An Image based Automatic 2D:4D Digit Ratio Measurement Procedure for Smart City Health and Business Applications

Frode Eika Sandnes[1,2] and Levent Neyse[3]

[1]Oslo and Akershus Univeristty College of Applied Sciences, Oslo, Norway

[2]Westerdals Oslo School of Art, Communication and Technology, Oslo, Norway

[3]Kiel Institute for the World Economy, Kiel, Germany

Abstract: 2D:4D digit ratios are used for several health and business related applications. Currently, digit ratios are measured manually. This study proposes an automatic digit ratio measurement approach that can be used in the context of smart city healthcare and business applications. Smart city healthcare needs to be founded on the principles of self-service and independence. The proposed approach assumes that an image of the hands of a user is acquired using some imaging device. First, the hands are separated from the background. Next, the hand outline is traced. The hand outlines are used to identify points of interest that are used to measure the finger lengths and digit ratios. Experimental results are promising, but further research is needed before the approach can be deployed in real-world settings.

1. Introduction

Digit ratio measurements are used in several avenues of research within healthcare, medicine and psychology [1, 2, 3]. The digit ratio is defined as the ratio of the index finger length (D2) divided by the ring ringer length (D4) and the ratio is also often referred to as the 2D:4D ratio. The 2D:4D ratio can be used as a crude indication of exposure to prenatal sex hormones [4]. The lengths are typically measured from the tip of the fingers to the basal crease of the finger where the fingers join the palm.

Still, such measurements are acquired manually from photocopies of the hands, or flatbed scans. However, recently a few experimental approaches have been proposed for the automatic measurement of the 2D:4D-ratio [5, 6, 7, 8]. The first ever-reported attempt [5] was designed for one hand measurement of the hand using a mobile handset camera. The idea was to use the built in camera flash to make it easy to separate the hand from the background since the eliminated hand is much brighter than the background. Based on successful binarization of the hand images an outline was extracted and converted to angular coordinates. The derivatives of the angular representations were used as basis for determining the feature points of interest, that is, the fingertips and finger cervices. This approach was designed for one hand, as the other is used to handle the mobile device. Moreover, it was based on the assumption that the fingers are sufficiently spread out and that the hand constitute a plane perpendicular to the viewing angle of the camera.

A two-hand approach for flatbed scans intended for finger ratio research was proposed [6] and later with an improved binarizer [7]. This binarization approach was algorithmically complex and vulnerable to certain background configurations. A different initiative to automate digit ratio measurements used so called colour structure codes to separate the hand from the background has been proposed [8], however, the reported results are limited. Although several advanced binarization algorithms are described in the literature [9, 10] it seems necessary with domain specific algorithms. The vast literature on skin detection is testament to this [11, 12].

Other research that share commonalities with automatic finger ratio extraction has been conducted into gesture recognition [13, 14]. However, the emphasis of gesture recognition is to classify hand postures while the objective of finger ratio algorithms is to make accurate and precise finger length measurements.

This paper describes a method that builds on the methods reported in [6, 7] were binarized images are scanned from one side to the other to construct the finger outlines for the two hands. The two hand scans are assumed to constrain the hand orientation and position. These constraints simplify the recognition process.

However, it is difficult to fully spread the fingers of two large hands on the glass plates of small A4 scanners. Moreover, when the hand is pressed against the scanner glass the fingers flatten and touch each other. This study therefore proposes a robust method for measuring digit ratios automatically to overcome many of the problems that present with manual measurements [15] such as non-standard and divergent measurement procedures and the presence of human bias when measuring ambiguous points of interest. Although it extends the methods in [6, 7] the method proposed herein is simplified and more robust as it relies on a standard clustering technique to obtain binarizing thresholds dynamically instead of finding these through manual experimentation. It is thus able to adapt to a wider range of lighting conditions, backgrounds and skin colours than the previous algorithms [6, 7]. Moreover, this study also provides results with the algorithm on a larger set of hand scans.

