10.24425/acs.2020.132587

Archives of Control Sciences
Volume 30(LXVI), 2020
No. 1, *pages 101–122*

Tracking of dynamic gesture fingertips position in video sequence

TOMASZ GRZEJSZCZAK, REINHARD MOLLE and ROBERT ROTH

The field of research of this paper combines Human Computer Interface, gesture recognition and fingertips tracking. Most gesture recognition algorithms processing color images are unable to locate folded fingers hidden inside hand contour. With use of hand landmarks detection and localization algorithm, processing directional images, the fingertips are tracked whether they are risen or folded inside the hand contour. The capabilities of the method, repeatibility and accuracy, are tested with use of 3 gestures that are recorded on the USB camera. Fingertips are tracked in gestures presenting a linear movement of an open hand, finger folding into fist and clenched fist movement. In conclusion, a discussion of accuracy in application to HCI is presented.

Key words: hand landmarks, tracking, fingertip detection, hand gesture, gesture recognition, HMI, HCI

1. Introduction

Gestures are commonly used as a nonverbal way of communication. In recent years, developing Human Computer Interfaces (HCI), many researchers have proposed numerous solutions to make gesture-based intuitive control possible. Some of them, such as finger gestures on touch screens of smartphones, or body gesture-based game controller, are well accessible and commonly used in everyday life. However, most often, a dedicated equipment is necessary. In this

T. Grzejszczak (Corresponding author), E-mail: tomasz.grzejszczak@polsl.pl, is with Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, Akademicka 16, 44-100 Gliwice, Poland.

R. Molle and R. Roth are with Chair of Automation, Computer Science University of Wuppertal, Germany.

The research was supported by Polish National Science Center under grant 02/010/PBU17/0090 (PBU/29/RAu1/2017/505). The calculations were performed with the use of IT infrastructure of GeCONiI.

Received 25.10.2019. Revised 24.01.2020.

paper, the approach of gesture recognition based on records captured by widely accessible video cameras is presented.

In the terms of gesture recognition method, the most crucial division is between stationary and dynamic gestures. Stationary gestures are presented once, while the dynamic gestures present the movement described in time. Thus, stationary gestures are usually presented in images, and the dynamic gestures – in video sequences. This brings another crucial distinguishing feature. Stationary gestures are usually presented distinctly with clearly visible finger position. Fingers in dynamic gestures tend to overlap and cover each other during movement. Thus, it is important to develop and test a method for accurate folded finger detection and tracking.

1.1. Overview of vision-based gesture recognition

Communication and human-human interaction can be represented in a verbal and non-verbal way. In non-verbal communication, a gesture is an intentional or unintentional action of the entire human body or its parts, aimed at communicating a certain message. The messages are conveyed with static poses or dynamic movements of body parts (e.x. hand, arm, hip) or facial expressions. Gesture recognition, defined as interaction in HCI, is an important topic in computer vision and has been intensively studied over the years.

The process of hand gesture recognition can be divided into three main stages [4, 36], namely: (i) segmentation of a hand region from the background, (ii) feature extraction and (iii) gesture recognition. Completing the whole process is necessary for sign language recognition [1] or in the HCI systems [39], but, as a number of challenging image processing and pattern recognition tasks are involved, many works are focused on improving particular processing steps, assuming some simplifying conditions for the others.

There are numerous approaches in the stage of segmentation of a hand region from the background. Hand region can be distinguished from background by skin detection and segmentation method [13, 14, 16]. Among many, those methods are based on skin color modeling [18], supported with spatial distribution of skin pixels [41], as well as analysis of the texture [12] and by adapting skin model [13, 42] to a presented scene, which increases the precision of segmenting skin regions. Those approaches heavily depend on the background and lighting conditions, thus in many works on gesture recognition a controlled background is assumed [20] or the hand region extraction is simplified using some markers [26, 35]. In recent years, infrared cameras and sensors found great application in segmentation process in gesture recognition. The applications vary from infrared thermal imaging used to segment a region of the human body temperature [19, 33] to many other depth detecting hardware solutions, such as Time of Flight camera, Kinect or Leap Motion [11, 21, 28].

The next step after detecting the region of interest by subtracting the background is to gather data necessary for proper gesture classification. There are mainly two approaches [3, 30]: (i) appearance-based and (ii) model-based. The latter focuses on fitting a predefined 3D hand model to an image subject to the analysis. Appearance-based approaches uses a number of computer vision-based techniques in order to extract the features from an image. The most common solutions applied here can be categorized into distance transform [23, 27, 40], template matching [29, 34, 40] and contour analysis [5, 9, 27, 32]. Depending on the applied method, different properties are used to locate the hand features: calculating the contour points distance to the palm center is used to measure the dissimilarities between hand shapes [32], silhouette analysis is aimed at finding concave regions from the contour [5] or landmarks are found by relations between the contour pixels [9].

