moreover, some insisted on the fact when they are not satisfied with the result, or want to change the point of view of the illustration, they sometimes can manually edit the outlines, but they more often have to "*take a novel picture and start the whole process again*".

*Esquisse*

We designed Esquisse, a novel tool that uses simple 3D staging to facilitate the creation of trace figures. We avoid direct vector graphics drawing by using adapted interaction techniques to manipulate 3D objects in a 3D scene that will later be rendered as a vector graphic image. Because full 3D manipulation can sometimes be problematic, Esquisse comes with a set of dedicated interaction techniques. It also offers a innovative render algorithm (*e.g.* render a stroboscopic effect) based on existing techniques. Esquisse was implemented as an add-on for Blender, a free and cross-platform software widely used to model, animate and render 3D scenes, only as a 3D engine software.

*Workflow*

In comparison with the previous workflow to create trace figures, we describe hereafter the Esquisse workflow:

**Step 1:** Creating a 3D scene using one of the pre-existing templates (*i.e.* predefined scenes that we designed from the taxonomy) or starts from scratch.

**Step 2:** Positioning the objects using the controls provided by Esquisse or use the default Blender 3-axis manipulator.

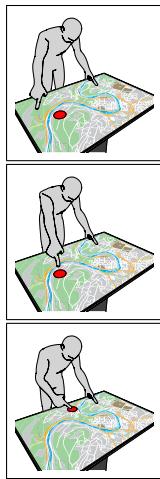
**Step 3:** Positioning the camera to get the desired point of view.

**Step 4:** Rendering the scene to create the trace figure as a vector-based SVG file.

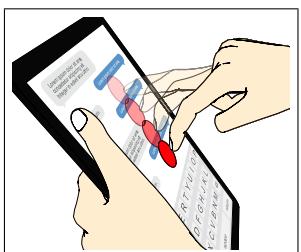
*Facilitate the 3D manipulation*

Simple objects can be positioned in the 3D scene easily using the 3-axis manipulator already available in Blender for translating, rotating and scaling the objects. However, when it comes to complex objects like a person's hand or body, manipulating the pose of the armature implies manipulating all the sub-objects (bones) in the 3D space.

We then created an anchoring system based on the inverse kinematics system offered by Blender. Anchors are points



**Figure 2:** From top to bottom. a) A red anchor is added to the screen. b) The index of the right hand is linked to the anchor. c) The anchor is moved and the arm follows.



**Figure 3:** Example of a stroboscopic effect generated by Esquisse.

on any object that are used to create constraints on fingers. When an anchor is moved, the linked finger follows and the hand posture adapts itself (See steps of the Figure 2). This system allows to easily define complex poses of a hand but the manipulation of non-anchored fingers remains difficult [10].

We added two specific controls for modifying hand poses. First, inspired by Achibet et al. [1], we added slider controls to manipulate the flexion of each finger of a hand model. Second, we built a module using a hand tracking camera (LeapMotion) to modify the pose and orientation of the 3D hand model by demonstration.

#### *Stroboscopic Effect*

As a way to express the movements [9], Esquisse allows to produce stroboscopic effects in the render by defining different keyframes in the scene as Chi et al. [5] did for a 3D body (see an example in Figure 3).

#### *Rendering process*

The staged 3D scene is then rendered as a SVG vector file using existing techniques [6, 7]. We extract the visible and hidden outlines of each objects in the scene and write them to a SVG file as filled polygons.

#### *Evaluation*

To evaluate Esquisse, we asked 8 participants to illustrate state of the art interactions with Esquisse. Results show that participants quickly managed to use Esquisse to produce satisfying figures. They also found it "amusing and working well", "interactive, cool and funny" or even said that Esquisse allowed them to "produce these figures [they] would not be able [to make] without it".

## Future work and directions

After Esquisse, this thesis could focus on the slideshows production, especially, on how to help animations staging in slides.

Independently of the application used (e.g. Apple Keynote, Microsoft Powerpoint, Google Slides), staging animations is a complex and tedious task. Indeed, it requires to control the variation of positions/shapes of the animated objects by manipulating the 2D space (*What are the trajectories of the animated objects at time t?*), the duration and type of the variation (*How do the objects move ? e.g. constant speed, acceleration, ease in/ease out*) and the synchronization between the animation of different objects (*Which animation starts when?*).

