An Interactive Holographic Light-Field Display with a Color-Aided 3D-touch User Interface

Iván Alexis Sánchez Salazar Chavarría[1], Tomoya Nakamura[1,2], Masahiro Yamaguchi[1]

[1]School of Engineering, Tokyo Institute of Technology, 4259-G2 Nagatsuta, Midori-ku, Yokohama, Kanagawa, 226-8502, Japan
[2]PRESTO, Japan Science and Technology Agency, Saitama, 332-0012, Japan
Keywords: Light-field, 3D-interaction, displays, holography, touchable interface

ABSTRACT

The author's group previously demonstrated a holographic light-field display with a 3D touch interface, based on the detection of scattered light by the user. That interface is now improved by realizing real-time interactivity and the implementation of 3D motion detection using the color information captured by an RGB sensor.

1 INTRODUCTION

Displays with touching interfaces are very widespread nowadays. As 3-dimensional (3D) display technology develops, the search for a proper 3D user interface (3DUI) becomes more important.

There are many approaches that realize 3DUIs. Recently, gesture interface methods based on Time-of-Flight (ToF) cameras, stereo cameras or structured light methods are very popular [7,8]. Incorporating such gesture interface with the 3D display can realize a 3D functional interface, which will be advantageous for intuitive and easy-to-use human-machine interaction. In such systems, the interaction measurement needs to be fully integrated with the 3D image. If those systems are isolated and there is a mismatch between the displayed content and the position of the user's hand or finger, it can cause an unpleasant interaction. Our group has proposed and developed in previous publications [1, 2] a 3D user interface using a projection-type light-field display based on a screen made by holographic optical elements (HOE). The interface works by detecting the light scattered by a user when touching the reconstructed light field. This can potentially solve the mentioned misalignment problem. The approach uses only an RGB camera, which makes it easy to implement.

The previous research presented in [1] and [2] did not achieve a real-time operational interface. The approaches presented were similar to static buttons. In order to develop interfaces like the ones used by modern touch panels that track the movements and gestures of a user, real-time operation is needed. This work presents the real-time implementation of [1] and [2]. We present a motion detection method based on the sequential detection of scattered light in the camera plane (plane *xz* in fig. 1). To detect movement direction in the axis perpendicular to the camera plane (*y* axis in fig. 1), we propose a color detection method based on the approach proposed in [2].

The successive detection of different colors of the light field can be encoded into a movement direction along this axis. This yields a detection method capable of determining movement direction in 3 axes, while having a precise location of the user interaction in the camera plane.

2 PRINCIPLE

2.1 Concept of a 3D touchable interface using the scattered light field

The approach proposed in [1, 2] consists in using the real image generated by a light-field display. This light focused in front of the user represents the displayed content (a button, a menu, etc.) which can be visually enhanced with 3D cues for increasing its usability. The user touches the light that represents the object, generating a blob of scattered light. This light is detected by an RGB camera, which processes it and gives the system the user's instruction. This method realizes an "on-air" interface that is very intuitive and only requires an RGB camera for its implementation (fig. 1).

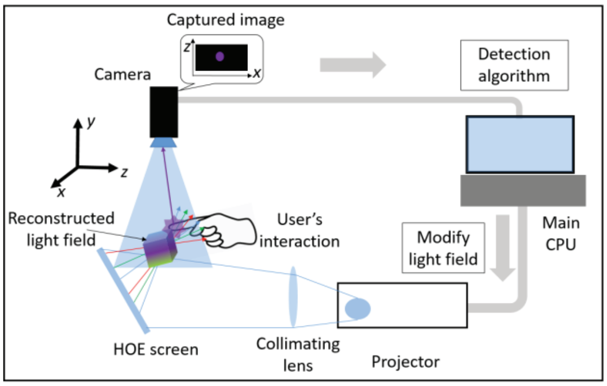

Figure 1. Outline of the experimental set-up and the concept of our scattered light-based interface

Apart from its conceptual simplicity, the proposed interface has the advantage of not requiring a special alignment with a 3D-position sensing hardware, which is the case of conventional gesture-sensors-based approaches [4]. Our approach avoids the misalignment and possible awkwardness that might arise from using one method for measuring the interaction and another one for displaying the 3D content. By using the light created by the object as the mean for achieving the touch-detection, the interaction will not take place as long as the user does not touch the light field, assuring no mismatch can occur.