The proposed algorithm could for instance be used for the unassisted capture of the digit ratio and hand measurements of individuals in various smart city public locations, shops or homes, using standard imaging technologies such as camera enabled self-service kiosks and flatbed scanners. Possible applications include providing customers with tailor made products and services while achieving increased privacy and increased measurement accuracy. Sensitive information such as fingertip patterns [16] can be immediately discarded and thus not transmitted or stored electronically.

2. Method

The 2D:4D finger ratio measurements are achieved by first binarizing the scanned image of the hands. Then the outline of the hands is generated based on the binarized image. Next, the curve of the hand image is analysed to determine the measurement points that serve as the basis for the 2D:4D measurement.

2.1 Hand background separation

A binarization operator separates the hand from the background. The separation procedure utilizes the characteristic difference that a hand is saturated to some degree and the background is completely unsaturated. However, the background may vary in brightness from black, via shades of grey to white due to shadows and characteristics of the scanner hardware. The saturation level is therefore used to classify each pixel in the image as hand or background. A simple measure of saturation can be calculated by projecting the pixel in RGB space onto the plane defined by the normal going along the diagonal of the colour cube from black to white, namely

$$x = r - b \tag{1}$$

$$y = g - b \tag{2}$$

The saturation s is the distance from the projected point to the origin or the colour cube

$$s = \sqrt{x^2 + y^2} \tag{3}$$

This is used to define a saturation function $d(r, g, b)$

$$d(r,g,b) = \sqrt{(r-b)^2 + (g-b)^2} \tag{4}$$

The saturation is calculated for a subset of regularly sampled pixels, and these saturation values are clustered as either hand or background using the K-means++

clustering algorithm [17]. The K-means++ algorithm is an extension of the well-known K-means algorithm improved with a randomized seeding technique.

The binarizing threshold is computed as the midpoint between the maximum value of the less saturated background cluster and the minimum value of the more saturated hand cluster, that is

$$T_1 = \frac{1}{2}\left[\max(background) + \min(hand)\right] \tag{5}$$

Each pixel is then binarized according to the threshold T_1, that is

$$binary(x,y) = \begin{cases} 1, & T < d(image(x,y)) \\ 0, & otherwise \end{cases} \tag{6}$$

Here, *image*(x, y) is the image pixel at pixel position x, y, *binary*(x, y) is the binary pixel at position x, y and $d(p)$ is the saturation function.

Next, a second pass is performed to emphasize the divide between fingers. There may not be any background pixels between these fingers and the darkness values are therefore used instead. Only the pixels classified as hand pixels during the first pass are processed to compute a second threshold T_2. This threshold is found by clustering a regularly sampled set of hand pixels into hand and background according to pixel intensity using the K-means++ algorithm. The following intensity function is used

$$I(r,g,b) = \frac{1}{3}(r + g + b) \tag{7}$$

All the hand pixels found through the first pass is re-classified as background pixels if their intensity is below the threshold T_2. This second pass ensures that the dark cracks between fingers sticking close together are classified as background even though their pixels are saturated.

2.2 *Landscape portrait adjustments*

The finger-ratio detection algorithm assumes that the two-hand finger-scans are oriented such that all the fingers point rightwards. To ensure that images satisfy these assumptions a hand orientation step and possible rotation step are employed.

First, a check is performed to determine if the image is in portrait orientation. If the image is in landscape orientation it is rotated 90 degrees to ensure that it is in portrait orientation. For an image to be in portrait orientation the height must be larger than the width.

