https://doi.org/10.5392/IJoC.2017.13.1.068

Interactive Typography System using Combined Corner and Contour Detection

Sooyeon Lim
School of Techno Public Talent
Dongyang University, Dongducheon, 11307, South Korea

Sangwook Kim
School of Computer Science and Engineering
Kyungpook National University, Daegu, 41566, South Korea

ABSTRACT

Interactive Typography is a process where a user communicates by interacting with text and a moving factor. This research covers interactive typography using real-time response to a user's gesture. In order to form a language-independent system, preprocessing of entered text data presents image data. This preprocessing is followed by recognizing the image data and the setting interaction points. This is done using computer vision technology such as the Harris corner detector and contour detection. User interaction is achieved using skeleton information tracked by a depth camera. By synchronizing the user's skeleton information acquired by Kinect (a depth camera,) and the typography components (interaction points), all user gestures are linked with the typography in real time. An experiment was conducted, in both English and Korean, where users showed an 81% satisfaction level using an interactive typography system where text components showed discrete movements in accordance with the users' gestures. Through this experiment, it was possible to ascertain that sensibility varied depending on the size and the speed of the text and interactive alteration. The results show that interactive typography can potentially be an accurate communication tool, and not merely a uniform text transmission system.

***Key words**: Interactive Typography, Interaction Point, Skeleton Matching, Corner Detection, Contour Detection.*

1. INTRODUCTION

With the recent invigoration of culture and arts scene, the influence of typography, an act that transforms and relocates a letter and sublimates it into a form of design, is becoming larger in our daily lives. The development of digital skills has extended the area of typography; it is deviating from paper, a limited space, and making its way into easily accessible and dynamic space such as monitor and screen. It is also actively utilized in not only brand or product marketing, but also in various media related exhibition.

Kinetic typography, a typography with movement, is used especially often in fields that require effective presentation and delivery of the text, such as media art, music video, advertisement, et cetera. As a result, it is taking center stage as an important design factor that even affects economic value such as enterprise sales. Despite the increase in demand for kinetic typography due to its endless potential as a communication language of great artistic value, drawbacks do exist: it is language-specific and its expression method is limited.

In order for an amateur to produce kinetic typography contents himself, it is possible to attain an elaborate output with various effects by making use of existing universal programs such as Adobe AfterEffect and Flash. However, the complex interface of existing tools is not an adequate environment for amateurs to easily make practical use of; it rather takes more time to learn the interface. Thus, it is required that a smooth interface be developed so that users can drive kinetic typography without particular expertise. And most of the existing kinetic typography utilizes the RSVP(Rapid Serial Visual Presentation) method, in which words or letters in a sentence move orderly, one by one, or words appear at a high speed, one at a time. In order for general users to produce kinetic typography in which words or letters move as a unit, they must use professional animation production tool or SVG(Scalable Vector Graphics) for programming. However, this method is inadequate for an efficient production of kinetic

* *Corresponding author, Email: sylim@dyu.ac.kr*
Manuscript received Feb. 06, 2016; revised Mar. 13, 2017; accepted Mar. 17, 2017

typography in that it requires professional skills and burdensome production process.

Kinetic typography is a technique to image and design texts for communication purpose. In order for the establishment of a kinetic typographic system, it is required that font rendering engine and message transmission systems, which allow various communicative expression and sensibility transmission between users, be developed. Also, because smart interaction between human and digital device has been made possible nowadays due to the development of various sensor and computing power skills, the demand for interface skills that back natural human gestures is galloping.

As a result, in order to reflect the active movement of users, the necessity of interactive kinetic typography engine development, in which components of a letter move in accordance with the user's gesture, has been on the rise, thus leading to the beginning of this research. From this point on in the paper, we use interactive kinetic typography as interactive typography. We developed interactive typography system with real-time response to user gestures. The developed system has features that are not limited to a specific language by performing image-based processing rather than text-based processing, thereby saving time and cost of developing typography for each language. While the goal of existing typography was to design a linear typographic outcome with one-way communication, our interactive typography aims to design various outcomes with nonlinear communication. This means our system can make the interaction with the user more emotional.

Our system interacts with the users in real time by recognizing gesture information with TOF(Time Of Flight) camera such as Kinect. A TOF camera measures distance by calculating the time it takes for an infrared light or a photic signal to reach an object and arrive back. Depth information acquired through this method goes under normalization and thus produces depth space and is transformed into a depth image with a depth level ranging from 0 to 255. By utilizing such depth image, computer vision difficulty for separating the object from the background can be solved with ease, and human or animal skeleton information and its movement information can be obtained with relative ease.

