

Multi-Finger Interactions with Papers on Augmented Tabletops

Son Do-Lenh, Frédéric Kaplan, Akshit Sharma, Pierre Dillenbourg

Ecole Polytechnique Federale de Lausanne (EPFL)

1015 Lausanne, Switzerland

{son.dolenh, frederic.kaplan, pierre.dillenbourg}@epfl.ch, akshit@cse.iitkgp.ernet.in

ABSTRACT

Although many augmented tabletop systems have shown the potential and usability of finger-based interactions and paper-based interfaces, they have mainly dealt with each of them separately. In this paper, we introduce a novel method aimed to improve human natural interactions on augmented tabletop systems, which enables multiple users to use both fingertips and physical papers as mediums for interaction. This method uses computer vision techniques to detect multi-fingertips both over and touching the surface in real-time regardless of their orientations. Fingertip and touch positions would then be used in combination with paper tracking to provide a richer set of interaction gestures that the users can perform in collaborative scenarios.

INTRODUCTION

Augmented tabletops, table surfaces that work both as an input device and a display, offer potential to bridge the gap between the digital and the physical world. Since hand and finger-based interactions provide very common and natural ways to interact with the computer, they draw attention from many studies in augmented tabletop research such as [4, 7, 12, 16, 24]. The tabletop nature of these systems also opens up the ability of intuitive manipulations with physical objects [21] such as paper. Paper is light, portable, tangible and inexpensive. This is why many researchers have focused on re-introducing paper in the digital world [1–3, 13, 14, 19, 26].

However, it is surprising to note that most of the existing systems try to present finger- or paper-based interface separately. Digital Desk [22], an early computer vision-based desk consisting of a projector and a camera, demonstrated finger interactions, allowing users to use their fingers to manipulate paper and digital objects. There are otherwise few studies that explore the potential of their combination. We believe that both finger- and paper-based interactions should be supported, and cross references should be allowed between the two, i.e. doing some gestures with hands has implications

on the way papers react and vice versa. Migrating the simplicity of tangible paper manipulations and the naturalness of finger-based interactions may allow greater flexibility in the way information is manipulated, with a richer set of interaction techniques. We expect that this merged interface may also have some implications on how collaborative task accomplishments can be structured.

In this paper, we propose a new computer vision-based approach that enables a natural form of interaction on augmented tabletops using bare fingertips and physical papers in parallel. We illustrate the potential of this approach by presenting a basic set of gestures that can be supported and an user study comparing this approach with traditional computer user interface in a collaborative learning task.

RELATED WORK

Since our method was inspired by prior research in two areas, hand- and finger-based and paper-based interfaces, we will describe them respectively.

Hand and Finger-based Interfaces

Several interactive interfaces have been developed to enable natural hand and fingertip interactions. The technologies used for these systems are mainly categorized into two approaches: sensing-based [4, 18] and computer vision-based. The works with computer vision-based techniques include using single color or infrared camera [9, 12], multiple cameras [15, 23], hardware supports, e.g. glass [7, 23], colored gloves or markers [3, 16]. The accuracy of fingertip detection using computer vision techniques depends on lighting conditions and may not be well maintained in the face of sudden changes on the table.

On the contrary, sensing-based systems are usually more robust as they made use of electronic devices (see Diamond-Touch [4] and SmartSkin [18] for examples). One of the drawbacks of sensing-based systems is the limitation of detecting only “touch” behaviour. They are often not capable of recognizing hands or physical objects not being put on the table. Our work uses computer vision approach since we believe that interactions should involve more than just “touching” the surface. The system should be able to detect gestures that are performed over the table as well.

Paper-based User Interfaces

Most of the existing paper-based interfaces fall into three main categories: using papers alongside digitizing tablets

[14, 19], using digital paper technologies [1, 2, 13, 26], and using papers tagged with markers (barcode, fiducial markers, etc.) [3, 9, 11]. The current prototype of our method uses real papers with ARTag [5]. A region of 2x2cm on each paper is needed for printing a two-dimensional visual marker. We can also use other fiducial marker detection techniques, e.g. AR-ToolKit or more sophisticated computer vision algorithms to track papers without visible markers.

This paper presents a novel method that enables multiple users to use both fingertips and physical papers as mediums for interaction on tabletops using only a color camera. Our paper builds on previous research works, but is essentially different in one or more out of three following points.

First, our method supports simultaneous multiple fingertip interactions regardless of their orientations. Several similar systems allow only one hand or fingertip at a time or assume a fixed finger orientation which is not realistic [9, 24, 25]. Our technique also supports bare fingertips, while some others require users to wear markers or gloves [3, 16].