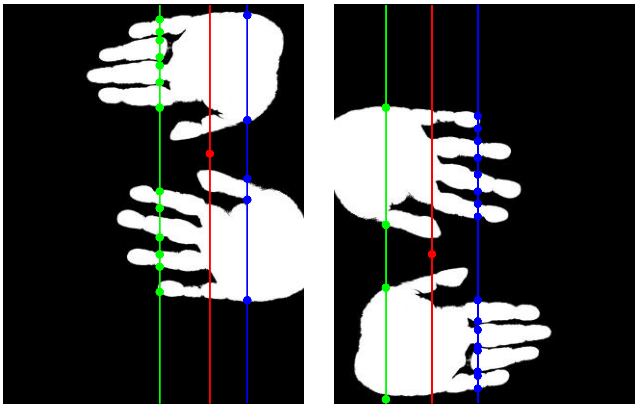

Fig. 1. Determining the finger pointing direction.

2.3 Micro image rotation

Next, the tilt of the two hands is determined. The tilt detection is performed in several steps. First, the centroid of the hands is computed by finding the midpoint of all the hand pixels. That is:

$$[x_c, y_c] = \left[\frac{1}{N}\sum_{i=1}^{N} x_i, \frac{1}{N}\sum_{i=1}^{N} y_i\right] \tag{8}$$

Here x_i,y_i are all the pixels labelled to be part of the hands and N is the number of hand pixels. This centroid is assumed to lie between the two hands. Therefore, the hand image is divided into two halves separated by the vertical line y_c. The centroid computation is thus repeated for the upper half and the lower half, respectively, giving the two new centres [x_{top}, $\mathrm{y}_{\mathrm{top}}$] and [$x_{bottom}$, y_{bottom}]. The angle of tilt A is then

$$A = 180 - a\tan 2\left(x_{bottom} - x_{top}, y_{top} - y_{bottom}\right) \tag{9}$$

To reduce the computational load it is only necessary to consider a subset of the image pixels in order to get a sufficiently accurate result. A total of 100×100 regularly spaced pixels were sampled in the above computation. Moreover, the comparatively expensive image rotation step is only performed if the tilt is larger than 5 degrees as small angles have little effect on subsequent processing.

2.4 Macro Image rotation

The final rotation detection step is to determine whether the fingers are pointing left or right. If the fingers are pointing left the image must be rotated by 180 degrees in order for the finger to point right.

Fingers are detected by scanning the image vertically. The finger side will lead to more transitions between background and hand pixels compared to the palm side (see Fig. 1).

The finger direction detection involves dividing the image into two halves along the vertical line x_c, that is, the vertical midpoint of the hand. For each side the binarized image is scanned vertically from top to bottom from one side to the other. If two consecutive pixels are different the difference is counted. After this step, there will be a sum of difference for each respective image half, namely left and right.

If the sum of differences on the right side is larger than the sum of differences on the left side, it is an indication that the finger are pointing rightwards. However, if the sum of differences for the left side is larger than that of the right side, the image needs to be rotated 180 degrees, that is

$$left > right \tag{10}$$

To speed up computation only a subset of 100×100 pixels was considered.

2.5 Noise removal

To further eliminate noise in the binarized image and enhance the hand contour a median filter is used. Experimentation reveals that a median filter with a size similar to 1% of the image gives good results. For example a 31×31 median filter was used for images with a resolution of 2548×3508 pixels. To achieve computational efficiency an adaptation of a generalised median filter was implemented. It comprises a sliding window that is moved across all the images of the image. For each pixel assessed the majority of pixels determine the final pixel value. For example, if the 961 pixels of a 31×31 window contains 481 or more white pixels the current pixel is set to white, otherwise it is set to black.

2.6 Hand outline extraction

The outline of the hand is obtained by scanning the image from right to left with one vertical line at a time from top to bottom (starting at y=0). That is, the image is scanned in the direction from the end of the image towards the fingertips. Once a pixel change is detected a potential finger candidate is found and the start point y_{start} is recorded. That is if

$$binary(x, y_i) \neq binary(x, y_{i-1}) \tag{11}$$

The end of the finger candidate is detected once the pixel changes back to the background colour which point y_{end} is recorded.