2.2 Use of color information to detect 3D motion

The concept of color information of the light-field used to increase the functionality of this interface has been demonstrated by our group. In reference [2], color identification is realized by measuring the light scattered when the user touches colored light fields representing "buttons" on air. The vector of normalized pixel intensity values $\left[\frac{R}{R+G+B}, \frac{G}{R+G+B}\right]$ of the picture of an interaction is compared to previously saved values using both Euclidean and Mahalanobis distances.

In the system shown in fig.2, the position of the finger is determined by the position of the detected color signal. However, since the camera is placed at the top of the system, the position along the axis perpendicular to the camera cannot be obtained. In this paper the color information is also utilized to create a cue to detect the direction a user is moving. Embedding different color information along y-axis, the color of the scattered light can be used to detect the direction of the movement of the user (see fig. 2).

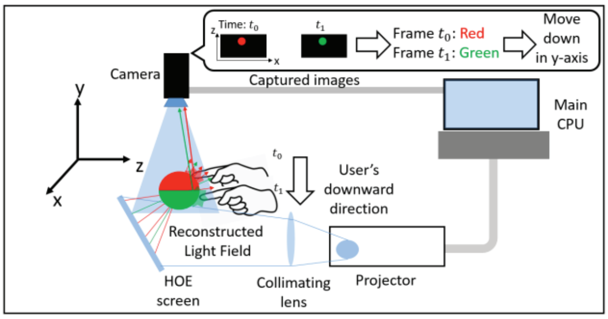

Figure 2. Detection of motion direction by measuring different colors along the y-axis

3 IMPLEMENTATION

3.1 Outline of the experimental system

The experimental setup is shown in fig. 1. It consists of a commercial-grade projector whose output is collimated by a lens. The projected light reaches a reflective HOE screen consisting of an array of small elementary holograms, each of which works as a convex mirror. Every pixel impinged on the screen by the projector diverges from the focal point of the holographic convex mirror. Therefore, the light-field can be created by the different rays diverging from the holographic mirrors [1]. The projector is used to modulate each pixel and change the content. The recording process is repeated in the same photopolymer with lasers of wavelengths of 633nm (red), 532nm (green), and 473nm (blue) to achieve a full-color display. An RGB camera located on top of the array detects the scattered light by the user's fingertip. The movement of the fingertip is then registered and sent to the main CPU to modify the projected light-field.

In the display part, to correctly align the projector's pixels and the HOE screen, an automatic registration method that projects binary sinusoidal patterns on the screen is used [3].

3.2 High-speed light-field computation and reproduction

Optimization of the algorithms presented in [1] and [2] is attempted to achieve faster processing. Fast matching of the elemental images with the calibration data to correctly project onto the HOE screen was achieved using Python and OpenCV, optimized with the Numba compiler [5]. The resulting implementation allowed us to pass from 100ms to 22ms (see tab. 1) in the process of matching the array of elemental images of 1920×1080 pixels to the positions of the 120×67 HOEs of the screen.

Table 1. Processing times of the two implementations considered

	Matlab [ms]	Python (+OpenCV and Numba) [ms]
Undistortion	100	22
Screen refresh	200	2

The original light-field is computed by rendering different views of a scene using Blender. The pixels of each view are spatially rearranged to generate the IP that is finally projected on the HOE screen (details can be consulted in [6]). Since the generation of this integral photography is computationally expensive, the movement of the light field is realized by displacing the elementary images with step sizes equivalent to the pixels entering each HOE.

