A Sketch-based Interface to Script Comics Reading

J. Raulet[*]
L.I.A.S.D - University Paris 8

V. Boyer[†]
L.I.A.S.D - University Paris 8

1 Introduction

The advent of smartphones and the emergence of tablet personal computers have led to modify the contents delivery device. Touchscreen devices and their end-user accessibility (user interface, add/remove applications, system upgrade) emphasize their popularity and are largely used to display media information contents (text, audio, still images, . . .) and multimedia (video games, . . .). Even if the screen resolutions are smaller than those of image acquisition systems, the user can have a global view of the image or the text and focus easily on a detail through the tactile user interface (zoom in/out). Remark that the book reading is structured and linear (from left to right and top to bottom for occidental reading for example), while the image viewing is non-linear and often structured (viewer can both focus on particular detail or have a global view and its viewing is almost unpredictible). For these cases, the scenarization of the book reading or image viewing is very simple (respectively one main predictible displacement or a free manipulation).

Some media, like comics are non-linear structures and can be hard to read sometimes (box sizes, overlapping, text, successive word balloons. . .). Comics reading on smartphones or tablet devices remains a problem because a global view of a comics page does not provide any suitable sufficient resolution. Moreover comics reading is non-linear and so there is not such predictible displacements. For these reasons the scenarization of the comics reading should be created to let the viewer concentrate upon the story and never suffer from any searches of successive story's elements that radically modify its perception.

At the moment the scenarization is realized, in the best case, manually in the industry through image manipulation tools, produces very limited results and is very expensive (i.e even if these tools permit to crop, resize an image. . . , they are not dedicated to create script reading).

As comics is a graphic media, we think that the scenarization process should be realized interactively by the comics creator or at least by a graphist or a designer. Sketch-based interface, as a powerfull way to express something, seems to be the most suitable solution to provide any scenarizations of the comics reading. As shown by Gross[Gross], kinds of systems exist and can be classified in terms of descriptiveness and generality. As CrossY[Apitz and Guimbretiere] we want to provide a system in which pen based interaction is used to point, edit or interact with. Igarashi et al.[Igarashi et al.] mentioned that one of key direction consists in to reinvent applications to leverage pen input's intrinsic capability for rapid, direct, modeless 2D expression.

We propose a sketch-based tool to script comics reading for different devices with their own screen size and resolution. Our tool is able to create global view, to extract images and boxes, to produce various transitions between images (fades, movements, accumulations, . . .), to display each image on the device screen with a displacement in the image. . . Moreover, we provide a viewer for each target device able to read the produced script and allowing user interactions. Note that more than a comics reading scenarization, it can be used to script any non-linear structured media.

2 Model Description

Starting with comics pages (see figure 1), our sketch-based model is designed to produce a description file and a set of images that will be used with our viewer to display the comics on devices. Our system is based on a low level recognition of simple glyphs. It allows the designer to create a scenarization for comics reading on different display devices considering three main problems (from top to bottom approach):

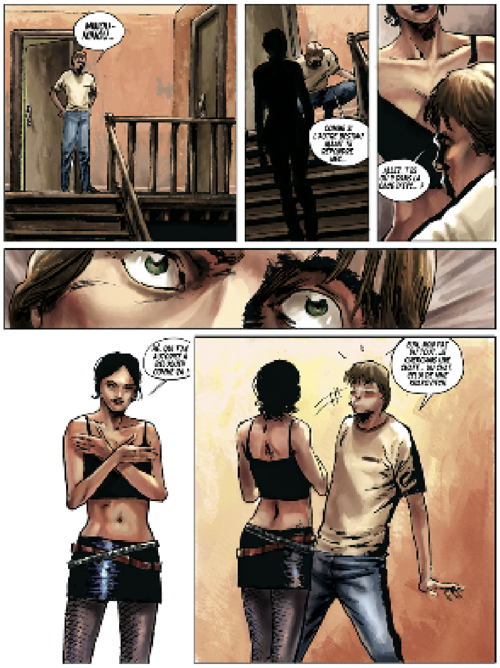


Figure 1: *Fate Spinner by A. Dollen and E. Heidinger.*

1. Allow the user to extract the successive images of the comics, hereafter called image decomposition;
2. Allow and help the user to specify the properties of each displayed image (when and how it appears and disappears, reading direction), hereafter called image properties;
3. Automatic generation of the description file.

Therefore comics reading is composed by many **screens**. Each of them is composed by one or more **images** often included in **boxes**. The **timing** depends largely on the screen composition and the images reading considering **text position**, **balloons** and zoom in/out. Moreover it can be reinforced by the screen transitions.

2.1 Image decomposition

To guide the designer to create the different screens, as a preliminary step we automatically extract boxes and images by detecting the background color used in the page. We find an image border by detecting color difference with the background. During this process, we create at the same time a polygonal bounding box. When

[*] e-mail:jraulet@ai.univ-paris8.fr

[†] e-mail:boyer@ai.univ-paris8.fr

two close neighbour bounding boxes have been created, we automatically merge them (see figure 2). Remark that we have automatically extracted both boxes and the image with the girl alone. Note that overlapping of images or boxes is not decomposed on multiple images and is therefore presented as one image. Then by pointing the images the user can specify the image sequence. In figure 2, from left to right, the user first points the 5th image and draw a line to the 6th one. Thereby, he specifies the first and second images of the sequence. Then he points the third (men's eyes) and the fourth image and finally points the two last images on the left. The final result is presented at the top of figure 3 which provides an easily way to create screens and transitions using these images.