A check is performed to see if the finger candidate segment $[y_{start}, y_{end}]$ is the result of noise or not by determining if the length of the segment is above a threshold W. The segment start and end-points that pass this test are stored in a list M_x. The threshold W was set to

$$W = \frac{y_{\max}}{100} \tag{12}$$

Initially, the number of segments resulting from the vertical scan is zero. Once the tip of the first finger is detected the number of segments increases to 2. As we scan downwards the x-axis the number of segments will increase to 16, which means that all the fingers beside the thumb have been detected. Next, the number of segments will decrease as the scan reaches the crevices of the fingers, and of course increase once the thumbs are detected. In other words, an interesting finger feature point is encountered once there is a change in the number of segments resulting from a scan. The second part of the hand outline extraction involves combining the traces stored in M into two continuous hand outline curves. These continuous curves are found by connecting neighbouring points in M.

2.7 Finger vector detection

The finger vectors are computed using fine-tuned fingertip points D and the crease R of each finger. Fine-tuning is needed since the vertical scan line may have a different angle to the fingertip tangent. First, lines passing through the middle of each finger are found by using the first quartile point and the third quartile point on the curve going from the crevice point and fingertip point defining the curve segment as basis for the midline. The first quartile point is found by taking the ¼ point along the outline trace between the two neighbouring points of interest (see Fig. 2). Similarly, the third quartile points of the finger are found by taking the 3/4 point between the estimated fingertip point and the two neighbouring root points.

The finger midline is defined by the two midpoints between the two points on the ¼ along the sides of the finger, and between the two points ¾ along the sides of the finger (see Fig. 2). The crevice and fingertip points are then updated according to where midlines cross the hand outline curve.

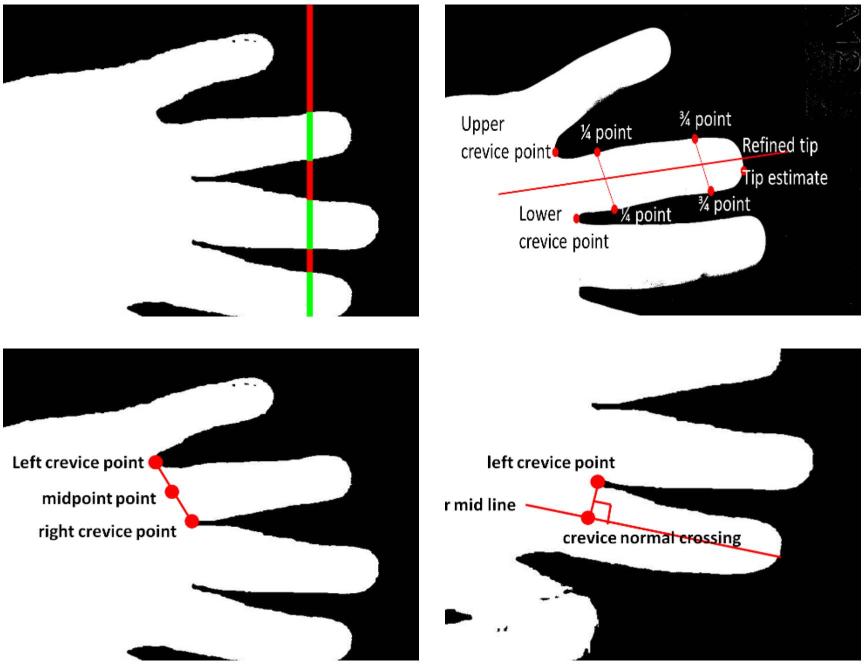

Fig. 2. Scanning for the hand outlines (top left), fine-tuning the fingertip point (top right), determining the finger root points of the ring finger using the midpoint (bottom left) and detecting the index finger root using the normal (bottom right).

