2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)

Improved Ultimate Link without Markers for Projective Transformation

Keiji Yamadera* and Michiharu Niimi†
* Graduate School of Computer Science and Systems Engineering
Kyushu Institute of Technology, Iizuka, Japan
E-mail: yamadera.keiji334@mail.kyutech.jp
† Faculty of Computer Science and Systems Engineering
Kyushu Institute of Technology, Iizuka, Japan
E-mail: niimi@ai.kyutech.ac.jp

Abstract—**In our previous work, we have proposed a method that links character string having a certain semantic meaning on papers in real world to the digital content in cyberspace, and call it "Ultimate Link." Ultimate Link adds additional information to characters by drawing colored circles on top of the characters without changing the shape of the characters. A URL can be generated by connecting additional information of semantically organized character strings from an image taken with a digital device. The conventional Ultimate Link drew markers to normalize the drawn circles (markers for projective transformation), similar to the finder pattern in a QR code. In order to achieve a more realistic application, it is desirable it. We report on an improved Ultimate Link that can handle complex multiple character strings. Specifically, improvements were made to enable extraction of information by identifying character strings with embedded additional information from sheets of paper printed with multiple character strings. As a result of experiments, we confirmed that information could be successfully embedded and extracted for both character strings in the target image.**

I. Introduction

Today, there is a growing movement toward "Digital Transformation (DX)," which utilizes information technology to enrich people's lives and transform existing values and frameworks in all areas of society. Then, DX is also being promoted in the field of education. For example, mobile information terminals such as tablets and notebooks are being introduced into educational settings to enable students to learn more effectively. However, several studies have shown the superiority of the learning effects of paper-based learning over information terminal-based learning[1]. We believe that if URLs can be attached to paper documents, it is possible to benefit from DX while efficiently deepening understanding of things.

In order to realize this, it is necessary to superimpose link information on a single semantic coherent string of text on the paper. For example, we assume that each rendered character of "Mt.Fuji" has additional information and is printed on papers, then the printed characters are captured by digital camera. From the captured image, a URL reconstructed by the additional information extracted from each rendered characters may be assumed to a web site of explaining Mt.Fuji.

As the method to do this, we have proposed Ultimate Link[2] that links semantically coherent character strings printed on paper with digital contents in cyberspace, allowing anyone to easily superimpose a large amount of information on the printed text without deforming the shape of the characters and extract information in real time by using standard cameras of common electronic devices. In [2], an image rendering only a single character string, which means there is no objects in the image without characters consisting a certain meaning, to be embedded with information was prepared, and then information was superimposed on the character and markers for the projective transformation necessary for image normalization during extraction were drawn around the character string to generate an information embedded image. Ultimate Link expresses bit information by arranging multiple colored circles on a font. In other words, the bit information corresponds to the color information. In this sense, we call a circle that retains bit information a bit-colored circle. In [2], the Ultimate Link system was evaluated through experiments of information superimposition and extraction using various types of character strings, such as multiple katakana-only character strings and multiple character strings containing both kanji and hiragana, in Japanese. However, a printed document containing only a single string of text is not realistic, and markers for projective transformation drawn around the string can disrupt the layout of the printed document. In this paper, we report on an improved Ultimate Link system , which aims to improve the Ultimate Link system for more realistic applications. Specifically, in the information embedding process, the circles conventionally used as bit-colored circles are given a function equivalent to projection markers. These circles are called position detection markers. In an image in which multiple character strings are rendered, information is embedded by drawing bit-color circles and position detection markers on the font only for the character strings in which the information is to be embedded. In the information extraction process, the printed information-embedded image is captured by a smartphone camera so that the entire image is captured, position detection markers are detected from the captured image, and information is extracted by identifying the character string in which the information is embedded.