3.3 Tracking in the camera plane (*xz* plane)

The light scattered by the user's fingertip (or stylus) interacting with the content (shown in fig.3) is detected by the RGB camera located on top of the HOE screen. An algorithm consisting of noise reduction, contrast stretching, and background subtraction is applied. A segmentation binary mask of the interaction area is acquired by applying an intensity threshold and morphological operations to the image with the background subtraction. Since we want to isolate the interaction point from the environmental light, we focus on obtaining only the point that reflects most of the scattered light. In this case, such point corresponds to the tip of the finger or stylus that we are using to interact. Therefore, the contour of the area of interest is obtained, and a box enclosing the ROI is created taking the pixel with the lowest *z* coordinate as reference.

Table 2. Processing times of our system

Process	Time [ms]
Image capture	20
Segmentation and contour extraction	70
Light field movement	20
Undistortion	22
Screen refresh	2
Total time per frame	134 (≈ 7 FPS)

3.4 Color measurement of the scattered light

The process to extract the ROI with enhancing the color information is shown in fig. 4. The pictures shown in fig.4 correspond to the ones obtained in the experiment displayed in fig.3

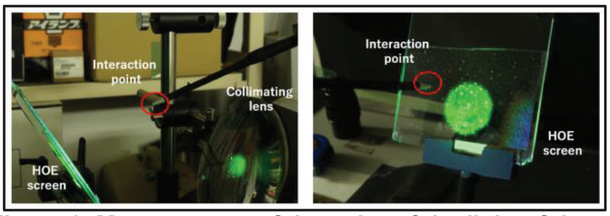

Figure 3. Measurement of the color of the light of the HOE screen scattered by an interaction point, lateral view (left) and front view (right)

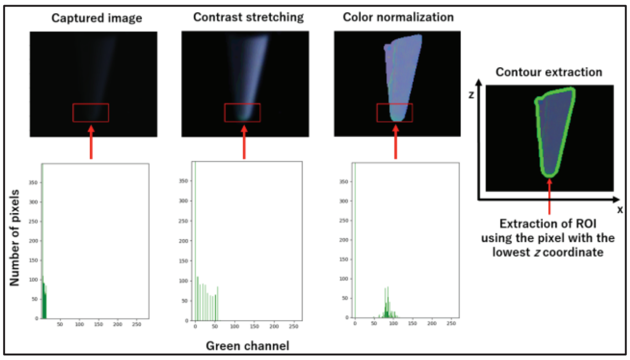

Figure 4. Process to extract the ROI and to enhance the color information

To be able to detect the color of the scattered light under room illumination, a color normalization step is applied to the image. This normalization consists in replacing the value of every channel (red, green, and blue) with its proportional value with respect to the total pixel count, namely:

$$\boldsymbol{C_N} = \frac{\boldsymbol{C_{cs}}}{\boldsymbol{R_{cs}} + \boldsymbol{G_{cs}} + \boldsymbol{B_{cs}}} \quad , \quad \boldsymbol{C} \in [\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}]$$

In the above equation, CS stands for "contrast stretched" and N means "normalized". This operation enhances the color signal that would otherwise be difficult to detect. The effect can be seen in the enhancement of the color signal in both the ROI and the histogram of the corresponding channel. This process was tested with 6 colors (red, green, blue, yellow, magenta, and cyan) as well as with the case in which the interaction point was not pointing to the colored light field (no color or "NC"). The distribution of the vector $\left[\frac{R}{R+G+B}, \frac{G}{R+G+B}\right]$ for each of the colors was plotted in RG space (see fig. 5). To be able to use the color information in the interaction, a color calibration should be performed beforehand. The calibrated data will be compared to the captured data to know what color has been touched. This comparison is done by computing the distance between the mean value of the previously saved data to the data of the picture captured during the interaction. Let $\overline{\boldsymbol{C}} = (C_{\hat{R}}, C_{\hat{G}})$ be the color values in the RG space of the captured image during the interaction and $\overline{\boldsymbol{\mu}}_{\boldsymbol{N}} = (\mu_{\hat{R}}, \mu_{\hat{G}})$ the mean value of the N color obtained from the distribution of fig 5.