Second, although there are some touch detection algorithms proposed using stereo or two cameras [15, 23], shadow information with infrared camera [24], or with hardware support [7] “clicking” behaviour is generally emulated by keeping the finger still for a certain amount of time (dwelling technique) [12, 25, 27]. Our method overcomes this problem, making use of a small simple diffuse laser source, permitting only one color camera to recognize when and where fingers touch the surface. A related system [20] uses infrared laser to detect touch and hence cannot sense other color objects.

Third, despite a large body of literature, the existing systems mostly enable users to use fingers to manipulate only virtual digital elements, e.g. projected images, maps and texts [12, 16]. This does not provide users with the tactile feeling that they have when working with paper. As for papers, they have been used mainly with pen [1, 2, 13], or just served as a solid quadrangle-shape background to make fingertip detection easier [15, 27]. There is surprisingly little work proposing interfaces that take advantage of fingertip gestures in combination with real physical papers. The most similar research to our approach regarding this point are presented in [3, 9, 25]. However, the technology presented in PaperWindows [3] is complicated and cumbersome to implement (12 cameras along with reflective markers on user’s hand); The Interactive Textbook system [9] used an infrared camera, therefore not being able to recognize color images if necessary; Both Interactive Textbook and WikiTUI [25] also did not support touch gesture, and as shown in [27] would require more time for typing or selection tasks that involve touching the surface.

DOCKLAMP: CONTEXT OF OUR WORK

The method presented in this paper is developed and tested with, but not limited to, the Docklamp, a portable projector-camera system designed at our lab (Fig. 1). It consists of two separated parts. The first part contains a projector and a color camera, while the second, at the bottom base, contains a

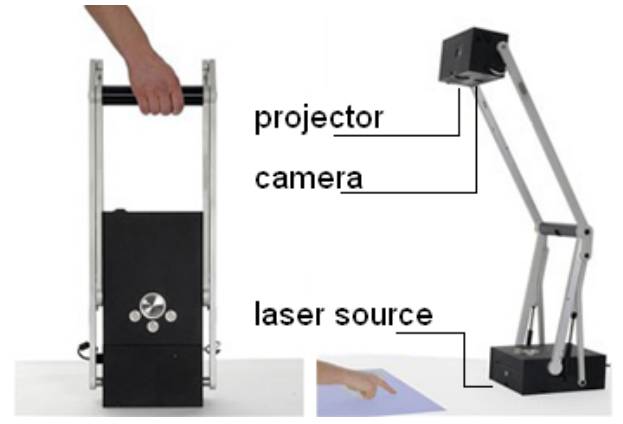


Figure 1: Docklamp consists of a projector, a camera on top, and a laser source at the bottom. It is portable to be carried around.

mini-PC and a mechanical system that permits an easy rotation of the whole lamp. The base also includes a small laser source that can emit diffused beams on top of the table to enable touch detection. Since the projector and the camera are rigidly fixed with respect to each other, the calibration between them needs to be done only once and remains the same even when the system is moved.

When opened, the Docklamp offers a projection size of 45cm by 35cm. Paper tracking with ARTag markers is run at the resolution of 1280x960. Fingertip detection is performed at 640x480. Due to the limitation of our commodity camera, the video is captured at 7 frames per second (fps) although both algorithms can run up to 15fps.

BASIC CAPABILITIES AND IMPLEMENTATION

Our method consists of three key components: fingertip detection, touch detection and touch on paper detection, each of which serves as a building block to the whole.

Fingertip detection

We detect fingertips’ positions by applying a three-step process. The first step is hand extraction which distinguishes hand regions from the rest of video frame. Then we use template matching technique to detect fingertips on the hand regions. A background model is to be updated for the next loop.

Hand extraction

Moving hands are extracted using background subtraction technique. It means that we make a subtraction between the current video frame recorded by camera and the estimated background (an image including only static objects on the table) to gain a difference image.

The process is done in the *rgb* (normalized *RGB*) color space ($r = R/S, g = G/S, b = B/S, S = R + G + B$, with R, G and B being the values in the red, green and blue channel of a *RGB* video frame) as it can eliminate lighting effects on color chromacity. The value of a pixel (x, y) in



Figure 2: A hand mask resulted from background subtraction, auto-thresholding and skin validation. The white part shows detected hand regions.

the difference image is the maximum difference value in all color channels between pixel (x, y) in the background and pixel (x, y) in the current frame.