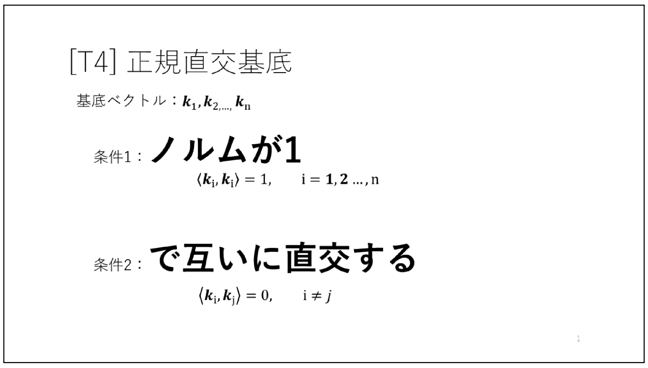


Fig. 1: Example of image for embedding information

In the following sections of this paper, an overview of Ultimate Link and a method to realize an improved Ultimate Link are described in Section II, experiments are conducted in Section III, discussions are given in Section III-D. Finally, in Section IV, we summarize the results and discuss future issues.

II. Ultimate Link

Ultimate Link is a method that links a certain semantic cohesion in the real world to accompanying/supplementary information in cyberspace. This method is consisting of two parts : superimposing (embedding) and extracting binary data on rendered characters. In embedding process, information is embedded by drawing bit-colored circles on the text that correspond to color information and bit information in order to superimpose bit information on the text. On the other hand, in extracting process, information is extracted by detecting bit-colored circles on the text and arranging them in an appropriate order. Each process is described below.

A. Embedding Process

The embedding process follows the flow below.

(1) Character recognition
(2) Character segmentation
(3) Character image thinning
(4) Pseudo-stroke order calculation
(5) Grid points assignment
(6) Grid points for position detection assignment
(7) Bit-colored circle drawing
(8) Final processing

We assume that the improved Ultimate Link deals with an image data that multiple Japanese characters are rendered, which is called a cover image. There are several character strings having certain meaning in the characters, and Fig.1 is an example of this. We explain each procedure listed above in detail using the example cover image. In addition, we use a character string : "ノルムが 1" (in Japanese) as a target string, which is a character string for information to be embedded, and denote it as TS. Note that TS is assumed to be a single line and written horizontally, so it is not possible to select TS that extends over multiple lines or a character string that is written vertically. In addition, we need to define the information to be embedded, which is a shortened URL.

(1) Character recognition
Character recognition by Vision API[3], which is an image recognition service provided by Google, is performed to the entire cover image. Firstly, the entire characters in cover image are recognized. Then, only the part of TS is extracted from the character recognition result.

(2) Character segmentation
TS consisting of multiple characters is divided into individual characters. In [2], this was performed by image processing for entire cover image, but it sometime failed when the cover image contains characters other than TS. Therefore, we use the response of the character recognition function of Vision API which includes the rectangular area information of each recognized character.

(3) Character image thinning Before performing the thinning of each segmented character image, the erosion is performed for the characters. After erosion of each character, we perform thinning to it. A binary image with a line width of 1 is obtained from this process. This information is used for the position of drawing circles.

(4) Pseudo-stroke order calculation
Ultimate Link expresses bit information by drawing bit-colored circles on characters. Here we introduce the concept of "pseudo-stroke order" of characters to define the order of bit information, i.e., we define the order in which bit-colored circles are drawn on the characters.
The pseudo-stroke order is calculated by tracing the order of foreground pixels from an image of a thinned character image. This order based on the contour tracking method and is calculated by the following procedure.
Step 1 : The first pixel which is found in the foreground by raster scanning is set to the target pixel.
Step 2 : The next target pixel is search by a counter-clockwise direction within the adjacent pixel of the target pixel.
Step 3 : Until there are no more adjacent pixels, Step 2 is repeated, and then return to Step 1.
Step 4 : Until all foreground pixels is searched, steps of 1 through 3 is repeated.
The pseudo-stroke order is calculated by the above procedure. The obtained pseudo-stroke order is saved for each character.

(5) Grid points assignment
By using pseudo-stroke order, we can scan the foreground pixels of thinning image with unique order. Following this order, we draw a circle of radius r with d as the distance between circles. In some cases, the circles may overlap depending on the character shape and scanning order. In such cases, we skip the foreground pixel and then we search the next one that the drawn circle is not overlapped with pseudo-stroke order. After this procedure, we can determine the center of circle on the foreground pixels of thinning image, which is called "grid point." Fig.2 shows