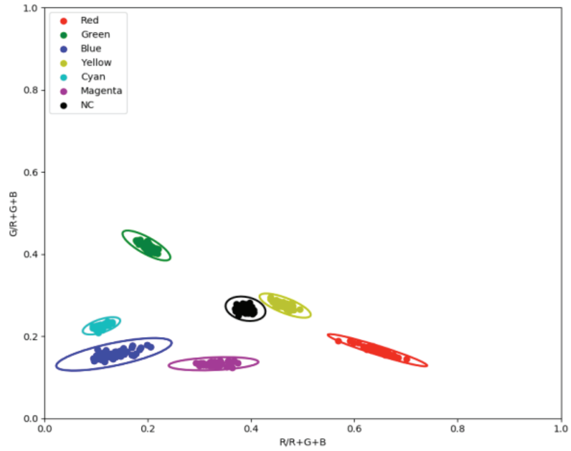

Figure 5. Distribution of normalized [R, G] vectors of the 6 colors and the no color case. A covariance ellipse is plotted for each distribution

We compared the performance of two distance functions: Euclidean distance and Mahalanobis distance, defined as:

Euclidean: $$\boldsymbol{d_E}(\overline{\boldsymbol{C}}, \overline{\boldsymbol{\mu}}) = \sqrt{(\boldsymbol{C_{\hat{R}}} - \boldsymbol{\mu_{\hat{R}}})^2 + (\boldsymbol{C_{\hat{G}}} - \boldsymbol{\mu_{\hat{G}}})^2}$$

Mahalanobis: $$\boldsymbol{d_M}(\overline{\boldsymbol{C}}, \overline{\boldsymbol{\mu}}) = \sqrt{(\boldsymbol{C_{\hat{R}}} - \boldsymbol{\mu_{\hat{R}}})^2 \boldsymbol{\Sigma}^{-1} (\boldsymbol{C_{\hat{G}}} - \boldsymbol{\mu_{\hat{G}}})^2}$$

$\boldsymbol{\Sigma}^{-1}$ denotes the inverse of the covariance matrix of the data. Once the color with the minimum distance to the $\overline{\boldsymbol{C}}$ vector is found (closest color, $\boldsymbol{C_{cl}}$), a criterion is applied to decide if the color corresponds to that category. In the case of the Euclidean distance, $\boldsymbol{d_E}(\overline{\boldsymbol{C}}, \overline{\boldsymbol{\mu}}) \leq \boldsymbol{2\sigma_{cl}}$, where $\boldsymbol{\sigma_{cl}}$ is the standard deviation of the $\boldsymbol{C_{cl}}$ distribution. For Mahalanobis distance, that criterion corresponds to $\boldsymbol{d_M}(\overline{\boldsymbol{C}}, \overline{\boldsymbol{\mu}}) \leq \boldsymbol{2}$. This is equivalent to ask for the captured vector $\overline{\boldsymbol{C}}$ to be inside an ellipse whose axes are twice the variance along the directions of maximum variance, defined by the eigenvectors of the covariance matrix of the color distribution. Such ellipses are the ones plotted in fig. 5.

4 RESULTS

4.1 Tracking in the *xz* plane ("drag" gesture)

The extraction of the ROI described in the previous section was used to track the interaction point along the *xz* plane. Localization of this interaction point in real-time allows us to have an implementation of an unlocking screen, demonstrating that this method can be used to interact with the reconstructed light field in real-time as shown in fig. 6.

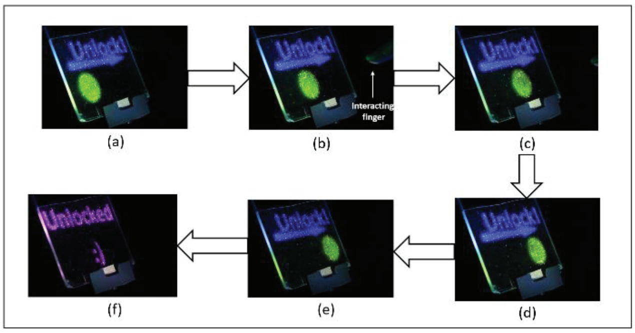

Figure 6. "Unlock" demonstration using the location of the interaction point in the *xz* plane

4.2 Detection of colors under dim illumination

The 6 colors plus the no color case were evaluated under both Euclidean and Mahalanobis distances. A sphere of each of the 6 colors was reconstructed, and the finger was used to interact with them for approximately 35 seconds (approximately 250 frames per color). Each frame was measured and compared to the previously saved values. The effectiveness rate was measured by dividing the correct color classification by the total number of captures (including successful, not identified, and erroneously classified). The Mahalanobis distance was seen to perform better than Euclidean distance in most cases (table 3).