For efficient development of this research, the existing kinetic typography system and studies on motion tracking technologies using Kinect will be simply examined in the section 2, and the configuration and implementation of interactive typography proposed in this research will be observed in the section 3. Section 4 will describe practical application in our system and evaluation of the results, and the conclusion will be specified in the final section 5.

2. RELATED WORK

2.1 Works for Creating Kinetic Typography

Various tools have been suggested to create kinetic typography. [1] provided a kinetic typography engine that was made in java language in order to applicate different functions for the change in properties, such as color and brightness of the text, and motion functions such as movement, size, and rotation. However, it is difficult to use in that it does not provide interface. [2] suggested a method in which hierarchy system that divides letters into meaningful units is used to produce and handle interactive letters with real-time movement. [3], a typography protection tool where GUI method is applied, provides interface for editor form and backs kinetic typography comprised with a maximum for fourteen words. [4] developed kinetic typography editor tool that can be synchronized with audio signals such as music and theatrical lines. [2]-[4] are kinetic typography production tools that are combined based on GUI; the user enters text, selects animation out of various templates, and edits parameters that controls animation. These researches also provide interface in the form of editor. While such GUI-based combined typography production tools are user-friendly, they are limited in flexibility in that they are domain-specific and thus can only apply ready-defined effects.

Nowadays, due to change in media, there are active studies on interactive typography which directly reflects user's participation to typography [5], [6]. [7] developed interactive typography that helps children's language acquisition by introducing game elements as a typography interaction method. In 'Letterspace' of [8], users make use of two 3D magnetic field sensors in order to utilize skills to restrict letter movement.

This research defines three common problems of kinetic typographic production tools introduced so far. The first problem is that it is restricted to certain users. The system fundamentally defines a small number of motion algorithms and only allows little edits in the limited range. The second problem is that the minimum unit of existing kinetic typography motion is based on letter units, thus unable to properly reflect the immediacy of emotional expression. Because kinetic typography system is focused on graphic aspects, it only designates motion effects in letter or sentence units, and goes with the method that changes the place of typography (motion path) linearly and visually communicates with the viewers. As a result, it is difficult to reflect active interaction of viewers. Since the system is only provided in certain languages, it requires a lot of time and cost to produce a multi-language kinetic typography. This is because the structural forms of languages from all around the world are highly varied.

2.2 Works for Motion Recognizing using Kinect

Nowadays, with the mounting importance of information skills such as wearable computer, perceptual user interface, affective computing, many computer vision researchers are taking note of research on human posture recognition. In order to recognize and make judgement on human posture with a computer, it is essential to be aware of how each body part changes with the lapse of time.

Kinetic sensor, a motion controller sensor of a game machine developed by Microsoft, is a sensor which can interact without help of a certain device, by only making use of the user's body. It is used in various fields such as game, education, advertisement, et cetera. A Kinect is embedded with one RGB Camera, two Depth sensors, and IR Camera, thus allowing motion recognition by using color image and image information. A Kinect sensor is a low-price depth camera. The data usage provided by Kinect takes off the burden of human/body detection or pose presumption needed for gesture recognition. As a result, research on HCI utilizing Kinect is actively going on [9]-[11].

In [12], mouse event was generated in accordance with distance and direction of the hand reached out from head, shoulders, the center of the body, et cetera. This showed that by using the event, PowerPoint presentation could be controlled. [13] suggested an interface that enables the movement of a third-dimensional image, expansion, contraction, or rotation of the direction of x-axis and y-axis. [14] extracted depth comparison characteristic by using depth image, and presumed human pose by using Random Forest classifier. Researches that developed robot interaction system, embedded with Kinect sensor, also exist. In [15], by recognizing third-dimensional pointing gesture through utilizing depth data acquired from Kinect sensor installed in a robot, a system that allows human-robot interaction was developed. [16] conducted a study on the method to reflect recognized motion data to character animation. However, while there are advantages, its low cost and the ability to track user movement without particular device, it also has a drawback: there is a blind spot, which is a disadvantage of camera-based motion tracking

3. INTERACTIVE TYPOGRAPHY SYSTEM

This research suggests language-independent interactive typography that shows real-time response to user gestures. In order to form a language-independent system, there must be a preprocessing in which the entered letter data is perceived as an image data rather than a text one. Then it is recognized with a computer vision technology and the interaction point is set. User interaction is managed through the utilization of the skeleton information that depth camera tracked. The outline of the suggested system is shown in Fig. 1.