The final step is to classify the extracted appearance features, compare, and recognize or identify the presented gesture. The most common classification procedures are clustering, support vector machines [15], hidden Markov models and neural networks [30]. The classification process depends on a particular application of the developed algorithm. The most common purposes of gesture recognition systems are desktop applications, sign language recognition, games, robotics and augmented reality [30]. However, it is notable that many solutions described in the literature propose novel feature extraction algorithms without focusing on a particular practical application.

Depth sensors have played huge role in HCI and gesture recognition research. In 2010, Microsoft released Kinect, that was previously known as ZCam [10], and gave the programming libraries for human pose detection. Unfortunately, precise hand landmarks recognition was not supported, thus researchers used the depth information to threshold hand region and used other vision-based approaches, such as contour analysis or distance transform [22, 31].

Recently developed Leap Motion controller is dedicated for hand pose estimation [25, 38] and is capable of accurate evaluation of hand landmarks position in time and space. Therefore, some recent research does not focus on detection and location of landmarks, but on the last stage, which is gesture recognition for different applications, such as recognition of numbers drawn in the air [37].

The latest research in hand landmark localization are inspired by deep learning and convolutional neural networks. The idea is to omit the image processing and train a large neural network with hundreds of marked images. In case of landmarks fitting, the first attempt was to estimate a human pose [2]. Recently this approach is tested to work with hand pose estimation using MediaPipe [24]. The concept articles about On-Device, Real-Time Hand Tracking with MediaPipe can be found online, however no research articles were published on this topic so far, thus it is difficult to compare it with the, described in this paper, image-processing approach. Moreover, both approaches are using different algorithms

needing different resources, so they can be more or less applicable in different projects. In this case it is not possible to state that one approach is better then another.

1.2. Contribution

The aim of this research is to follow the capabilities of depth-sensor based solutions, but with the use of standard images from digital camera that are easily accessible and widely used in many computers or smartphones. Therefore, among all different vision approaches, the research presented in this paper is performed with the use of vision-based system processing consecutive 2D images stored as a video sequence. First of all, each frame from a video sequence is segmented to obtain a hand region mask. Next, each frame is subject to gradient direction visualization, distance transform, template matching and a set of heuristic rules to find a set of hand landmarks. For each video frame, the set of detected landmarks is added to a time set containing the whole path of landmarks movement.

The contribution consists in application of heuristic rules on directional image and distance transform to track detected hand landmarks in video sequence. The techniques were developed and compared with state of the art in previous research of hand landmarks detection and localization in stationary color images [7]. In this paper, the input is a video sequence, so the method is amended to constantly track hand landmarks. In this research, the method based on the directional image has been applied, as the alternative techniques focus exclusively on the contour and they fail to detect the landmarks of folded fingers. The results of an experimental study clearly demonstrates the advantages and possibilities of continuous hand landmarks tracking, both in case of clearly risen fingers and in case of folded fingers.

2. Hand landmarks tracking

This section present the overview of hand landmarks detection method that mainly benefits from the analysis of the directional image, which makes it possible to determine the locations of the landmarks positioned inside the skin presence masks. The overview of the method is presented in Fig. 1. Each step of the algorithm is described as a subsection.

2.1. Input

Dynamic gestures are presented in the form of a video sequence. Each frame is grabbed and processed individually, creating a set of input images (Fig. 2a). It is possible that information from previous or next frames can be used to narrow the search region, however this is an optimization process that can be applied