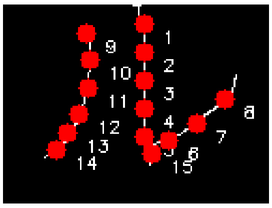
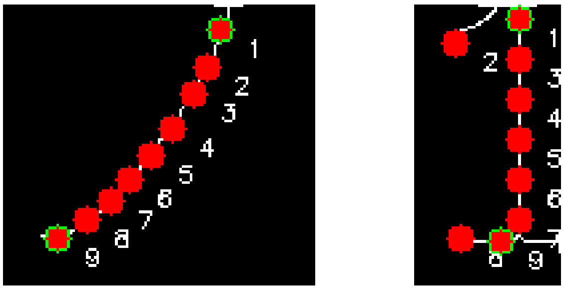


Fig. 2: Example of grid points



(a) The first character of TS **(b)** The last character of TS

Fig. 3: Grid points for position detection (2 for each character)

an example of drawing grid points.

(6) Grid points for position detection assignment
In order to identify the region of TS with the information is embedded in the extraction process, we define position detection markers on grid points. The markers are consisted of four grid points obtained by the first and last grid points of the first character and that of the last character. In the TS example, the first and last grid points for "ノ" and the first and last grid points for "1." Fig.3 shows the results of the grid points for position detection. In the figure, the grid points for position detection are circled by green circles.

(7) Bit-colored circle drawing
For the obtained grid points, colored circles which color is selected by the embedded information are drawn. We call it a bit-colored circle. The position at which bit-colored circle is drawn is randomly selected from among the grid points with a fixed random seed according to the length of the embedding information. One circle represents two bits (four patterns), and five colors should be in total, since colors are also needed for markers for position detection. In the HSV color space, especially in the OpenCV COLOR BGR2HSV FULL color space, each HSV value is described as 0 to 255, and within this space, five colors are set by changing the H value as shown in Table I, where S = 255 and V = 255. The reason why we chose these colors is that it makes the proposed method easy to implement. The colors in Table I are, from top to bottom, markers for position detection and embedded bit information: $(00)_2$, $(01)_2$, $(10)_2$ and $(11)_2$.

(8) Final processing
As bit-colored circles are superimposed on rendered character images, we replace original characters with it. In addition, we draw position detection markers on the first and last characters at the grid points for position detection

TABLE I: Colors for grid points/detection markers

H value	Actual color	Role
16		detection marker
46		00
106		01
166		10
226		11

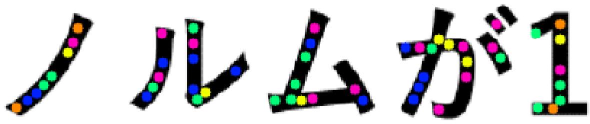


Fig. 4: Example of information embedding

obtained in (7). Fig.4 shows an example of a string with embedded information. Multiple colored circles are drawn on the characters according to the embedded information.

B. Extraction Process

An image with embedded information created in Section II-A is printed on low-quality paper, and the embedded information is extracted from an image of the printed paper captured by a camera of a digital device such as a smartphone (hereinafter referred to as "captured image").

The extraction process follows the flow below.

(1) Search the position of a string with embedded information in the captured image
(2) Image binarization
(3) Character recognition
(4) Character segmentation
(5) Bit-colored circle detection
(6) Generate temporal character fonts
(7) Drawing the reference grid points on the temporal character font
(8) Drawing white circles at the center coordinates of the detected bit-colored circles
(9) Alignment based on detected bit-colored circles and reference grid points.
(10) Nearest circle detection
(11) Bit information extraction

The following explanation of each procedure uses, as an example, an image printed with information embedded in " ノルムが 1" created in Section II-A and captured with a smartphone camera.

(1) Search the position of a string with embedded information in the captured image
We search the position of the string with embedded information in the captured image. The position of a character string in which information is embedded is identified by detecting the position detection markers drawn on the first and last characters of the character string. The position detection markers are detected by generating a mask image in the color of the position detection marker set in advance in the HSV color space, expanding the mask image, and

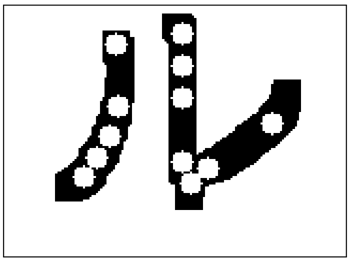


Fig. 5: Drawing reference grid points

Fig. 6: Drawing detected circles

detecting circles by Hough transform on the expanded mask image. This identifies the coordinates of the four position detection marker points and fills in the area outside of the four points with white color. This process identifies the character string in which the information is embedded because only the target character string remains in the image, and all other character strings are filled with white.