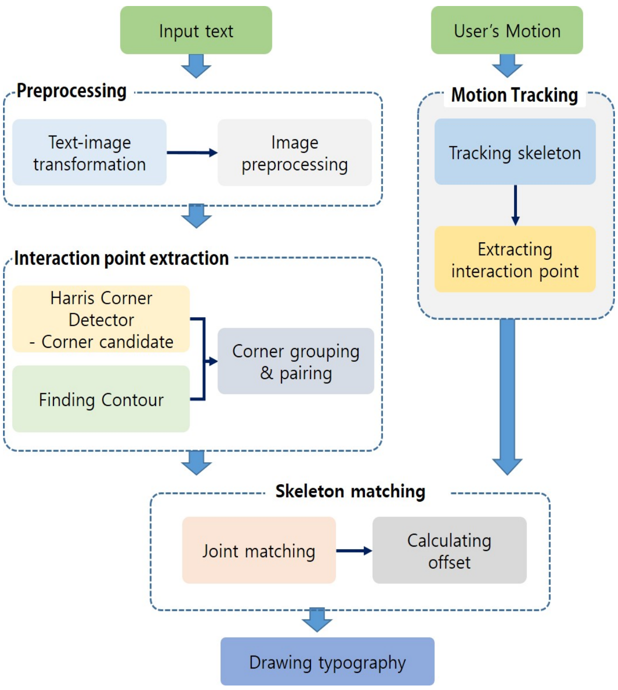

Fig. 1. System overview

3.1 Preprocessing

In order to maximize the text motion effect in real time, text data is converted to an image. First, text data is printed on a certain-sized canvas. Then, the printed image is captured to form a grey-scale image. Such letter recognition process that uses black and white grey-scale image has two advantages. First, it allows the system to be language-independent by perceiving all computer-enterable letter data as image. Another advantage is that it enables more dynamic interaction through dividing the image components. Just as Fig. 2, transforming the partial components of the letter (see (b) of Fig. 2) rather than transforming the letter itself (see (a) of Fig. 2) gives the user a bigger visual effect and allows more lively interaction.

(a) (b)

Fig. 2. Transformed images:
(a) an example of transforming an assembled letter,
(b) an example of transforming a disassembled letter.

Also, in this research, in order for an accurate corner detection operation following the separation of the contact surfaces of letter strokes, an erode operation, one of the morphology filters, was carried out upon the acquired black and white image. Fig. 3 shows the letter before and after the erode operation.

(a) (b)

Fig. 3. Before (a) and after (b) the erode operation was carried out.

3.2 Interaction point extraction

In order to interact while maintaining the form of the letter, it is necessary to be aware of the start and end point of the line that forms the letter. This is a distinctive characteristic that can be found in Asian languages.

3.2.1 Harris Corner Detector: Corner point data is utilized in various fields of computer vision such as object recognition, movement recognition, stereo matching, et cetera. Corner refers to a curve that has a drastic change in the angle or the cross point of two or more contours in an image. Since corner point has the form data of an object, finding the corner point of an object is very important in detection and matching of the object. There are various ways to detect a corner from an image [17]-

[22]. This paper uses the Harris Corner Detector [22]. The fundamental principle of the Harris Corner Detector is as follows: form a certain-sized window in the concern sector, move it into all direction, and recognize the dot with a largest intensity change as the corner (see Fig. 4).

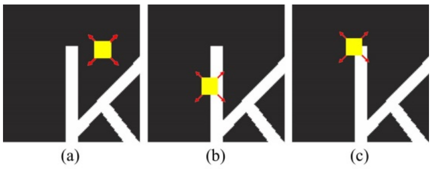

Fig. 4. The corner detection method of the Harris corner detector : (a) “flat” region, (b) “edge” region and (c) “corner” region

As shown in Fig. 4, there is almost no change of intensity in the flat region, when moving to the upper, lower, left, and right sides. In the edge region, the pixel value changes only when the window moves left and right. However, the intensity change largely at all points in the corner region.

If the change of (x, y) in the image is referred to as $(\Delta x, \Delta y)$ the auto-correlation function that calculates the change rate of intensity is as follows [23].

$$C(x,y) = \sum_W [I(x_i,y_i) - I(x_i+\Delta x, y_i+\Delta y)]^2 \qquad \textbf{Eq. (1)}$$

where $I(\cdot,\cdot)$ denotes the image function and (x_i, y_i) are the points in the window W(Gaussian) centered on (x, y).