The Otsu’s algorithm [17] is then performed on the difference image to get a thresholded binary mask. This algorithm automatically decides a threshold that separates the parts of the difference image belonging to background from those belonging to moving objects. Any moving object that is smaller than a predefined threshold, which represents the size of a normal hand, is eliminated to discard noises and small coincidental moving entities.

We use a validation step to ensure only moving blobs that have enough skin-colored pixels inside will be marked as a hand. We based this step on the technique proposed in [10]. A pixel is considered as a skin-colored pixel if it satisfies a heuristic rule¹ defined in RGB color space with $(R, G, B) \in [0, 255]^3$. After this step, we obtain a so-called “hand mask” whose pixels have value of 1 (or white) if they are inside the hand regions and 0 (or black) if outside (Fig. 2).

Fingertip detection

A fingertip can be seen as a connected component of several points that are near one end of a cylinder (the finger). An observation shows that if a circle having a certain radius whose center is one of those points is to be drawn on the image, it would be divided into two parts: one part is totally inside of the fingertip (red segment in Fig. 3a), the other is totally outside (blue segment). Based on this fact, we use a geometric template that can detect multiple fingertips (Fig. 3b). A similar idea is used in [12] but the cylindrical property of the finger is not ensured, so it might mistakenly detect a point at the end of a triangle as a fingertip.

A pixel p is a fingertip point if a set of following conditions is satisfied ($r1, r2$ are manually chosen thresholds):

¹
 $((R > 95) \wedge (G > 40) \wedge (B < 20) \wedge (|R - G| > 15) \wedge (R = \max(R, G, B)) \wedge (R - \min(R, G, B) > 15)) \vee ((R > 220) \wedge (G > 210) \wedge (B > 170) \wedge (|R - G| \leq 15) \wedge (B = \min(R, G, B)))$

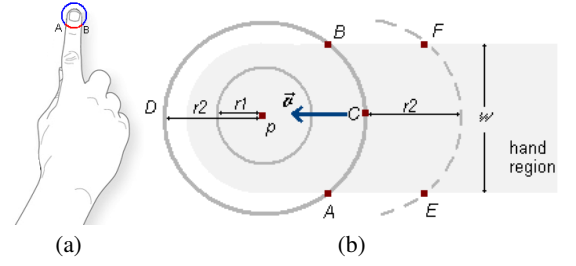


Figure 3: The template used for fingertip detection and its result. a) Observation: a circle drawn on any fingertip is divided into two parts b) Our geometrical template is checked against every pixel p in the image c) Result of fingertip detection. Pens that are hold in hand can also be detected.

- Every pixel within a distance $r1$ from p have a value of 1 in the hand mask since it lies in the finger region.
- When checking every pixel on the circle border whose center is p , radius is $r2$, we see 2 arc segments (\widehat{ADB} and \widehat{ACB}), every pixel on \widehat{ADB} has value of 0, every pixel on \widehat{ACB} has value of 1. (Note that A, B, C are found at runtime. D is only used for explanation)
- Let C be the middlepoint of arc \widehat{ACB} . Count the number of 1-valued pixels on the circle whose center C , radius $r2$. This number needs to be in a specific range.
- The lengths of line segment AB and EF are as long as a fixed threshold w which represents the finger width. This rule ensures the cylindrical property of the finger.

The finger orientation can be easily achieved from the model by drawing a vector \vec{v} from C towards p (Fig. 3b). Figure 3c shows the result of this step.

Background updating

A common and effective background estimation approach is the running average technique [8, 12, 25]. One problem with the running average technique is that it often includes user’s hand into background and in the meantime cannot reflect spontaneous events (e.g. physical objects such as mobile phones and pens suddenly put on the table) that often occur in real-life situations. Even “worse”, the Docklamp,



Figure 4: We use convex hulls of detected hands and fingertips to support background estimation. a) Video frame and detected fingertips b) Estimated background

unlike traditional designs, enables users to rotate the whole lamp, causing sudden changes in the video scene.