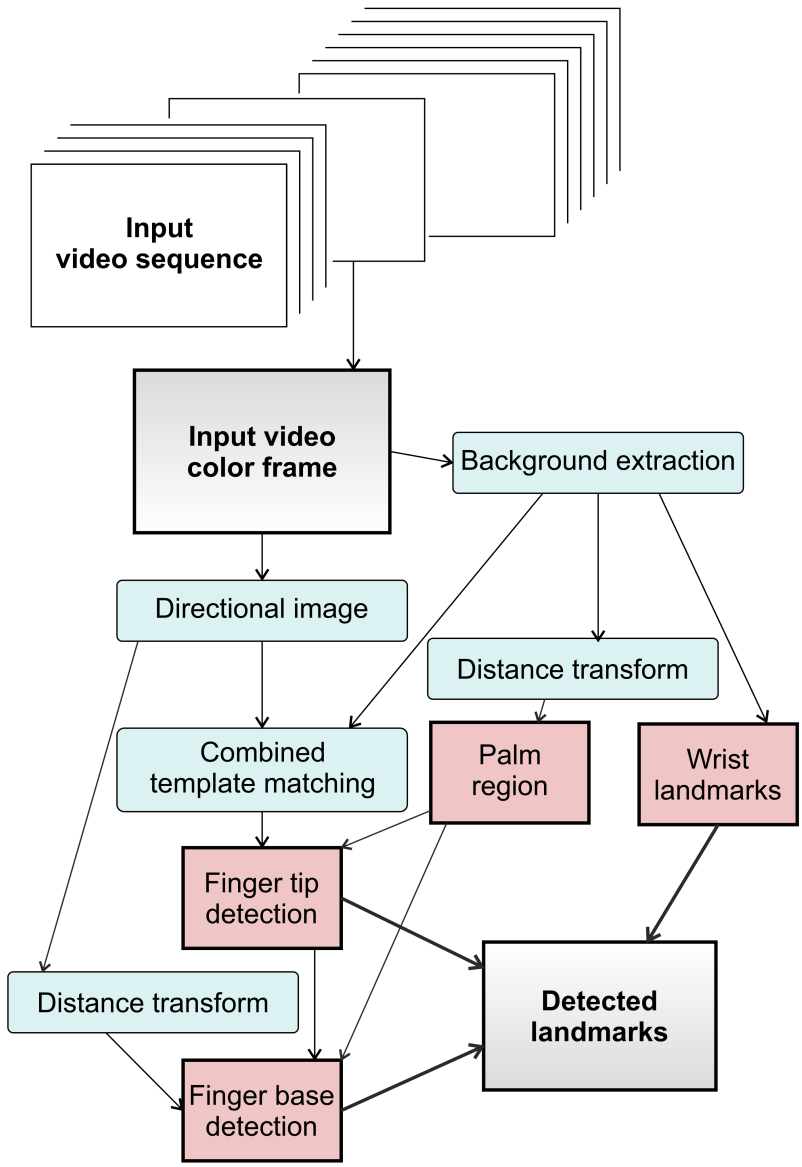

Figure 1: Hand landmarks detection and localization flowchart

once this approach is fully tested. Thus, a dynamic gesture recording is in fact a set of input images of quantity equal to recording time multiplied by recording frame rate.

The input for the image processing algorithm is the stationary, 3 channel RGB image frame from the video. The video resolution is 640×480. Some methods

of the algorithm uses the normalization, that changes the size of the input image to maximum height or width of 300 pixels. The normalization is used in methods that has insignificant influence on the final accuracy of detected landmark (e.g. wrist localization or finding hand orientation). During tests the video resolution, and the normalization parameter $\max(n, m) = 300$ provided the best compromise between processing time and accuracy of landmark localization. The input is a color image stored in 3 dimensional matrix in RGB color space, however, for some image processing methods, the image is converted to gray scale, or HSV color space.

2.2. Image processing methods

Among many image processing methods applied in this research, the most relevant are background extraction, directional image formation, template matching and distance transformation. These methods are commonly known, well described and often found in literature [5, 9, 23, 27, 27, 29, 32, 34, 40, 40].

Directional image [17] is a set of oriented segments. Basing on the input image, the average tangent directions for every $3x3$ group of gradients are approximated. Next, the image is divided into $5x5$ chunks, and for each chunk, the average tangent vector is determined and characterized by angle and variance. Directional image is a visualization of short line segment inclined by tangent angle in each chunk (Fig. 2b). If the variance is higher then a certain value, the line is not drawn (Fig. 2c).

Another method, applied as an initial processing step, is segmentation. In this case, the hand region should be segmented from the background creating a hand skin presence map (Fig. 2d). There are numerous ways of automatic background extraction discussed in section 1.1 overview of vision-based gesture recognition. The described method can work with skin detection algorithm, however at this point, to avoid the error propagation, all recordings were taken on a controlled, stationary background. The skin presence mask, represented as binary image D, is obtained with simple threshold based on saturation range

$$D(x, y) = \begin{cases} 0 & \text{for } S(x, y) < t, \\ 1 & \text{otherwise,} \end{cases} \tag{1}$$

where destination pixel in D image is set depending on value of pixel in source image S and threshold value t is set experimentally to ensure full skin presence mask with smooth contour. The changes of parameter result in images presented in Fig. 3c, d. As the threshold is based on saturation, to ensure the high saturation distinguishability, the background is white, well illuminated with several light sources, that eliminates shadows. The example in Fig. 3 shows that the value of t for this recording can safely be set to any value in range $(56, 96)$ without major influence on the result.