The shift image is approximated by a Taylor expansion truncated to the first order items,

$$I(x_i+\Delta x, y_i+\Delta y) \approx I(x_i,y_i) + [I_x(x_i,y_i) I_y(x_i,y_i)] \begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} \qquad \text{Eq. (2)}$$

where $I_x(\cdot,\cdot)$ and $I_y(\cdot,\cdot)$ denote the partial derivatives in x and y, respectively.

By putting (2) into (1),

$$c(x,y) = \sum_W [I(x_i,y_i) - I(x_i+\Delta x, y_i+\Delta y)]^2 = [\Delta x \quad \Delta y] \begin{bmatrix} \sum_W [I_x(x_i,y_i)]^2 & \sum_W I_x(x_i,y_i) I_y(x_i,y_i) \\ \sum_W I_x(x_i,y_i) I_y(x_i,y_i) & \sum_W [I_y(x_i,y_i)]^2 \end{bmatrix} \begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = [\Delta x \quad \Delta y] C(x,y) \begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} \qquad \text{Eq. (3)}$$

By using $C(x, y)$ of (3), the cornerness function R can be calculated by the following method.

$$R = det(C(x,y)) - k[trace(C(x,y))]^2 \qquad \text{Eq. (4)}$$

Where *det()* denotes the determinant and *trace()* denotes the diagonal sum of the matrix. R is positive in the corner region, negative in the edge regions, and small in flat region regardless of the sign.

It was possible to detect the corner point by setting the input value of the Harris Corner Detector as the following rate: W as 9, k as 0.04. Fig. 5 shows the result.

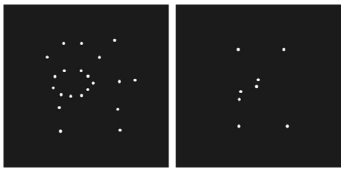

Fig. 5. The corner points of the letter(한, K) by the Harris corner detector

3.2.2 Contour Detection: In order to draw a straight line or a curved line, the elements of a letter, it is necessary to know the start and end point of the line. Because this data cannot be obtained with only corner point detection, contour detection must also be carried out.

It was possible to detect the curve by using cvFindContours function of OpenCV library. Fig. 6 shows the curve detection result of the Korean letter ‘한’ and the English letter ‘K’.

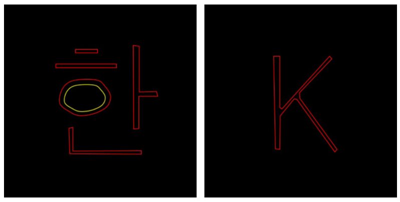

Fig. 6. The detected contour of the letter images

As shown in Fig. 6, due to the characteristics of the letter image, the detected curve is close and there is no self-intersection.

3.2.3 Corner point grouping & pairing: By combining the ready-detected corner point and the curve data, it is possible to group the corner points according to contour.

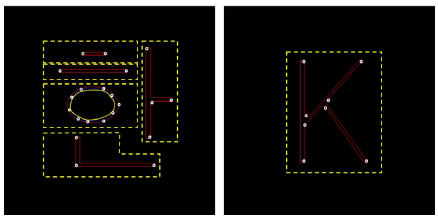

Fig. 7. The grouping of the corner points according to region of interest.

By making use of the corner points and curve data that are grouped according to contour, it is possible to tell distinguish circle and square, which are closed form figures(see Fig. 8). When detecting the curve of a circle or a square, external boundaries as well as inside hole boundaries are simultaneously detected. When an inside hole boundary is detected, it is perceived as a closed figure and thus draws the bounding box of the figure. If there are four corner points in the edge of the bounding box, it is perceived as a square. If not, a circle.

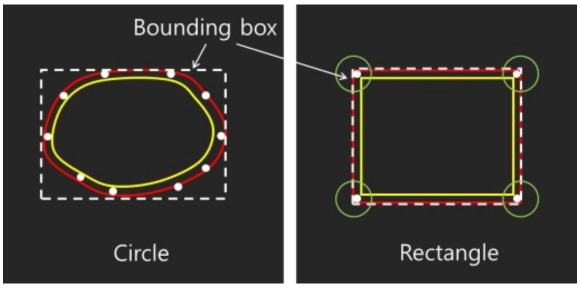

Fig. 8. The distinction of closed figures using bounding box

Corner points aligned according to contours set each other's ROI(region of interest) and seek the probability of each ROI's average pixel value being 0. If the calculated probability value is 0.8 or more, it means that a drawing route exists. Thus, two corner points must be set as final interaction point. If in this case, two or more overlapping routes with the same probability value exist, the further corner point must be chosen as the interaction point. This prevents the overlap of drawing. In Fig. 9, since the probability value of ① and ② are both 1 and they overlap, the one that has more distance between the corner points, the corner point of ②, is set as the interaction point.