To overcome this problem, we suggest a solution using statistical information of the pixels in conjunction with information at object level, namely fingertips and moving hands. Let H be the set of convex hulls of hands with at least one fingertip, the value of a pixel p in background model at time $t + 1$, denoted as $B_{t+1}(p)$ is computed as follows (α is a coefficient, $I_t(p)$ is the value of pixel p in current frame at time t).

$$B_{t+1}(p) = \begin{cases} B_t(p) & \text{if } p \in \{H\} \\ (1 - \alpha) \cdot B_t(p) + \alpha \cdot I_t(p) & \text{otherwise} \end{cases}$$

This equation shows two points about the value of pixel p in background model at time $t + 1$. Firstly, its value will not be changed if it is inside a hand's convex hull that consists of at least one fingertip. In other words, all hands having at least one fingertip would not be updated unexpectedly into the background model. Background regions corresponding to hands and fingers are still preserved as in the previous frame. Secondly, if the pixel is not inside any of the hands detected, its value is normally computed according to the running average technique, i.e. as a weighted sum of its previous value and the value of the pixel at the same position in the current video frame. We use a high value for α ($\alpha = 0.08$ in our experiments) to make the system almost immediately update the changes taking place on the table. Fig. 4 shows an example for our knowledge-based approach.

Touch detection

As previously mentioned, “clicking” behaviour is generally implemented by dwelling, i.e. keeping the finger unmoved for a certain amount of time, or by multiple cameras. Integrating a small laser source in the Docklamp's base enables us to detect touch using only a color camera. Our laser source spreads a very thin sheet of harmless diffused laser just above the table. The finger touching the surface will result in a red-colored dot in the video frame. In our configuration, it is interesting to see that a touch could be reliably identified at those pixels that have ($230 < R < 255$) and ($0 < G < 160$) (Fig. 5). We group those red pixels into a connected cluster. The average coordinate of a cluster represents a touch. In practice, we also check touch validity

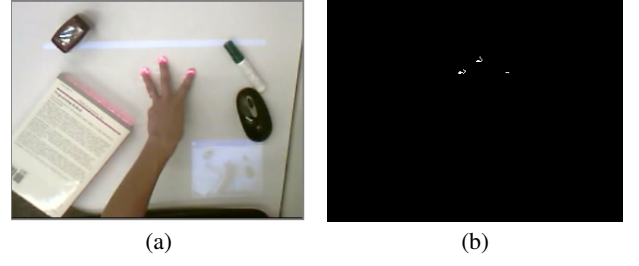


Figure 5: Touch detection: a) The red dots appear on fingers when they touch the table b) Detected touches.

by ensuring that it is close to one of the fingertips detected at the previous step.

Touch on Paper detection

Based on the two components above, we envision a natural and intuitive method for interacting on augmented tabletops with users using their fingers to point or touch the paper. To this end, we use the ARTag library [5] to track real papers in real-time. Each paper within the workspace contains a visual marker at a corner that helps the system to detect its position. A paper can be considered as a control device with some control regions marked on it. Each control region has a particular action associated with it and can be activated by touching it with a finger.

For this purpose, we specify control regions on the paper in real world measurement system (cm or mm) and match this control region with fingertip/touch's positions to know if the control region has been activated. More specifically, a control region can be defined as a region P in real world system with respect to the ARTag marker that is printed on the paper. A mapping of real world measurement system to camera's measurement system (pixels) is obtained by calibrating the camera initially. Using this mapping we can obtain the region P in pixel distance, and still with respect to the marker. Once the position of marker in camera's system is known every frame, we can obtain the coordinates of the region P in camera's system. The system will then use these coordinates along with fingertip and touch positions to decide whether to activate the region P 's action. That is, any fingertip or touch that appears in the region P will send a specific command to the computer. The Paper Keyboard presented below is an example of this type of interaction.

INTERACTION TECHNIQUES

Combining finger and touch gestures with real papers opens a wide variety of interaction techniques in a simple and natural way on augmented tabletops. Figure 6 depicts a basic set of gestures supported by our approach. They can be used separately (i.e. finger *or* paper interaction), or in combination (finger *alongside* paper interaction, or fingertip interaction *on a paper*).

We illustrate the interaction techniques through typical scenarios of their operations. We take the example of a concept map application with which multiple users collaborate