The creases R of the ring fingers are simply found as the midpoint between the two neighbouring crevice points at the left and right side of the finger. The crease point for the index finger has only one reliable crevice point. The estimate of the crease is therefore defined as the point where the normal of the midline intersects the crevice (see Fig. 2).

2.8 Digit ratio computation

Once reliable finger crease points R and fingertip points D have been determined, the finger ratio for finger i and j is simply the ratio of the two finger lengths defined by the vector going from the finger crease R to the fingertip D:

$$DR(i,j) = \frac{|R_i - D_i|}{|R_j - D_j|} \tag{22}$$

Note that the digit ratio is computed for the left and the right hands, respectively.

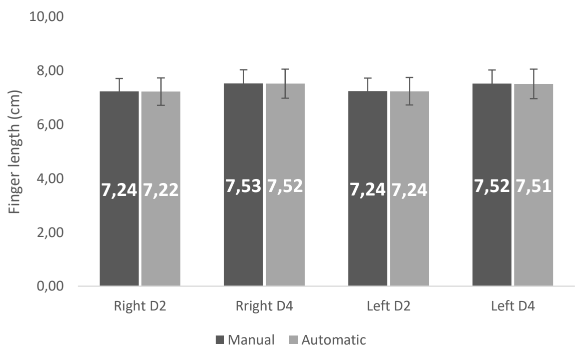

Fig. 3. Finger length measurements. Error bars shows standard deviation.

3. Experimental evaluation

The method proposed herein was tested with 284 high quality hand images acquired using flatbed scanners. The scans were acquired by different researchers with different equipment, different setups and a diverse range of individuals. Common to all scans were that both hands of the individual were pressed palm-down against the scanner glass. The index and ring finger lengths for all the hand images were manually measured using the standardized procedures outlined in [14] as reference.

The algorithm was implemented in Java using a custom-made image analysis library. The Apache Commons machine learning library was used for clustering. The results were run on a high-spec Windows laptop. The unoptimized code took about three hours to process the images, that is, less than one minute per image.

The result revealed that the algorithm was unable to process 44 of the scans (15.5%). A manual inspection of the problematic images revealed that the main reasons were inability to separate the hands from the background or inability to acquire all the points of interest. For example, some of the images had backgrounds with a similar colour to the skin colour of the hands, which the binarizing algorithm was unable to handle. Some of the scans also showed hands where the fingers where facing each other instead of pointing in one direction. Clearly, the scanline procedure is unable to capture the points of interest successfully if the fingers are not close to parallel with respect to each other.

The algorithm was able to acquire finger length measurements for the remaining 84.5% of the scans. A filtering step was performed to discard measurements that were out of range. The manual measurements revealed that the finger length varied from 5.92 cm to 9.07 cm. Therefore, results outside the more generous range of 5.5 cm and 9.5 cm were discarded. In total, 119 images (42.3%) had to be discarded due to their out of range values since these would give unreliable digit rates. The remaining 121 scans (42.6%) were therefore used in the subsequent analysis.

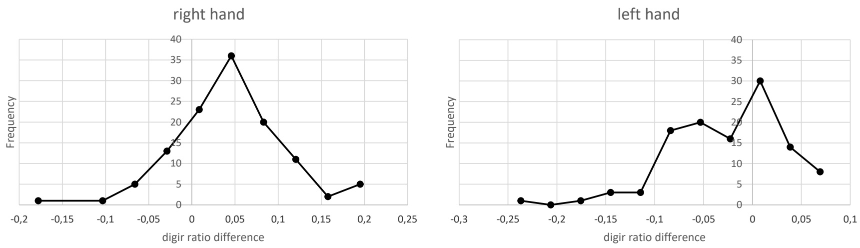

Fig. 4. Histograms of the differences between the manually and automatically measured digit ratios for the right and the left hand.