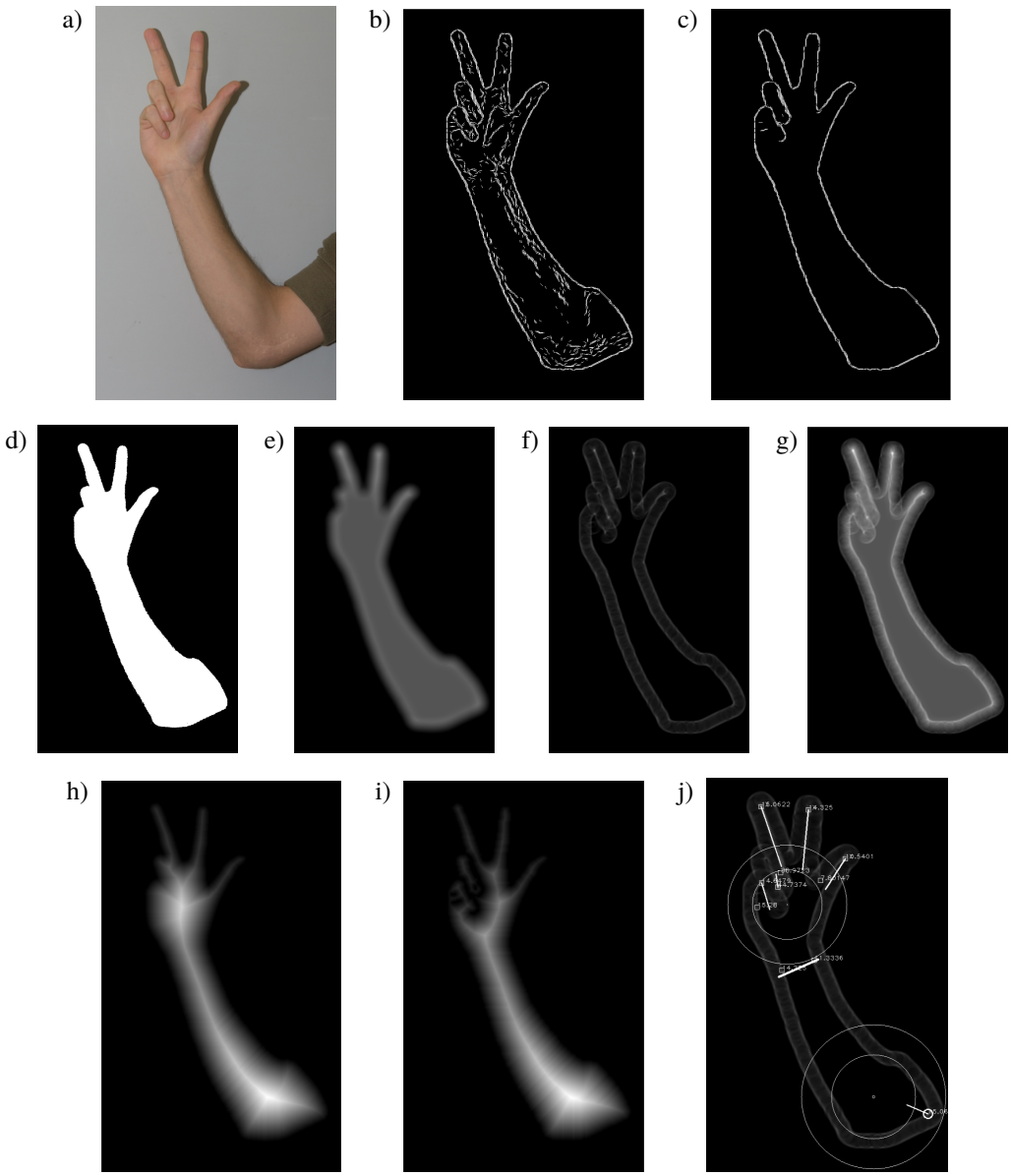

Figure 2: Steps of hand landmarks detection and location algorithm presented with the results of image processing methods

As the fingertips are round, the hand mask image (Fig. 2d) is subject to cross correlation template matching of a circle template image. Local maxima in the output (Fig. 2e) refer to risen finger tips. On the other hand, matching a ring template to directional image (Fig. 2c) gives the localization of folded fingers

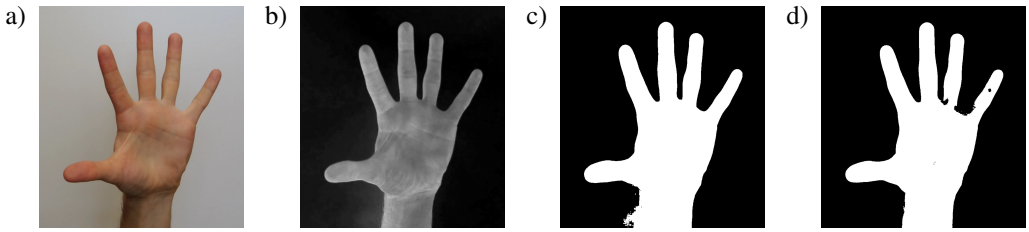

Figure 3: Influence of threshold t parameter on mask image D. a) Part of source image S, b) saturation channel and the application of formula (1) with c) $t = 56$ and d) $t = 96$

and other concave regions, such as space between fingers (Fig. 2f). Adding both template matching matrices results, gives the information of both folded and risen fingers (Fig. 2g), that are located in local maxima. The size of the templates (circle diameters) are dynamically set proportionally to detected hand size that is calculated from distance transform.