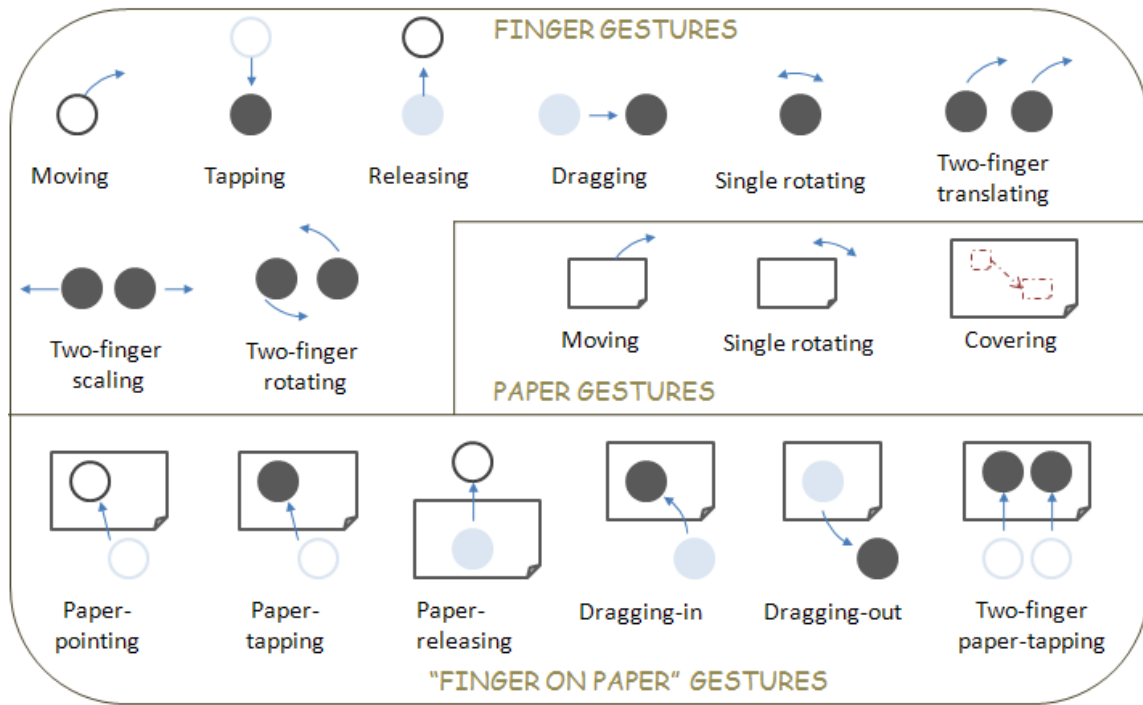


Figure 6: The basic gestures supported by our approach. A hollow circle shows a fingertip hovering the surface. A solid circle shows a fingertip touching the surface. A rectangle represents a paper.

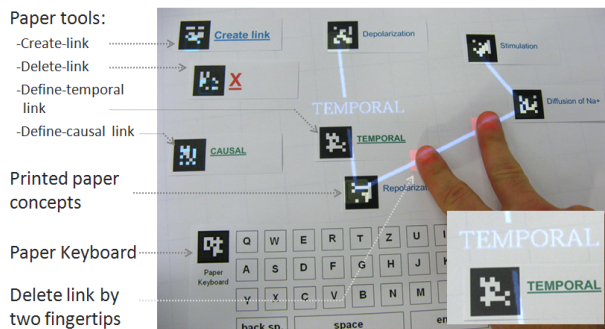


Figure 7: Paper concepts, some paper tools and the Paper Keyboard. The close-up shows a “define-link” tool and the link definition projected on the table.

to build a graph with the concepts representing nodes and the relations between concepts representing links.

Paper is used in the illustration in three object types (Fig.7). *Printed paper concepts* are the concepts that are printed on small pieces of tagged papers. *Paper tools* include several special pieces of tagged papers, each of which represents a specific command to the system, such as creating or deleting a link. *Paper Keyboard* is the keyboard layout that is printed on a paper.

Concept manipulations

Since printed paper concepts are tangible, the users can manipulate them easily with hands. A concept could also be assigned to a blank tagged paper: one types a text definition

using the Paper Keyboard. To begin, one moves and hovers her finger over the table (*moving* gesture) and aims at a specific key on the keyboard (the *paper-pointing* gesture). The key below her finger will be highlighted with visual feedback from the projector as a confirmation. The typing action is done by performing a *paper-tapping* gesture, i.e. fingertip touching the paper and a *paper-releasing* gesture, i.e. lifting the finger up off the paper. As one is typing, a line of text will appear on the table, right next to the Paper Keyboard showing what is being typed. *Single rotating* the Paper Keyboard will rotate the text accordingly. One can do this gesture by simply rotating the paper, having the text oriented better for her partners to see.

After typing the text, the user presses a key to assign this text to the blank paper. One can still interact with this paper as the other printed concepts, except that a digital text is now printed on the paper as its definition. Alternatively, the text can be moved from the table to the blank paper by a *dragging-in* gesture: one simply moves her finger over the text, taps on it, drags it onto the paper (being holded stable) and releases the finger to complete the action. In much the same way, one can un-assign a defined blank paper concept by a *dragging-out* gesture. One drags the digital text that is projected on the paper out of it, leaving the paper blank as it was initially.