Fig. 3 shows the mean finger lengths acquired using manual measurements and the automatic procedure. The figure shows the two sets of measurements are similar, but the mean does not show discrepancies that exist for individual measurements. Moreover, t-tests revealed significant differences between the manually and automatically acquired digit ratios for the left ($t(119) = 1.98$, $p < .001$) and the right hands ($t(119) = 1.98$, $p < .001$).

To more closely explore the discrepancies between the manual and the automatic procedures the histograms of the differences between the digit ratios for the left and the right hands are plotted in Fig. 4. The histograms reveal large differences between the manually acquired digit ratios and those acquired automatically. The width of the distribution reveals that variations occur with more than 0.1, which is above the desired level for the method to be considered accurate. Although the histograms centre around zero, it is a bias towards the positive side for both hands. This suggests that there is a systematic error with the automatic procedure.

Manual inspection of the individual results reveals several possible explanations for these results (see Figs. 5 and 6). First, not all the points of interest are identified correctly, especially for hands that are at an angle in relation to the scanning direction. Moreover, there seems to be frequent discrepancies between the actual finger crease at the root of the finger and the crease estimate acquired using the midpoint or normal (see Fig. 6 top left). The quality of the scanned images also affects the results. In particular if there are visual artefacts caused by poor scanners (see Fig. 6 top right), skin coloured background (see Fig. 6 bottom left) and incorrect skin detection if the hand is not fully pressed against the scanner glass (see Fig. 6 bottom right).

These results suggest that future efforts should focus on further improving the robustness of the skin detection, making the point of interest detection invariant to the direction of the hands and consider each hand individually. Moreover, it may be necessary to optically search for the actual finger creases at the root of the fingers in order to achieve a sufficient accuracy, as estimations based on the hand outline appears to be unreliable.

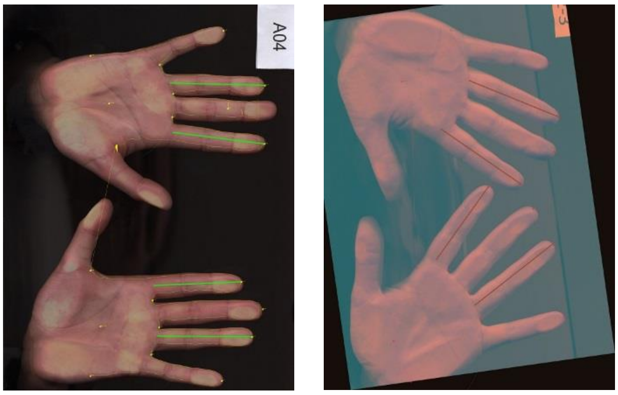

Fig. 5. Successful measurements uncorrected (left), rotation correction (right).

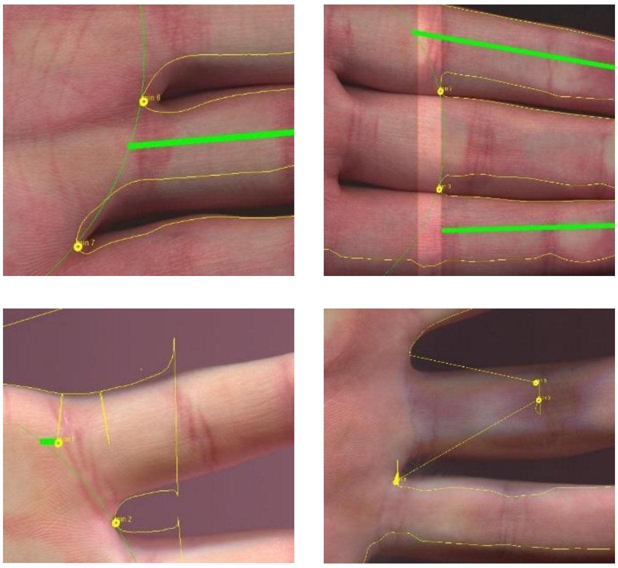

Fig. 6. Unsuccessful measurements. Inaccuracy of crease point based on outline (top left), scanner noise (top right), skin-background similarity (bottom left) and finger above the glass (bottom right). The yellow curves shows the binarization boundaries.