Values of distance transformation are the values of distance of the pixel to the nearest contour pixel. Local maxima refer to the center of the circle inscribed in the contour. This property is used to locate palm region and track finger base. Distance transform is applied on mask (Fig. 2d, results in Fig. 2h) and on directional image (Fig. 2c, results in Fig. 2i).

2.3. Wrist landmarks

A hand can be located on the image in any direction with any sleeve length that influences the skin presence mask. The first step is to determine the hand orientation and divide the skin presence mask into hand region and forearm region.

The idea that wrist region can be located on a local minimum of hand profile was tested and proven [8]. The algorithm consists of the following operations:

1. Prepare the input data and the mask contour.
2. Determine the angle of the longest chord of the contour.
3. Rotate the image by the chord's slope and calculate the profile.
4. Find the local minimum of the profile.
5. Compute the final wrist point on the original image.

There are two expected outcomes: determination of hand orientation and emergence of hand region from the skin presence mask.

Hand orientation is determined based on the longest chord inside the skin presence mask. The longest chord is supposed to be a longest line segment with end points located on mask contour with each point contained in the skin presence mask. The angle of slope of this line segment reflects the hand orientation angle.

Wrist line, that divides the skin presence mask into hand region and forearm region, is perpendicular to the longest chord and is located in local minimum of lengths of perpendicular chords. In order to locate the wrist line, the image is rotated by the longest chord angle, so all hand presence pixels can summed in each row. Next, treating the sum of pixels in each row as a profile function, local extrema are found and the local minimum is assumed to be a wrist line. This locally shortest chord end points are treated as two wrist points. The detected wrist line can be observed in Fig. 2j.

Once the skin presence mask is divided into two regions, it is still unclear which one is the hand region, therefore both regions are processed further.

There are some special cases, e.g. short sleeve. In this case local minimum is a global minimum on the end of profile and the whole skin presence mask is a hand region. However research has shown that those cases are irrelevant and brings small influence on the final hand landmarks detection procedure.

2.4. Palm region

In order to calculate hand and fingers size, a palm region has to be located. Palm region is the circle inscribed in the contour, that is a largest circle that can be fitted inside the region. The easiest way to find it is to perform the distance transformation of the region. Distance transformation, for each pixel, sets its value to the distance to the closest background pixel. The maximal value of the transform is the radius of the circle and the location of the maximum is the center of the circle. The algorithm outline is:

1. Prepare the mask image D.
2. Perform the distance transform. Calculate $x_{\max}$, $y_{\max}$ and $r = D(x_{\max}, y_{\max})$, where r is the maximal value of distance transform in point $x_{\max}$, $y_{\max}$.
3. Multiply r by the enhancement coefficient.
4. Draw a filled black circle of radius r in position $x_{\max}$, $y_{\max}$.
5. Perform step 2–3 one more time, to detect second palm region candidate.

Four circles can be observed in Fig. 2j. Each pair with common center is located in the potential palm region. At this point, due to the fact that the hand region is undetermined, potential palm regions are calculated on both sides of the wrist line. The bigger circle with enhanced radius is used to distinguish between folded and risen fingers and between finger tips and finger base.

2.5. Finger tip detection

To locate the finger tips, the template matching is used. In the original idea [34] finger tips are located in local maxima of cross correlation of circle template and hand mask (Fig. 2e). Unfortunately, this method does not detect folded

fingers. In order to find folded fingers that are inside skin presence mask it is necessary to know what is inside the hand contour. The proposed modification uses the ring template matching on the directional image (Fig. 2f). Finger tips are searched among the local maxima in the matrix presenting the sum of two template matching output matrices (Fig. 2g).

At the beginning, there are two empty sets $A = B = \emptyset$. Once the global maximum in Fig. 2g is found, its position $p_{x,y}$ is stored in one of the two sets, depending whether it was detected below or above the wrist line. The detected maximum position

$$\{p_{x,y}\} \cup \begin{cases} A & \text{if } ap_x + bp_y + c > 0, \\ B & \text{otherwise,} \end{cases} \tag{2}$$

where a, b and c are coefficients of wrist line. Pixels values in the vicinity value in the vicinity of this maximum are set to 0. The whole procedure is repeated until no local maximum is present or maximum of 5 points are found on one side of wrist line, which brings $|A| \vee |B| = 5$. The algorithm is:

1. Prepare the connected template matching image (Fig. 2g) TM_{C+R}.
2. Find point p that is $max(TM_{C+R})$.
3. Draw filled black circle of template radius in position $p_{x,y}$.
4. Assign the point to a set, according to formula (2).
5. Repeat 2–4 until $|A| \vee |B| = 5$ or $max(TM_{C+R}) = 0$.