Table 3. Effectiveness of color detection with Euclidean and Mahalanobis distances

Distance	Red	Green	Blue	Yellow	Cyan	Magenta	No color
Euclidean	**66 %**	30 %	91 %	39 %	40 %	94 %	79%
Mahalanobis	64 %	**71 %**	**99 %**	**79 %**	**56 %**	**98 %**	**99%**

4.3 Movement direction along the *y* axis

A bicolor sphere like the one shown in fig. 7 is reconstructed to prove the concept of using color to detect the movement direction along the y axis. When projected together, the colors interfere with each other, making the effectiveness of the detection fall to 60% in the case of green and 80% in the case of blue when using Mahalanobis distance. The movement along the y axis is realized under dim illumination. The reconstructed light field of the bicolor ball follows the finger of the user along the y axis as expected. However, movement errors also occur due to misclassification of the touched color, the touch color not being identified and errors in the sequence detection. The program is set to move the light field down whenever two consecutive frames be classified as [Blue, Green]. In occasions, for example, the frames might be classified as [Blue, Not Identified] or [Blue, Blue] instead, giving an erroneous movement.

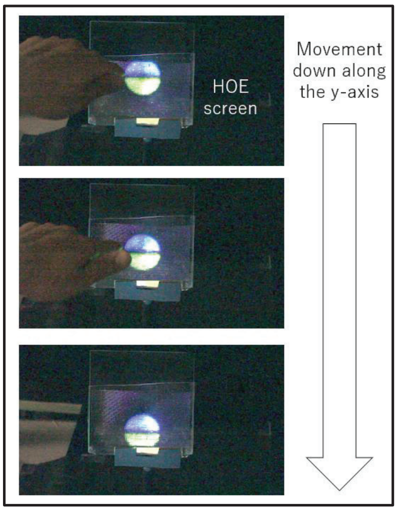

Figure 7. Color-aided movement along the y-axis

5 CONCLUSIONS

Real-time implementation of the 3D touch graphic user interface has been demonstrated. The optimized algorithm allows interactive finger motion detection and light-field image updates. Color detection applied to this interface is improved for dim illumination. Moreover, a color encoding method is proposed for the detection of movement direction along the axis perpendicular to the camera. Future improvements are the use of GPU oriented approaches to increase the frame rate in the light field display stage. To have a more robust color-based cue of motion, a pattern recognition approach in which the whole moving gesture is previously estimated could be of great help.

REFERENCES

[1] M. Yamaguchi and R. Higashida, "3D touchable holographic light-field display", Applied Optics, Vol. **55**-3, pp. 178-183 (2016)

[2] S. Sakurai, T. Nakamura and M. Yamaguchi, "The use of color in scattered light for a 3D touchable holographic light-field display", JSAP-OSA Joint Symposia, 13a-C301-4 (2016)

[3] T. Nakamura and M. Yamaguchi, "Rapid calibration of a projection-type holographic light-field display (…)", Applied Optics, **56**-34, pp. 9520-9525 (2017)

[4] S. Yamada, T. Kakue, T. Shimobaba and T. Ito, "Interactive holographic display based on finger gestures," Scientific Reports, 8:2010 (2018)

[5] https://numba.pydata.org/

[6] M. Martínez-Corral and B. Javidi, *Fundamentals of 3D images and displays: a tutorial*. Adv. In Opt. and Phot., **10**-3, OSA (2018)

[7] V.K. Kiran et al, "Exploring direct 3D interaction for full horizontal parallax light field displays using Leap Motion controller", Sensors, 15 (2015)

[8] L. Shaw et al, "Kinect the dots: 3D control of optical tweezers", Journal of Optics, IOP, (2013)