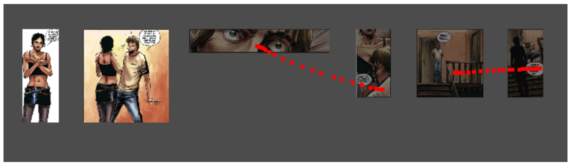
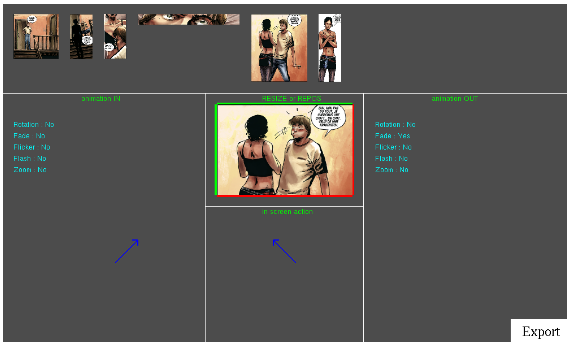


Figure 2: *Result of box and image detections.*



Rotation : No
Fade : No
Flicker : No
Flash : No
Zoom : No

Rotation : No
Fade : Yes
Flicker : No
Flash : No
Zoom : No

Figure 3: *Result of the screen composition step.*

2.2 Defining image properties

The figure 3 presents a global view of our system. The user can specify the different image properties (for the two first ones, he should first select the image on the top of figure 3):

- **image animations** both when it appears and disappears. This is realized by sketching respectively on the bottom left and right of our interface (see figure 3). Textual informations summarize the user interaction. We provide a various collection of possible animations defined for each of them by a glyph: fade (△); flash (+); zoom-in (□); zoom-out (◇); rotation both CW and CCW (◯); flicker (- - -); 8 directions (←, ..., ↖).
- **image size and position** when it appears and **reading direction**. The center of figure 3 is used by the user to sketch this property. The part "resize and repos" allows the user to specify the image size and position to be displayed when it appears. Note that green line depicts a border of the original image while red line mentioned that the image is cropped. On this example (see figure 3), the top left corner of the image will be displayed when it appears. In the part "screen action", the user specifies the reading direction. Based on comics structure analysis, we provide 8 reading directions (←,. . . , ↖) and any zigzags with only two parrallel lines (N, Z, . . .).
- **image combination** depicting accumulation of multiple images and deciding when they disappear. The user can encircle consecutive images to provide image accumulation on the screen. This is achieved by a sketching on the top of our system. Therefore, image transitions defined for disparition are performed when the first image which is not in the accumulation appears.

2.3 Export files and device viewer

Each generated images is stored and a description file (xml) containing properties of each image is generated. We provide a viewer for each device. This viewer is able to read the xml file and automatically displays the images, applies image properties (i.e computes and generates the transitions, adapts size and position, applies reading direction and provides images combination). Note that a user interface is always proposed to start and stop the comics reading; classical manipalutation like zoom in/out, displacements can be achieved by the user any time.

3 Results and Discussion

A collection of examples are given on our web site (http://www.ai.univ-paris8.fr/∼jraulet/SigAsia2011). For each of them, we give a video of our sketch-based interface and the result produced on a device. As one can see, for a complete comics page, we are able to generate a result in at most five minutes. Remark that with other solutions used in the industry, the user treats a comics page at least in one hour. The example used in this paper (see figure 1) was generated in two hours per page through an industrial process but sounds effects and musics were added. Moreover we provide also a result more intuitively and more suitable for the viewer (i.e he focusses only on the story and never suffers from any searches of successive story's elements).

4 Conclusion

We have proposed a sketch-based tool to script comics reading for different devices with their own screen size and resolution. The user sketches easily all image properties. Our tool provides a complete solution to script comics reading in a few minutes per pages compared to at least few hours with other solutions. In future works, we plan to manage text positions and balloons to improve the comics reading. Moreover we desire to detect image caracteristics to propose automatically reading directions to the user. Finally, our tool is designed to script any non-linear structured media and for example one can easily imagine to create efficient slideshow with this tool.

Acknowledgment
This work was supported by Tekneo SAS (producer of entertaining applications (video games), as well as serious applications (Serious games)). Jeremy Raulet was a recipient of a CIFRE fellowship.

References

APITZ, G., AND GUIMBRETIERE, F. Crossy: a crossing-based drawing application. In *Proceedings of the Siggraph 2005*, ACM Press / ACM Siggraph, 930.

GROSS, M. D. Visual languages and visual thinking: Sketch based interaction and modeling. In *Proceedings of Workshop on Sketch-Based Interfaces and Modeling 2009*, ACM Press / ACM SIGGRAPH.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. Teddy: A sketching interface for 3d freeform design. In *Proceedings of the Siggraph 1999*, ACM Press / ACM Siggraph, 409–416.