Wrist line divides the skin presence shape into hand region and forearm region. It was tested that more points are detected on the hand region, because the directional image is more detailed there, producing higher template matching values. Therefore, the bigger set – or the one with 5 elements – is the set of the finger tips position.

2.6. Finger base detection

Each finger base is found starting from each finger tip and moving inside the inner contour produced by directional image. Technically, for each finger tip, the closest local maximum of directional image distance transform is found. Then the point is moved along the local maxima until it is close to the palm region circle. The steps consist of:

1. Prepare the distance transform image from ring template matching (fig. 2i) DT_R and the fingertip point $p_{x,y}$.
2. Draw filled black circle of radius $r = 0.9DT_R(x, y)$ in position $p_{x,y}$.

3. Find new point p_{new} that is $max(DT_R)$ on a circle border (with constrain $|pp_{new}| = r$.
4. Assign $p := p_{new}$.
5. Repeat 2–4 until $|pp_{Palm}| < r_{Palm}$, where p_{Palm} is the center of palm region with enhanced radius r_{Palm}.

The output of finger tip and base detection is presented in Fig. 2j. The paired finger tip and finger base are connected by line segment. The numbers on the detected points are part of debugging process and are irrelevant.

2.7. Output

The output of each image is a set of detected hand landmarks and their positions on image. The set contains of two wrist points located on two sides of the detected wrist line segment. Moreover, one chosen fingertips set is included in the output hand landmarks set along with finger base points. Palm region is helpful in finger detection, however it is not a hand landmark, so it is not appended to the output set.

3. Validation procedure

The validation procedure is designed to prove the accuracy and repetability of the presented hand landmarks tracking. In the presented tests, all 5 fingertips are located on the video sequence presenting stationary and dynamic gestures. The tests focus on the accuracy of points detection while performing a linear movement.

3.1. Tests

Before the main dynamic gestures tests, an initial test checking the repeatibility of the method is performed. An open hand static gesture is recorded. The hand does not move, however the frames are changed due to small light changes and frame grabbing properties. The expected output is that all detected points are in the exact same position during the entire test.

The most distinctive feature of this approach is the capability of detecting fingertips of folded fingers. The presented experiments are designed to verify the accuracy of finger tracking in sequences where fingers are folded, overlapping or are detected inside the hand blob. Three tests are proposed in order to fully cover and prove the concept. During each test one dynamic gesture is presented, where the fingertips are moving along a designed line. Even that, some gestures are presented as stationary, they become dynamic by moving them along a path. The detected fingertips positions are gathered and their positions are compared with lines coordinates.

The easiest test for every finger position detecting method is an open hand gesture with spread fingers. To verify the accuracy of finger tracking in this easiest test, the open hand and each of the spread fingers are moved along the horizontal line (Fig. 4a).

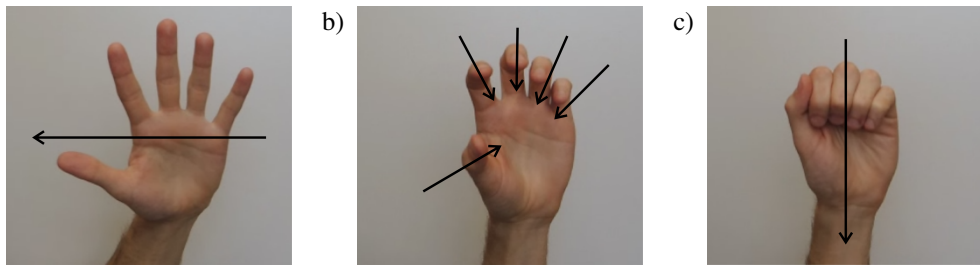

Figure 4: 3 types of dynamic gestures that are tested. Each fingertip is tracked, while a) open hand is moved horizontally, b) hand is clenched, c) fist is moved vertically

Next, to test the finger tracking while folding fingers, the gesture of fist clenching is recorded. Starting from an open hand with spread fingers, the fingertips points are moving along a line in the direction of the palm center. While observing the contour during the movement, it can be noticed that the fingers start to overlap while folding, and the contour becomes more and more circular. At this point many contour-based methods are unable to properly locate fingertips [6, 7] (Fig. 4b).

The last experiment tests the tracking of folded fingers in the most difficult gesture, that is a clenched fist. Here, the stationary fist gesture is moved vertically down, thus all detected fingertips should move along a vertical line (Fig. 4c).

3.2. Measurements

The processing of each video frame produces a hand landmarks set. The full video analysis produces a time sequence of hand landmarks sets as each landmark changes its position in time. During each frame of the recording, the P number of points are detected. The recording consists of N frames. Each detected point is denoted by $p^{[i,j]} = \left(p_x^{[i,j]}, p_y^{[i,j]}\right)$, where $i = 1, 2, ..., P$ and $j = 1, 2, ..., N$.