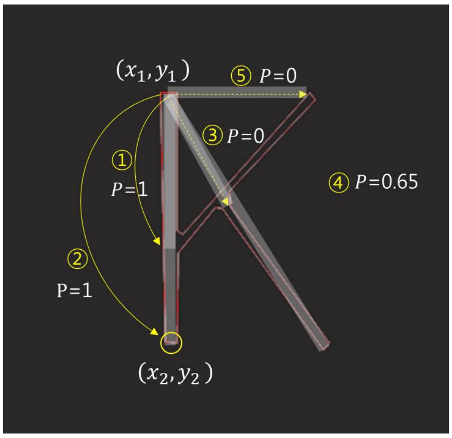

Fig. 9. Calculating the probability of the ROI(gray box) average pixel value being 0(white)

The interaction points are paired into a start point and an end point and are saved into an arrangement of ArrayList form for the final drawing. Fig. 10 is a flowchart which shows the interaction point setting process that has been explained so far.

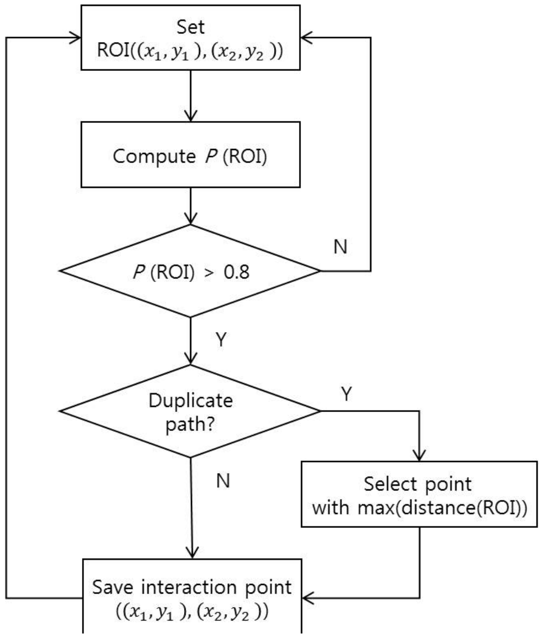

Fig. 10. The flowchart of setting the interaction point

3.3 Skeleton synchronization

3.3.1 Joint matching: After setting the interaction points to draw the letter, the interaction point and the user gesture data must be synchronized in order to enable interaction in accordance with user gesture. User gesture data can be attained through user's skeleton collection from Kinect.

Kinect perceives depth information by exporting ultrared light from the sensor and recognizes the reflected ultrared light. By using the perceived ultrared light, skeleton is organized through 79999the internal library of Kinect SDK. Also, by mapping and tracking each coordinate of the joints, the user's body is understood and the change of location according to movement is shown. Kinect V2, released by MS in 2014, showed increase in the resolution of an RGB data(1920 × 1080) and the depth date(512 × 424). It also provides 25 joint data per person, maximum 6 people. By using NUI API of Kinect, human posture is recognizable through acquiring skeleton data.

Fig. 11 shows the skeleton information that can be attained with Kinect V2. The locations of the joints are marked as small circles. In this system, the 25 joint data in this figure were prioritized when they were synchronized with interaction point. They were divided into two groups, based on the order of priority. While skeleton matching, skeleton point group that is marked green is prioritized over orange skeleton point group. This action was taken in order to show the dynamic visual effects. It also considered the fact that arm and leg are the human body parts with the largest motion mutation.

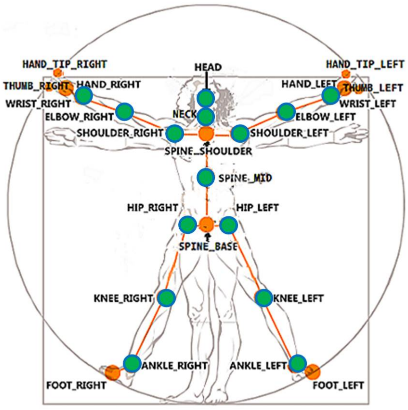

Fig. 11. The acquirable joints from Kinect V2
(matching priority: ● =1, ● =2)

The detected joint point and the interaction point are matched by using the Euclidean distance. The dots with the closest distance are matched. However, the letter image must be divided into four, and the matching is made in the quadrant, in accordance with the priority of joint point.