Esquisse is designed to help novice users to produce contents that they would not be able to produce without these systems while still allowing expert users to boost their performance. The complexity (in terms of tediousness and time consumption) of the photo tracing task was replaced by a simpler and faster 3D staging task when mastered.

Esquisse introduces an additional dimension as a way to enhance users capabilities. Indeed, we show that using a 3D scene can be useful to simplify the production of 2D trace figures. In the case of animations staging, time, spatial and event dimensions could help this production. Finally, the use of additional dimensions seems to be at first one potential solution to simplify production workflows and increase the expressivity for the production of these two types of illustrations. First, it could be interesting to know how to generalize this principle to different types of illustrations. Then, if a user wants to produce different types of illustrations for a same content (e.g. transform an animation into a trace figure), how to facilitate the transition to avoid the user to start over?

## Bibliographie

- [1] Merwan Achibet, Géry Casiez, Anatole Lécuyer, and Maud Marchal. 2015. THING: Introducing a Tablet-based Interaction Technique for Controlling 3D Hand Models. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 317–326. DOI: <http://dx.doi.org/10.1145/2702123.2702158>
- [2] Axel Antoine, Sylvain Malacria, and Géry Casiez. 2017. ForceEdge: Controlling Autoscroll on Both Desktop and Mobile Computers Using the Force. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3281–3292. DOI: <http://dx.doi.org/10.1145/3025453.3025605>
- [3] Axel Antoine, Sylvain Malacria, and Géry Casiez. 2018. Using High Frequency Accelerometer and Mouse to Compensate for End-to-end Latency in Indirect Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 609, 11 pages. DOI: <http://dx.doi.org/10.1145/3173574.3174183>
- [4] John Canny. 1987. A computational approach to edge detection. In *Readings in Computer Vision*. Elsevier, 184–203.
- [5] Pei-Yu (Peggy) Chi, Daniel Vogel, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2016. Authoring Illustrations of Human Movements by Iterative Physi-
- cal Demonstration. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 809–820. DOI: <http://dx.doi.org/10.1145/2984511.2984559>
- [6] Elmar Eisemann, Holger Winnemöller, John C Hart, and David Salesin. 2008. Stylized vector art from 3D models with region support. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1199–1207.
- [7] Stéphane Grabli, Emmanuel Turquin, Frédéric Durand, and François X. Sillion. 2004. Programmable Style for NPR Line Drawing. In *Rendering Techniques 2004 (Eurographics Symposium on Rendering)*. ACM Press, Norrköping, Sweden. <https://hal.inria.fr/inria-00510169>
- [8] Han-Jong Kim, Chang Min Kim, and Tek-Jin Nam. 2018. SketchStudio: Experience Prototyping with 2.5-Dimensional Animated Design Scenarios. In *Proceedings of the 2018 Designing Interactive Systems Conference (DIS '18)*. ACM, New York, NY, USA, 831–843. DOI: <http://dx.doi.org/10.1145/3196709.3196736>
- [9] Erin McAweeney, Haihua Zhang, and Michael Nebelung. 2018. User-Driven Design Principles for Gesture Representations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 547, 13 pages. DOI: <http://dx.doi.org/10.1145/3173574.3174121>
- [10] Ying Wu and Thomas S Huang. 2001. Hand modeling, analysis and recognition. *IEEE Signal Processing Magazine* 18, 3 (2001), 51–60.