In many research procedures, user is asked to follow the assumed path, however here, user was asked to perform any linear movement, and the paths were estimated from all detected points. Thus, the detected path is the best line that fits the detected points. As the last step, the estimated line or point was checked by expert, whether it really was the fingertip movement path. Usually, gross errors (visible for example in (Fig. 6a in $y \in (120, 180)$) influenced the estimated line position, and the role of the expert was to remove those points until the detected path covers the real path movement.

In order to verify the repeatibility and accuracy of the proposed method, for each gesture an estimator is chosen. Error d_i is measured as the distance between detected and the expected point position.

In case of stationary gesture, all detected points should not change their position in time, hence the estimator is a point $\hat{p}^{[i]} = (\hat{p}_x^{[i]}, \hat{p}_y^{[i]})$, where

$$\hat{p_x}^{[i]} = \frac{1}{N}\sum_{j=1}^{N} p_x^{[i,j]}, \qquad \hat{p_y}^{[i]} = \frac{1}{N}\sum_{j=1}^{N} p_y^{[i,j]}. \tag{3}$$

In the case of dynamic gesture, the estimator is a line given by equation

$$\overline{a}_i\hat{x} + \overline{b}_i\hat{y} + \overline{c}_i = 0. \tag{4}$$

Coefficient $\overline{a}_i$, $\overline{b}_i$ and $\overline{c}_i$ are calculated based on the average value of all detected points coordinates with the use of line fitting method. The error in points detection is measured as the distance d_i from detected points to the estimator. In case of point estimator

$$d_i = \sqrt{\left(\hat{p_x}^{[i]} - p_x^{[i]}\right)^2 - \left(\hat{p_y}^{[i,j]} - p_y^{[i,j]}\right)^2} \tag{5}$$

and in case of line estimator

$$d_i = \frac{1}{N\sqrt{\overline{a}_i^2 + \overline{b}_i^2}} \sum_{j=1}^{N} \left|\overline{a}_i p_x{}^{[i,j]} + \overline{b}_i p_y{}^{[i,j]} + \overline{c}_i\right|. \tag{6}$$

The results of tests are presented in Fig. 5 for stationary and in Fig. 6 for dynamic gestures.

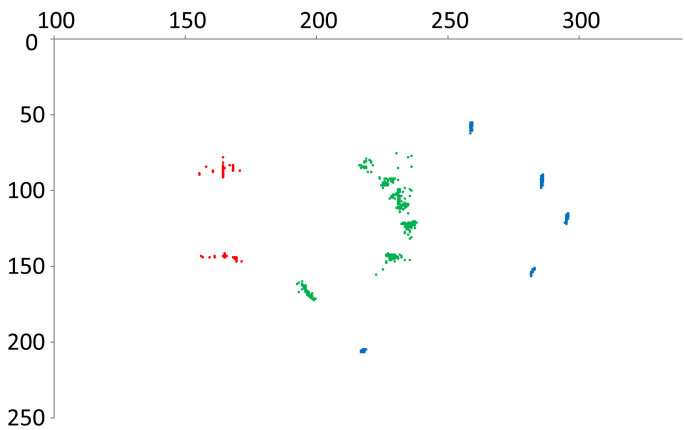

Figure 5: Hand landmarks of a static gesture detected in time. For each frame, the full set contains of 2 wrist points (red), 5 fingertips (blue) and 5 finger bases (green)