(2) Image binarization
After processing (1), binarization is performed using the predefined colors in HSV color space to improve the accuracy of character recognition.

(3) Character recognition
Character recognition is performed on the binarized image created in (2). Since only the character string to be embedded is available in the image, the result of character recognition for the entire image can be used as that of the target string recognition.

(4) Character segmentation
The string is divided into individual characters in the same way as during the embedding process.

(5) Bit-colored circle detection
The method for detecting bit-colored circles is almost the same as the method for position detection markers in (1), but an opening process is applied to the expanded mask image before the Hough transform to remove noise.

(6) Generate temporal character fonts
The pseudo-stroke order information saved during the embedding process generates a binary image with a line width of 1. Base on this information, we make a temporal font image by expanding it, which means like a reverse step of thinning image. The image generated by this process is called a temporal character font.

(7) Drawing the reference grid points on the temporal character font
The grid points are obtained from the pseudo-stroke order information saved at the time of embedding, and the grid points used to embed the information are obtained using the same random seed as that used for embedding. These grid points are reference grid points. These reference grid points are drawn as white circles on the temporal character font. Fig. 5 shows an example of an image with reference grid points drawn.

(8) Drawing detected bit-colored circles
The center coordinates of the bit-colored circle detected in (5) are drawn as a white circle as shown in Fig. 6.

(9) Alignment based on detected bit-colored circles and reference grid points
Based on Fig. 5 and Fig. 6, the detected bit-colored circle position is aligned to the reference grid point position. This procedure is one of the most important steps in the extraction process. In (5), the bit information embedded in the string itself was extracted by detecting bit color circles, but the order of the bit information was not recovered. The order of the bit information is recovered by finding the reference grid point closest to the center coordinates of the detected bit-colored circle in (10), described below, and arranging the bit information corresponding to the color of the bit-colored circle in the order of the reference grid points in (11). In this case, the coordinates of the bit-colored circle obtained in (5) are obtained from the coordinate system of the image from which the string was segmented from the captured image, and thus no normalization to the coordinate system of the reference grid point is performed. In this paper, normalization is performed by aligning one character at a time using the coordinates of the bit-colored circle detected for each character and the coordinates of the reference grid point. Therefore, if the alignment is not performed accurately, the mapping between the detected bit-color circle and the reference grid point in (10) and (11) fails, and the embedded information cannot be recovered. Thus, this alignment procedure is an important step in the extraction process.
The alignment method is based on Procrustes analysis[4]. Procrustes analysis is a method of superimposing two point groups that correspond to each other in such a way that the squared error between the point groups is minimized. Thus, we apply this to the coordinates of the detected bit-colored circles and reference grid points. Therefore the transformation matrix that minimizes the squared error between the coordinates of the detected bit-colored circles and the coordinates of the reference grid points can be calculated, then the matrix is applied to the coordinates of the detected bit-colored circles and superimposed on the coordinates of the reference grid points.

(10) Nearest circle detection
The nearest neighbor circle is obtained by finding the

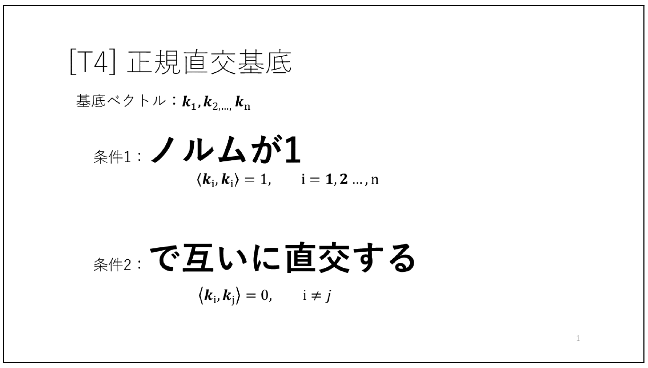
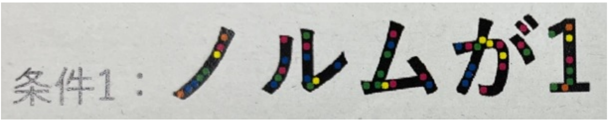
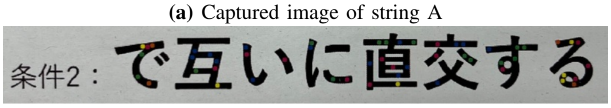


Fig. 7: Image for embedding information



(a) Captured image of string A



(b) Captured image of string B

Fig. 8: Captured images

center coordinates of the detected bit-colored circle that has been aligned and the center coordinates of the nearest reference grid point.