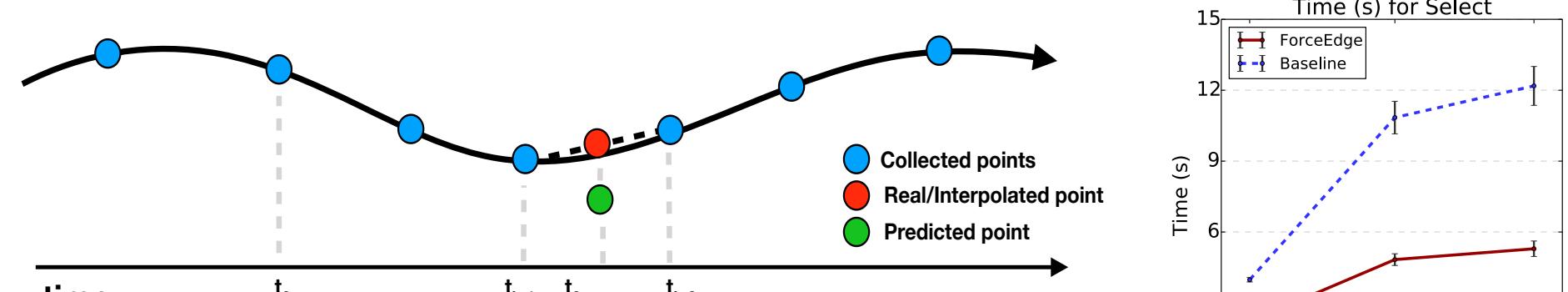
# Comprendre et améliorer la relation entre simplicité d'interaction et expressivité pour la production d'illustrations

Axel Antoine (1,2), Sylvain Malacria (2,1), Géry Casiez (1,2)

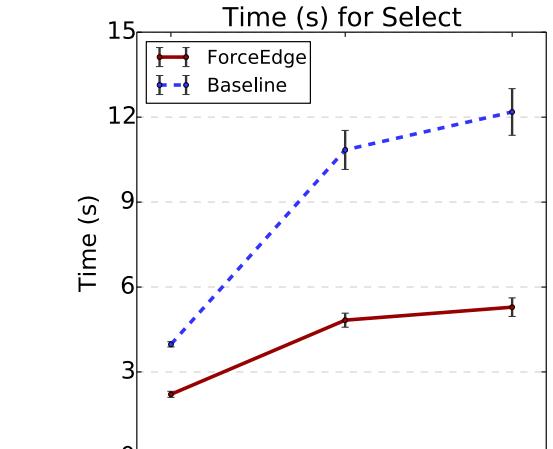
(1) Université de Lille, UMR 9189 CRIStAL (2) Inria Lille Nord Europe

## Que sont les illustrations ?

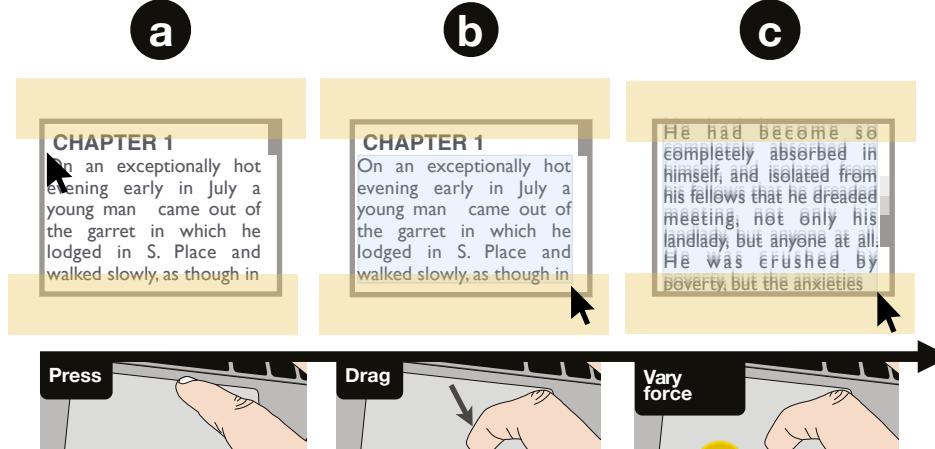
Dans la communauté IHM, les illustrations sont utilisées à des fins de support visuel pour décrire des techniques d'interaction, des systèmes ou encore des données. Il en existe plusieurs types : tracés simples, diagrammes, diaporamas, animations 2D, figures intéractions, vidéos, etc..