Euclidean distance is an algorithm that calculates the distance between two dots in a n-dimension. If there are $p=(p_x, p_y)$ and $q=(q_x, q_y)$ in two-dimension, the Euclidean distance that uses the distance between the two dots as the measurement for interrelationship is as follows [24].

$$d(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \qquad \text{Eq. (5)}$$

3.3.2 Calculating offset: If the joint point data is directly matched with the interaction point, large residual may occur and thus lead to a disorder of the letter forms. As a result, a correction work is required. The offset value, necessary for the drawing of the letter, is used. Location difference between the joint point (p_x, p_y) and the interaction point (q_x, q_y) is calculated and used as the value of offset $(x, y) = (p_x - q_x, p_y - q_y)$. Fig. 12 shows the interaction point, the joint point, and the determination process of the offset value for the letter ‘한’.

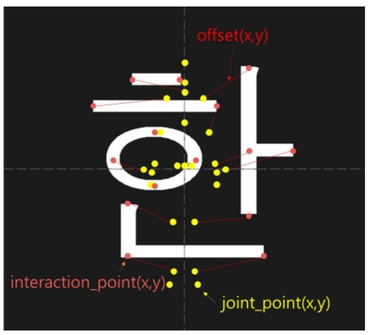

Fig. 12. An example of skeleton matching of the letter ‘한’

3.4 Drawing typography

All data needed for letter drawing is saved in the arrangement of the ArrayList form. The following is the data structure for the drawing.

```
public struct draw_element {
    int mode;
    int joint_x1, joint_y1;
    int joint_x2, joint_y2;
    int joint_x3, joint_y3;
    int interaction_x1, interaction_y1;
    int interaction_x2, interaction_y2;
    int interaction_x3, interaction_y3;
    double offset_x1, joint_y1;
    double offset_x2, joint_y2;
    double offset_x3, joint_y3;
}
```

In the structure, mode is the variable which saves the data about what it is: line, curve, or circle. Each point is composed of three coordinates. In the case of a line, only two coordinates, the start point and the end point, are needed. On the other hand, concerning a curve or a circle, three coordinates are required. The drawing for the circle did not use four coordinates and instead used three coordinates and combined two curves This is because after going through a test in which a circle was drawn by using four coordinates, the output result turned out that the size of the circle was too small and that it showed more passive response toward user interaction.

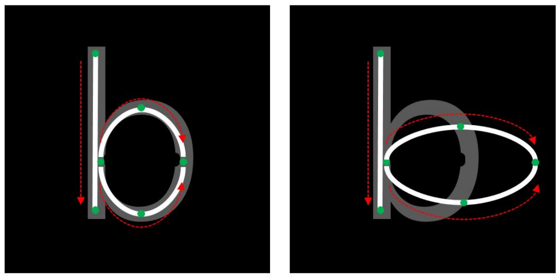

Fig. 13. The result of drawing a circle by using semicircle

4. IMPLEMENTATION & EVALUATION

4.1 Implementation

The system suggested in this research was materialized by using Kinect for Windows SDK 2.07 and OpenCV 2.4.12, Visual Studio 2015 C++ language.

Fig. 14 is a screen in which checking and editing the automatically calculated skeleton data is possible. The user interface was produced so that users can check the skeleton data matched by the algorithm suggested in this research and the offset value (Fig. 14-①), and edit them if needed (Fig. 14-②).

Fig. 15 shows interactive typography materialized through the extraction of skeleton data from the original image using Kinect and the connection of the letter’s interaction point.