(11) Bit information extraction
The bit sequences corresponding to the colors of the nearest neighbor circles are ordered. In this way, the embedded information is extracted.

III. Experiments

Two experiments were conducted to evaluate the improved Ultimate Link proposed in Section II. In the first experiment referred as Experiment I , we conducted information embedding and extraction processing experiments on two Japanese character strings A: “ノルムが 1” and B: “で互いに直交する” in an image containing multiple character strings as shown in Fig.7. The second experiment referred as Experiment Ⅱ was conducted to investigate the performance of the Ultimate Link system. Specifically, experiments were conducted by changing the distance between the camera and an image printed on a paper surface containing embedded information denoted as “target paper” , to investigate whether the extraction process is possible or not by changing the conditions under which the target paper is captured.

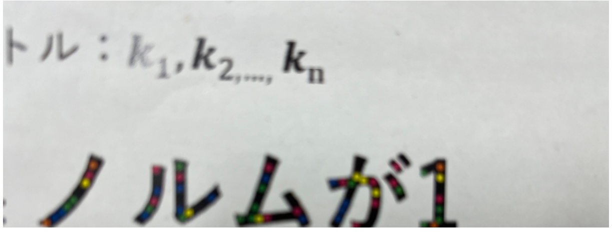
A. Common settings for each experiment

The target image is a 2339 × 1315 binary image, and two of the target strings are rendered in bold gothic font with a font size of 54pt. In the goal of our research, we aim to be able to use a variety of fonts and character sizes. However, in this experiments, we chose the font type and size so that it makes the proposed method easy to implement without considering additional technical improvements.

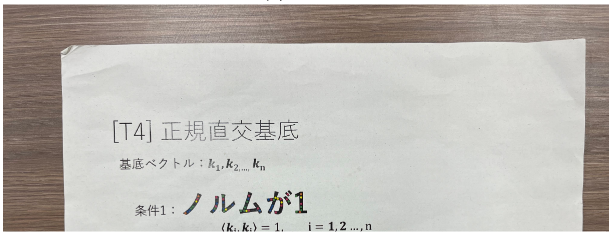
We prepared shortened URL E_1 and E_2 as embedding information, and embedded E_1 : “bit.ly/4aO6jiU” in string A and E_2: “bit.ly/3U5rYgp” in string B.

Embedding was performed on two target character strings, and images with embedded information were printed on A4 paper and captured with a smartphone camera. The smartphone used was an iPhone 13. Fig. 8(a) and Fig. 8(b) are partially enlarged images, and the full image size is 4032 × 3024.

The system was implemented using Windows 10 (CPU: Intel(R) Core(TM) i7-8565U, memory: 8GB), Python 3.11 as the programming language, and OpenCV as the image data processing environment.



(a) Term1



(b) Term4

Fig. 9: Captured images for Term1 and Term4

B. Experiment I

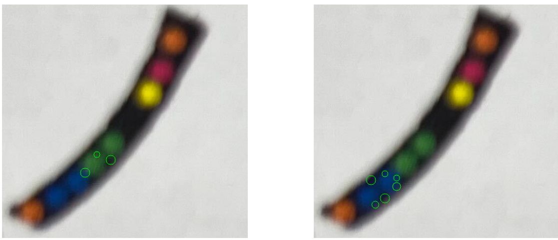
We confirmed that both strings A and B were successfully embedded and extracted, and that the URL could be recovered. The extraction process took 3.11 seconds for the string A and 3.26 seconds for the string B.

C. Experiment Ⅱ

This experiment was conducted on the extraction process by capturing the target paper from the front and changing the distance between the target paper and the camera to confirm whether the information was extracted correctly. The string A was chosen as the target string.