Link creation, definition and deletion

There are two options to create a link. First, one can bring two paper concepts close to one another to create a link, projected by the projector, between them (“bring-close” gesture). Second, one can use a “create-link” paper tool: put it

close to a concept and then move it close to another. Once created, the links remain even when people accidentally hide the tag, or take the concepts out of scene. This feature can be used as a way to simplify complex maps by temporarily taking away unnecessary concepts, leaving only concerned elements on the table.

To define a link description, one types a text on the table, then moves two paper concepts and their digital link towards the text. As soon as this link touches the text, it will stick to the text permanently. The text will be shown in the middle of the link as its description and moved when the link is moved. A simpler way to define a link description would be using the “define-link” paper tools. Each of the tools bears a predefined specific text representing the relationship, e.g. causal, temporal, whole/part, etc. One places the “define-link” paper tool in the middle of the link, causing the text printed on the tool to stick to the link.

Users have two possibilities to delete a link: either by using “delete-link” paper tool, or by performing an erasing gesture. To perform this gesture, one stretches two fingers, e.g. the index and the middle finger, moves them into the middle of a link, and lets them be on the link for two seconds to delete it.

Map saving and loading

Some special paper tools with different sizes and dimensions can be used in our system for the function of “snapshot”. Using a *covering* gesture with paper, i.e. bringing and putting a “snapshot” paper tool with a specific size on the table, will save all concepts’ positions and links under it. One can take away these paper tools as physical versions of the concept map. They can be also transferred from one person to another.

The next time, one brings the snapshot paper over, putting it on the table. A digital graph representation will be reconstructed from what the group has done. Users may continue to alter the digital map using their fingers (e.g. *translating*, *scaling*, *rotating*, Fig.6). Since our method is capable of not only detecting fingertip positions, but also finger’s orientations, *single-finger rotating* gesture can therefore be supported to rotate the view.

USER EXPERIENCE AND DISCUSSION

As the scenario above involves many interaction techniques and may cause certain limitations in observing differences in collaboration process and performance, we decided to conduct two user studies with two parts of the system. The first study is an experiment about the efficiency of the Paper Keyboard. The idea behind the Paper Keyboard and not a virtual keyboard projected by the projector is that the hands when typing can cover the projected keyboard layout, hampering users from seeing clearly the key to press. In this study, we saw that a novice user was able to learn fast on the course of typing with little difficulty. During a 20-minute training session, we observed the maximum typing speed recorded with Paper Keyboard was 62.2% as fast as the standard keyboard in the same condition.



(a)



(b)

Figure 8: The Paper Concept Map experiment: Three subjects using a) Computer b) Our interface. People tended to involve more in the task and collaborated differently using our interface.

Due to the lack of space, we just present the second study with concept map that concerns a broader scope: a complex collaborative learning task with a great number of tangible papers on the table.

Paper Concept Map experiment

In this scenario (Fig.8), we observe qualitatively how finger and paper-based interactions can be used together and how the collaboration process changes with these new interaction techniques. We expect that by allowing both interactions, complicated collaborative tasks that involve many tangible objects will be differently structured and achieved.

We conducted an experiment in which 30 undergraduate students, divided into ten groups of three, have to do a collaborative learning task: building a concept map about neurons and their functional mechanisms. Five groups did the task with a program (IHMC CmapTools) on traditional computer as a baseline comparison, and five groups used our interface. This interface was a partial implementation of the whole scenario above. The functionalities provided in this implementation are as follows. Subjects can use only pre-printed paper concepts, and “define-link” paper tools (without typing concepts or link’s definitions on-the-fly). They were also allo-

wed only to use the “create-link” paper tool to make a link. The reason we did not support the “bring-close” option is, in the pilot study, it created lots of accidental links due to the density of physical paper concepts on the table. The two ways of deleting a link with paper and finger were both provided. No typing, saving and loading actions were included.

After a hands-on explanation about the system, each member of a group had to read a different one-page document and explain content of the text to the two partners. The group was then given 23 important paper concepts from the three texts altogether and was asked to build collaboratively a map that shows the relations of those concepts with five “define-link” paper tools (causal, temporal, whole/part, place, property). One might argue that 23 is not a great number, but given the study context with cognitive load demanded by the learning task and small size of projection, we reckon that it is a suitable number.