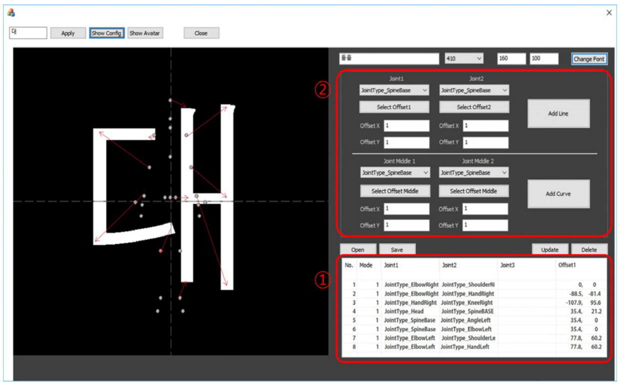

Fig. 14. User interface of interactive typography system

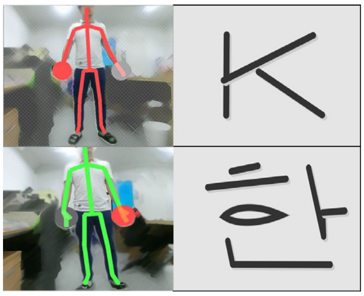

Fig. 15. Skeleton image and interactive typography image for original image

4.2 Evaluation

The standard to objectively draw comparison of extraction algorithm for straight or curved line, one of the implementation technology for interactive typography, is settled in neither domestic nor foreign land. Comparative studies concerning interactive typography is especially difficult because it requires not only line extraction but also drawing along the extracted line. In order to evaluate the suggested system, an experiment to figure out the accuracy of interaction point extraction was carried out, targeting the text data of various languages. Table. 1 shows the reappearance and accuracy rate of the interaction points detected from text images, calculated by using the following formula. The reappearance rate is calculated with the probability of the actual number of detected interaction points divided by the actual number of all interaction points. The accuracy rate is calculated with the probability of the actual number of detected interaction points divided by the actual number of all detected interaction points.

$$Recall = \frac{the\ actual\ number\ of\ detected\ interaction\ point}{the\ actual\ number\ of\ all\ interaction\ point}$$

$$Precision = \frac{the\ actual\ number\ of\ detected\ interaction\ point}{the\ actual\ number\ of\ all\ detected\ interaction\ point}$$

Table 1. Recall and precision of interaction point detection.

Language (number of letters)	Recall	Precision
Korean(135)	87.1%	90.2%
English(52)	82.0%	83.6%
Average	84.6%	86.9%

We also conducted an experiment in which various letters were entered and five participants were asked about the level of satisfaction about the interaction of the printed typography. The result is shown in Table. 2. Due to the characteristic of the experiment using Kinect, while typography was managed very effectively within 3~4 meters, recognition rate was not particularly good at a distance of 7 meters or more.

Table 2. The result of experiment on user satisfaction.

Language (number of letters)	Satisfactory	Dissatisfactory	Satisfaction rate(%)
Korean(135)	112	23	83.0%
English(52)	41	11	78.8%
Average satisfaction rate		80.9%	

5. CONCLUSION

Interactive typography exists as not only a tool to convey meanings, but also an environment that can provide viewers with interaction based on various media. This research has suggested an interactive typography system with real-time response to user gesture. In order to attain this, an interaction tool with user-friendly interface, which provides optimal experiences for users, was developed in this research. The suggested system has two advantages. First, resource consumption can be cut down to a large extent by combining and automatizing the building process according to language characteristic. Second, it is possible to induce immediate visual effect, vie for user's interest, and increase a sense of immersion by coming up with a dynamic scenario that shows dynamic response toward user's motion, instead of a static one. As a result of the experiment, the satisfaction rate about the interaction turned out to be 80.9%.

Interactive typography which shows response in accordance with user movement is a skill that can even convey emotion. Thus, this research has competitive value in that it provides basic reference needed in order for interactive typography to overcome language barriers and make its way into the global market. Also, since it is highly applicable to current smart media or mobile environment, it is likely that interactive typography will help invigorate the related industrial fields such as entertainment, design, media, et cetera.

Henceforth, we plan to conduct experiments concerning typography management skills on more various languages. We also plan on the materialization of more elaborate typography management skill that enables the control of sentence units, by expanding the movement model.

REFERENCES

[1] J. C. Lee, J. Forlizzi, and S. E. Hudson, "The kinetic typography engine: an extensible system for animating expressive text," In Proceedings of the 15th annual ACM symposium on User interface software and technology, 2002, pp. 81-90.