a) Un diagramme



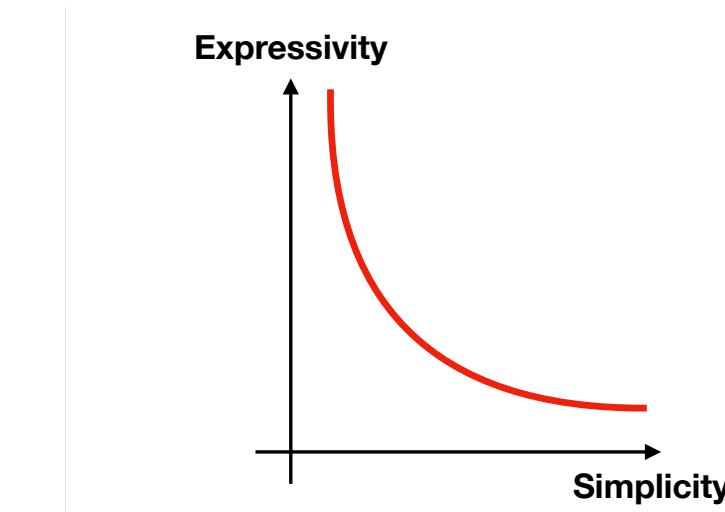
b) Un graphique



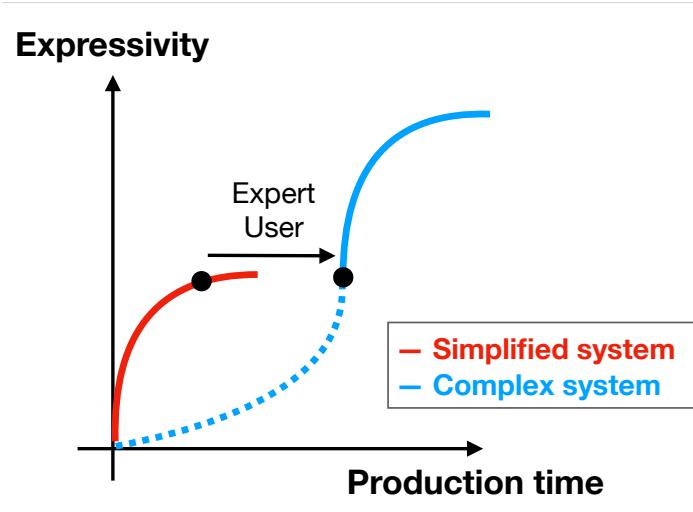
c) Une figure vectorielle

## Objectif de la thèse

Etudier et concevoir des systèmes et techniques permettant aux novices de produire facilement et rapidement des illustrations avec un haut niveau d'expressivité tout en permettant aux experts d'augmenter leur performance et de construire par dessus les illustrations produites.



Interaction entre expressivité et simplicité



Rouge) Courbe de production pour un novice.  
Rouge+Bleu) Courbe de production pour un expert

## Esquisse : Faciliter la mise en scène 3D pour faciliter la création de figures vectorielles

### Esquisse

Esquisse est un outils permettant de générer des images vectorielles à partir d'une scène 3D. Pour simplifier les manipulations 3D (notamment les mains et les personnages), Esquisse propose plusieurs interactions.

### Effet stroboscopique

Afin de pouvoir exprimer le mouvement dans les figures, Esquisse permet de définir des keyframes sur les objets de la scène. Les keyframes sont rendues avec des niveaux de transparence différents.

### Système d'ancrage

Les ancrés sont des points pouvant être placés sur tous les objets afin de définir des contraintes sur les doigts. Quand une ancre est déplacée, le système cinématique intégré à Esquisse adapte automatiquement la posture de la main.

### Sliders et LeapMotion

Pour modifier la posture des mains sans les ancrés, des sliders permettent de définir l'ouverture de chaque doigt. Une intégration du LeapMotion est aussi fournie avec Esquisse.