All subjects noted that it took very little to learn and to use the interface using our technique. The most common positive comments are “appealing and easy to use”, “better for our teamwork”. As opposed to the study in [6], in which the technique can be intrusive to users due to the heights of “tangible bricks”, we saw that our subject can concurrently and freely interact on the table surface with their hands’ movements not obstructed by physical objects.

Our early observations shows an important difference in the linearity in doing the task. The process of building concept maps using our interface is normally divided into two phases: individual work phase, i.e. participants simultaneously creating links for concepts from their own texts, and group work phase in which they combined the works together and elaborated about the relationships (this pattern emerged in three groups out of five). Conversely, all five groups of student using computer collaborated right at the beginning of the task. From the first moments, they built the whole concept map together, explaining their own text on the course of building the map. We believe that it is due to the tangible nature of the system and by the combination of finger and paper-based interactions. It allows more possible actions simultaneously (paper-paper and paper-finger), allowing subjects doing their own tasks without waiting for their partners and hence in some way, structures the collaboration flow into different phases.

Besides that, the interface also contributed to the level of participation in the task. Two of the groups using computer never changed mouse during the task with only one person creating links and moving concepts all the time. All subjects using our system manipulated with the concepts and collaborated with their partners in creating or deleting links many times. They reported that they enjoyed having more gestural interactions (handing a paper piece over to a partner, helping to re-arrange the layout, using fingers to delete a link while a partner ready to take over the concepts after that, etc.).

During ten experiments with controlled lighting condition (indoor, no direct sunlight, illuminance varying from 500-

700 lux), there was no serious error caused by the computer vision algorithms presented in the Implementation section. However, the users occasionally occluded the visual tag markers of paper concepts, culminating in them not being detected. The system also mistakenly recognized or missed fingertips several times. This often happened when the user tried to interact with their fingers upfront, or being in an awkward position that the camera could not see. We saw that the system could recover autonomously when the users moved the hands out of scene for about two seconds.

As for usability issues, subjects made several complaints about the system since there were too many concepts in the task. They stated “it is hard when I have to look for a concept among the others”, “the projection is too small for these papers” (the projection size is 35x45 cm, a paper concept size is 2x7 cm), “I cannot put my fingers in between the link”. It implies that our method, or tangible interfaces with fingertip generally, is probably best utilized when the distribution of paper pieces on the table is not too dense.

It is interesting to find out that the subjects used evenly both way of deletion: erasing gesture by hand and using a paper. Although it took more waiting time for the link to be deleted by finger, some people said that fingertip deletion is more natural for them since they do not have to think and look for the special tagged paper among a lot of other papers. The others meanwhile claimed that they use two fingers to delete link at the beginning of the task, but preferred to use paper to delete in later stage as it is small and hence more suitable in cramped space. They both suggested that either one interaction is not enough if the task involves a great number of tangible papers to look for. It shows that these two types of interaction are complementary to one another and allow greater flexibilities for the users.

CONCLUSION

We have presented an original vision-based method that supports simultaneous multifinger-based and paper-based interactions. The approach is easy and inexpensive to construct. The systems using our approach can allow users to interact in different ways: using fingers, physical papers and both together. The user study suggested that the use of these new interaction styles extends the types of actions that people can perform under augmented tabletop environments and somehow, fosters the collaboration process in groupwork, especially with complex tasks that involve many tangible papers.

ACKNOWLEDGEMENTS

This work is funded in part by the pro*doc program of the Swiss National Science Foundation. We are thankful to Mirweis Sangin for the help in experiment planning, Quentin Bonnard, Cécile Bucher, Khaled Bachour, Aurélien Lucchi, Florence Colomb and Guillaume Zufferey for their supports during the study and the early drafts of this paper.

REFERENCES

1. Signer Beat and C. Norrie Moira. Paperpoint: a paper-based presentation and interactive paper prototyping tool. *Proceedings of the International*