[2] J. Lewis and A. Weyers, "ActiveText: a method for creating dynamic and interactive texts," In Proceedings of

the 12th annual ACM symposium on User interface software and technology, 1999, pp. 131-140.
[3] J. Forlizzi, J. Lee, and S. Hudson, "The kinedit system: affective messages using dynamic texts," In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2003, pp. 377-384.
[4] J. Kato, T. Nakano, and M, Goto, "TextAlive: Integrated Design Environment for Kinetic Typography," In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM, 2015, pp. 3403-3412.
[5] G. Bachfischer and T. Robertson, "From Movable Type to Moving Type-Evolution in technological mediated Typography," In Proceedings of the AUC Academic and Developers Conference, 2005.
[6] [6] G. Bachfischer, T. Robertson, and A. Zmijewska, "A moving type framework," In Proceedings of the 10th WSEAS international conference on Communications. World Scientific and Engineering Academy and Society (WSEAS), 2006, p. 607–612.
[7] N. M. Lau and V. H. Chu, "Enhancing Children's Language Learning and Cognition Experience through Interactive Kinetic Typography," International Education Studies, vol. 8, no. 9, 2015, pp. 36-45.
[8] P. S. Cho, "Computational models for expressive dimensional typography," Doctoral dissertation, Massachusetts Institute of Technology, USA, 1999.
[9] S. Mitra and T. Acharya, "Gesture recognition: A survey," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 3, 2007, pp. 311-324.
[10] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," Pattern Recognition, vol. 40, no. 3, 2007, pp. 1106-1122.
[11] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," Computer vision and image understanding, vol. 104, no. 2, 2006, pp. 90-126.
[12] T. Osunkoya and J. C. Chern, "Gesture-based Human Computer Interaction using Kinect for Windows Mouse Control and Powerpoint Presentation," In Proceedings of the Midwest Instruction and Computing Symposium, Wisconsin, USA, 2013.
[13] K. T. M. Tran and S. H. Oh, "Hand Gesture Recognition for 3D-Heritage-Tourism using Microsoft Kinect Sensor," Advanced Science and Technology Letters, vol. 30, 2013, pp. 145-148.
[14] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," Communications of the ACM, vol. 56, no. 1, 2013, pp. 116-124.
[15] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, M. Buss, "Real-time 3D hand gesture interaction with a robot for understanding directions from humans," In Proceedings of the International Symposium on Robot and Human Interactive Communication, 2011, pp. 357-362.
[16] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2008, pp. 125-134.
[17] S. M. Smith and M. Brady, "SUSAN - A New Approach to Low Level Image Processing," International Journal of Computer Vision, vol. 23, no. 1, 1997, pp. 45-78.
[18] M. Trajkovic and M. Hedley, "Fast Corner Detection," Image and Vision Computing, vol. 16, no. 2, 1998, pp. 75-87.
[19] S. C. Bae, I. S. Kweon, and C. D. Yoo, "COP: a new corner detector," Pattern Recognition Letters, vol. 23, 2002, pp. 1349-1360.
[20] L. J. Latecki and R. Lakämper, "Convexity rule for shape decomposition based on discrete contour evolution," Computer Vision and Image Understanding, vol. 73, no. 3, 1999, pp. 441-454.
[21] U. Orguner and F. Gustafsson, "Statistical characteristics of harris corner detector," In Proceedings of the IEEE/SP 14th Workshop on Statistical Signal Processing, vol. 2629, 2007, pp. 571-575.
[22] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," In Proceedings of the Alvey Vision Conference, vol. 15, 1988, pp. 147-151.
[23] K. G. Derpanis, The Harris Corner Detector, 2004. Accessed Aug. 13. http://www.cse.yorku.ca/~kosta/CompVis_Notes/ harris_detector.pdf, 2016.
[24] R. Fabbri, L. Da F. Costa, J. C. Torelli, and O. M. Bruno, "2D Euclidean distance transform algorithms: A comparative survey," ACM Computing Surveys (CSUR), vol. 40, issue. 1, no. 2, 2008.

Sooyeon Lim
She received the B.S. in electronics engineering from Kyungpook National University, Korea in 1988, respectively and also received M.S., Ph.D. in computer engineering from Kyungpook National University, Korea in 1991, 2006 respectively. Currently she is assistant professor in Dongyang University and Ph. D. Candidate of DigitalMediaArt department in Kyungpook National University. Her research interests are Interactive Art, Motion Graphics and 3D Holographic Projection System.

Sangwook Kim
He received B.S. degree from Kyungpook National University, Korea, in 1979. He received M.S. degree from Seoul National University, Korea, in 1981. He received Ph.D. in Computer Science and Engineering from Seoul National University, Korea, in 1989. He is currently Professor of School of Computer Science and Engineering at Kyungpook National University. He is interested in mobile media systems, media contents, human computer interaction, social network analysis and Internet-of-Things.