### Rendu vectoriel

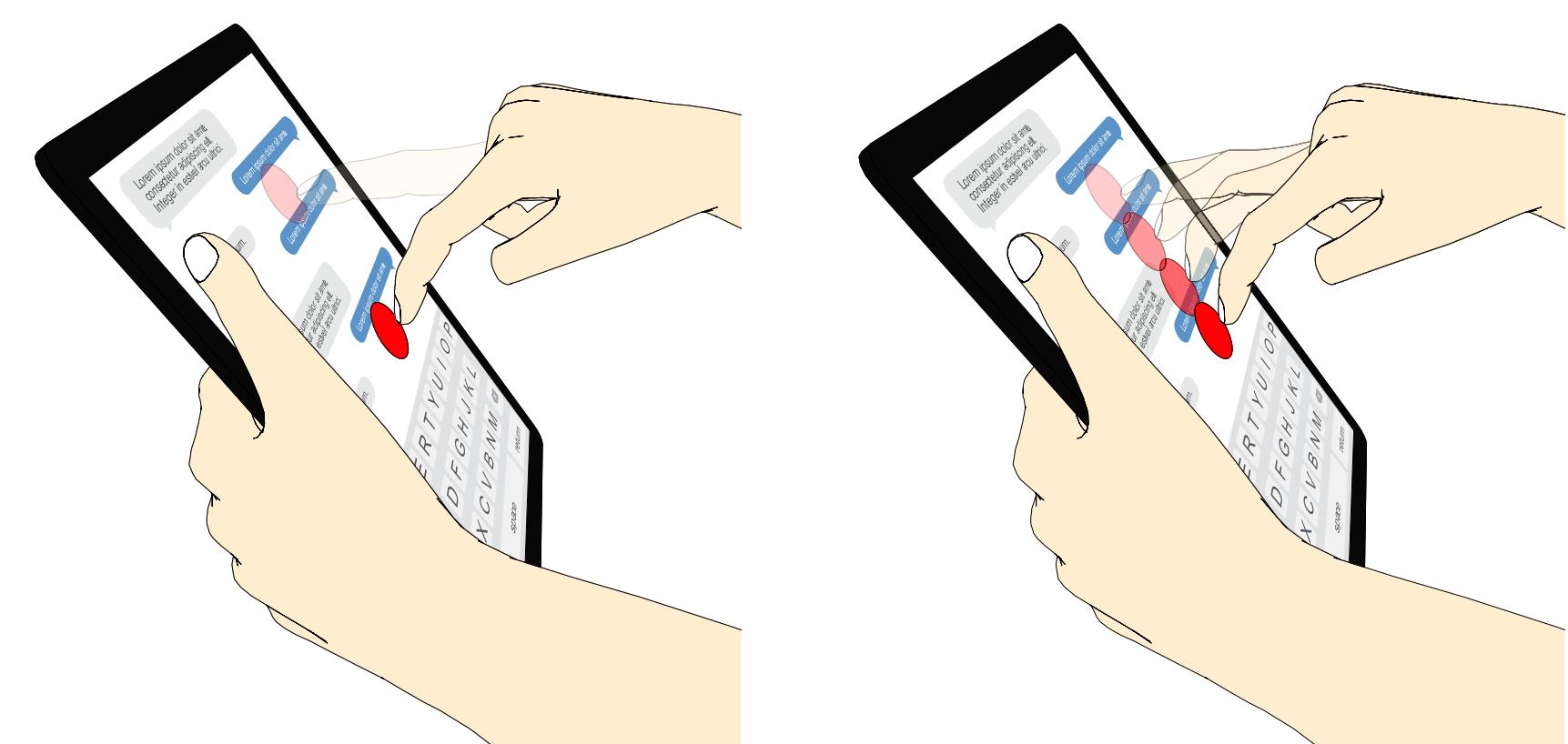
Esquisse utilise des techniques existantes de rendu graphique permettant d'extraire les contours des objets présents dans la scène afin de les convertir en polygones à dessiner dans le fichier SVG.

### Evaluation et résultats

8 participants ont illustré 4 techniques d'interactions de la littérature. Les participants ont su rapidement apprendre et utiliser Esquisse pour produire des figures vectorielles.



a) Ajout d'une ancre. b) L'index droit est lié à l'ancre. c) Quand l'ancre est déplacée, le bras s'adapte automatiquement



a) Effet stroboscopique de l'index sans interpolation  
b) Effet stroboscopique de l'index avec interpolation

## Directions futures

### Faciliter la création d'animations 2D

Concevoir un workflow permettant de savoir et définir la trajectoire des objets, la manière dont ils se déplacent et la synchronisation entre les différents objets animés

## Perspectives

### Une dimension additionnelle qui simplifie ?

Esquisse a montré que la 3D pouvait aider à produire des figures 2D. Des dimensions temporelles, spatiales et événementielles distinctes peuvent-elles aider à créer des animations ?

### Transfert de production

Comment assurer le transfert d'une production entre deux systèmes ? Comment passer d'un type d'illustration à un autre sans tout recommencer ?