4. Conclusions

Automatic detection of digit ratios has potential for several smart city applications. An automatic procedure for measuring digit ratios was therefore presented. The method was tested on a set of 284 images and the results compared with measurements acquired manually. Although the results are encouraging, the accuracy and robustness is still not yet sufficient for professional applications, confirming that the automatic detection of digit ratios is a harder problem than it seems. Future work will therefore focus on improving the algorithm for separation of hands from the background, handle each hand separately and develop an angle invariant hand outline analyser. Most importantly, an improved automatic procedure needs to perform an optical local search for the actual finger crease point where the finger joins the palm of the hand. The automatic detection of finger creases is a challenging problem due to the large diversity in the hand colour, texture and shape.

References

[1] Putza DA, Gaulinb, SJC, Sporterc, RJ et al (2004) Sex hormones and finger length What does 2D:4D indicate? Evol Hum Behav 25: 182–199.

[2] Voracek M, Pietschnig J, Nader IW et al (2011) Digit ratio (2D:4D) and sex-role orientation: Further evidence and meta-analysis. Pers Indiv Differ 51: 417–422.

[3] McIntyre MH, Barrett ES, McDermott R et al (2007) Finger length ratio (2D:4D) and sex differences in aggression during a simulated war game. Pers Indiv Differ 42: 755–764.

[4] Manning JT, Scutt D, Wilson J et al (1998) The ratio of 2nd to 4th digit length: a predictor of sperm numbers and concentrations of testosterone, luteinizing hormone and oestrogen. Hum Reprod 13: 3000-3004.

[5] Sandnes FE (2014) Measuring 2D: 4D finger length ratios with Smartphone Cameras. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC), IEEE, pp. 1697-1701.

[6] Sandnes FE (2015) An Automatic Two-hand 2D:4D Finger-ratio Measurement Algorithm for Flatbed Scanned Images. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC), IEEE Computer Society Press, pp. 1203-1208.

[7] Sandnes FE (2015) A Two-Stage Binarizing Algorithm for Automatic 2D:4D Finger Ratio Measurement of Hands with Non-Separated Fingers. In: Proceedings of 11th International Conference on Innovations in Information Technology (IIT'15), IEEE, pp. 178-183.

[8] Koch R, Haßlmeyer E, Tantinger D et al (2015) Development and implementation of algorithms for automatic and robust measurement of the 2D: 4D digit ratio using image data. Curr Dir Biomed Eng 1: 220-223.

[9] Fukumoto M, Suenaga Y, Mase K (1994) Finger-Pointer: Pointing interface by image processing. Comput Graph 18: 633-642.

[10] Sauvola J, Pietikäinen M (2000) Adaptive document image binarization. Pattern recogn 33: 225-236.

[11] Vezhnevets V, Sazonov V, Andreeva A (2003) A survey on pixel-based skin color detection techniques. In: Proceedings of Graphicon 3: pp. 85-92.

[12] Kakumanu P, Makrogiannis S, Bourbakis N (2007) A survey of skin-color modeling and detection methods. Pattern recogn 40:1106-1122.

[13] Pavlovic VI, Sharma R, Huang TS (1997) Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. IEEE T Pattern Anal 19: 677-695.

[14] Freeman WT, Roth M (1994) Orientation Histograms for Hand Gesture Recognition. Technical report. Mitsubishi Electric Research Laboratories, Cambridge Research Center, TR-94-03a.

[15] Neyse L, Brañas-Garza P (2014) Digit ratio measurement guide. No. 1914. Kiel Working Paper.

[16] Coetzee L, Botha EC (1993) Fingerprint recognition in low quality images. Pattern Recogn 26:1441–1460.

[17] Arthur D, Vassilvitskii S (2007) k-means++: the advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '07). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 1027-1035.