The distance between the target paper and the smartphone camera was set under the following four conditions : Term1, Term2, Term3 and Term4 that the distance between the target paper and the camera is about 10 cm, 15 cm, 20 cm and 30 cm.

Fig. 9 shows two examples of captured image for Term1 and Term4. The experimental results confirmed that the information was extracted correctly under the terms 2 and 3, where the camera focused on the target character string and no other



(a) Example A **(b)** Example B

Fig. 10: Circle detection failure

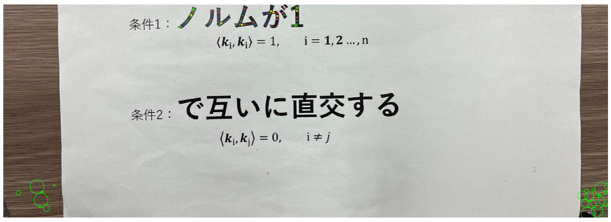


Fig. 11: Failure to identify the embedding position

images other than the target paper were captured within the camera's angle of view.

In Term 1, as shown in Fig.9(a), the camera was not focused on the target character string and it was blurred, so the detection of the colored circles failed as shown in Fig.10, and the information was not extracted correctly. In Term 4, as shown in Fig.9(b), there are objects other than the target paper in the camera's angle of view, so as shown in Fig.11, many false positives occurred when detecting the position detection marker, and parts other than the target character string remained, so the information was not extracted correctly.

D. Discussion

In Experiment I, we confirmed that both strings A and B successfully embed information. This is because the high character recognition accuracy of the Vision API's character recognition function enables it to recognize characters across the entire image even when there are multiple character strings, extract the recognition results for the target character string from the overall results, determine the rectangular area for each character, divide the character string, and perform information embedding processing for each character. In the extraction process, both strings A and B successfully identified the location of the string in which the information was embedded. This is because the position detection markers detected during the extraction process to identify the position of the character string in which the information was embedded from the captured image. Furthermore, we confirmed that both strings A and B can be correctly extracted without markers for projective transformation. This means that the alignment by Procrustes analysis based on the reference grid points and the detected bit-colored circles eliminates the need to normalize the image by a projective transformation using markers for the projective transformation.

In Experiment Ⅱ, the results show that information was successfully extracted at a distance of approximately 15 to 20 cm, which is the distance at which the camera was able to focus on the target character string and at which no objects other than the target paper appeared within the angle of view of the camera. The extraction process failed at distances of about 10 cm, where the camera could not focus on the target string and the image was blurred, and at about 30 cm, where the camera's angle of view included objects other than the target paper. Based on the results of this experiment, it is considered that a distance of approximately 15 to 20 cm between the target paper and the camera is appropriate for the current extraction processing system.

IV. Conclusions

In this paper, we propose an improved Ultimate Link method for realistic applications that can embed and extract information in target character strings in character images containing multiple character strings without using markers for projective transformation. Experimental results show that the proposed method succeeds in embedding and extracting information in two target character strings. We also investigated the characteristics of the Ultimate Link extraction process by changing the conditions under which the target paper was captured. In this experiment, we conducted the experiment on letters printed on paper with a white background, so it is necessary to conduct the experiment on posters and other postings with a colored background. In addition, since the experiment was conducted under a single lighting condition, i.e., under fluorescent light, it is necessary to conduct the experiment under multiple lighting conditions, such as under sunlight or in a dark room.

In the method used in this experiment, the images captured by the smartphone camera were once imported to a PC for extraction processing. For future practicality, we think it is necessary to develop an application that can complete the information extraction process and display a Web page within a smartphone and evaluate its practicality.

Acknowledgment

This research was supported by JSPS Grant-in-Aid for Scientific Research JP 19K12879.

References

[1] J. Noyes and K. Garland, "VDT versus paper-based text: Reply to mayes, sims and koonce," *International Journal of Industrial Ergonomics*, vol. 31, no. 6, pp. 411–423, 2003.

[2] K. Yamadera and M. Niimi, "Ultimatelink Between Characters Having a Certain Meaning in Physical Space to URL in Cyberspace with Robust Print and Scan," in *Proceedings of the 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, Oct. 2023, pp. 514–518.

[3] *Vision API*, https://cloud.google.com/vision?hl=ja.

[4] S. P. H., "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1996.