- Conference on Tangible and Embedded Interaction*, pages 57–64, 2007.
2. P. Brandl, M. Haller, J. Oberngruber, and C. Schafleitner. Bridging the gap between real printouts and digital whiteboard. *Proceedings of the conference on Advanced Visual Interfaces AVI'08*, 2008.
3. Holman David, Vertegaal Roel, Altosaar Mark, Troje Nikolaus, and Johns Derek. Paper windows: interaction techniques for digital paper. *Proceedings of CHI'05*, pages 591–599, 2005.
4. P.H. Dietz and D.L. Leigh. Diamondtouch: A multi-user touch technology. *ACM Symposium on User Interface Software and Technology (UIST'01)*, pages 219–226, 2001.
5. Mark Fiala. ARTag, a fiducial marker system using digital techniques. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 590–596, 2005.
6. M. Fjeld, K. Lauche, M. Bichsel, F. Voorhorst, H. Krueger, and M. Rauterberg. Physical and virtual tools: Activity theory applied to the design of groupware. In *B. A. Nardi and D. F. Redmiles (eds.) A Special Issue of CSCW: Activity Theory and the Practice of Design*, volume 11(1-2), pages 153–180, 2002.
7. Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. *ACM symposium on User interface software and technology*, pages 115–118, 2005.
8. Scott R. Klemmer, Mark W. Newman, Ryan Farrell, Mark Bilezikjian, and James A. Landay. The designers' outpost: a tangible interface for collaborative web site. *ACM symposium on User interface software and technology (UIST'01)*, pages 1–10, 2001.
9. H. Koike, Y. Sato, Y. Kobayashi, H. Tobita, and M. Kobayashi. Interactive textbook and interactive venn diagram: natural and intuitive interfaces on augmented desk system. in *CHI '00*, pages 121–128, 2000.
10. J. Kovac, P. Peer, and F. Solina. Human skin color clustering for face detection. *EUROCON 2003. Computer as a Tool. The IEEE Region 8*, 2:144–148, 2003.
11. Nelson Les, Ichimura Satoshi, Ronby Pedersen Elin, and Adams Lia. Palette: a paper interface for giving presentations. *Proceedings of the SIGCHI conference on Human factors in Computing Systems*, 1999.
12. J. Letessier and F. Berard. Visual tracking of bare fingers for interactive surfaces. *ACM Symposium on User interface Software and Technology (UIST '04)*, pages 119–122, 2004.
13. Chunyuan Liao, François Guimbretière, Ken Hinckley, and Jim Hollan. Papiercraft: A gesture-based command system for interactive paper. *ACM Trans. Comput.-Hum. Interact.*, 14(4):1–27, 2008.
14. W. Mackay, G. Pothier, C. Letondal, K. Boegh, and H. Erik Sorensen. The missing link: augmenting biology laboratory notebooks. In *ACM symposium on User interface software and technology (UIST'02)*, pages 41–50, 2002.
15. S. Malik and J. Laszlo. Visual touchpad: A two-handed gestural input device. In *Proceedings of the 6th International Conference on Multimodal Interfaces, ICMI 04*, pages 289–296, 2004.
16. C. McDonald, G. Roth, and S. Marsh. Red-handed: collaborative gesture interaction with a projection table. *International Conference on Automatic Face and Gesture Recognition, FG2004.*, pages 773–778, 2004.
17. N. Otsu. A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66, Jan. 1979.
18. J. Rekimoto. Smartskin: an infrastructure for freehand manipulation on interactive surfaces. *SIGCHI Conference on Human Factors in Computing Systems (CHI'02)*, pages 113–120, 2002.
19. Lisa Stifelman, Barry Arons, and Chris Schmandt. The audio notebook: paper and pen interaction with structured speech. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 182–189, 2001.
20. Carlo Tomasi, Abbas Rafii, and Ilhami Torunoglu. Full-size projection keyboard for handheld devices. *Communications of the ACM*, 46(7):70–75, 2003.
21. B. Ullmer and H. Ishii. The metadesk: models and prototypes for tangible user interfaces. *ACM Symposium on User interface Software and Technology, (UIST97)*, pages 223–232, 1997.
22. Pierre Wellner. Interacting with paper on the digital desk. *Communications of the ACM*, 36(7):87–96, 1993.
23. A. Wilson. Touchlight: An imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th International Conference on Multimodal Interfaces, ICMI 04*, pages 69–76, 2004.
24. Andrew Wilson. Playanywhere: a compact interactive tabletop projection-vision system. *ACM symposium on User interface software and technology*, pages 83–92, 2005.
25. Chih-Sung Wu, Susan J. Robinson, and Ali Mazalek. Turning a page on the digital annotation of physical books. In *Proceedings of TEI '08*, pages 109–116, 2008.
26. Ron Yeh, Andreas Paepcke, and Scott Klemmer. Iterative design and evaluation of an event architecture for pen-and-paper interfaces. 2008.
27. Z. Zhang, Y. Wu, Y. Shan, and S. Shafer. Visual panel: virtual mouse, keyboard and 3d controller with an ordinary piece of paper. In *PUI '01: Workshop on Perceptive user interfaces*, pages 1–8, 2001.