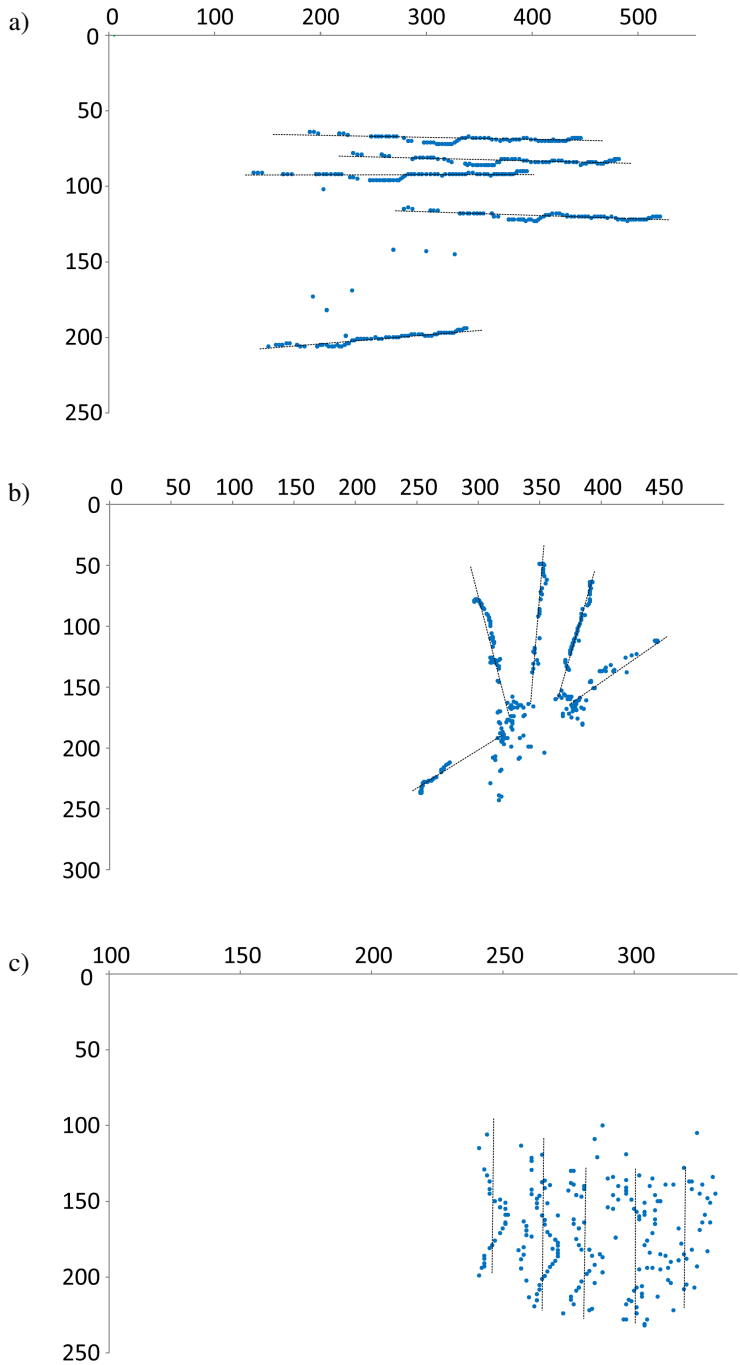

Figure 6: 3 tests of hand landmarks dynamic gesture detection in time. The detected finger tip points are showed for each video frame. The dashed lines correspond to the expected finger movement path

Usually in error measurement, the line fitting methods with the least mean square of line regression is calculated. The error is calculated as Δy which is the distance only in y coordinate. In case of vertical movement, producing vertical line, as shown in Fig. 4c, the Δy cannot be calculated, because coefficient $b = 0$ in (4) and there is no y variable in the formula. Thus, the error in both cases is calculated as the Euclidean distance from point to point or point to line.

3.3. Results

For each point $p^{[i,j]}$ from the time sequence, the distance is measured. For all points of the same type, the average of the distance $\overline{d_i}$ is evaluated along with its standard deviation σ_i. The results for all tests are presented in Tables 1–4. Each value of i was named by the detected finger name.

While reading the results, one need to keep in mind that all presented calculations and results are performed on pixel coordinates. Due to vision system construction and camera limitations, the measurement accuracy is 1 pixel and all measured values of x and y are integer numbers. Fractions are the result of formulas (5), (6) and averaging. In the used recordings, the indicative diameter of finger is 30 pixels, the length of index finger is 120 pixels, palm diameter is 135 pixels and wrist diameter is 85 pixels. Taking into consideration video resolution and hand distance from the camera, each pixel corresponds to approximately 0.7 mm. Therefore, that is only informative, because in the method, the template sizes adjust dynamically, so the errors would be proportionally the same, regardless of hand distance from the camera.

The first test was performed for stationary gesture recorded in 80 frames. The results are presented in Table 1. All of the detected fingertip points were close to each other in each frame with $\overline{d_i} \approx 1$ and $\sigma_i < 1$, that is close to measurement accuracy. Other landmarks are more spread (Fig. 5), however they were not tested.

Table 1: Results of stationary gesture test

Finger	$\overline{d_i}$	σ_i
Thumb	1.57	0.97
Index	1.28	0.45
Middle	1.18	0.38
Ring	0.92	0.54
Pinky	1.02	0.17

Next, the dynamic gestures are tested. Figure 6 a–c clearly refer to values in Tables 2–4. The first dynamic gesture (Table 2) test shows that nearly all points ale located on the line with small mean error and standard deviation. The only high values are in the thumb points, due to gross error.

Table 2: Results of open hand dynamic gesture test

Finger	$\overline{d_i}$	σ_i
Thumb	28.01	17.65
Index	1.23	1.01
Middle	1.35	1.13
Ring	1.34	1.05
Pinky	2.75	1.25

Fist clenching (Table 3) characterizes in small $\overline{d_i}$ errors, however the standard deviation σ_i is high. This is due to the fact, that while the fingertips are beside the hand contour, they are detected accurately. When the fingers are overlapping the contour, the detection error increases. Thus, due to many correctly detected points, the average is small, however due to some large errors, the standard deviation is high.

Table 3: Results of fist clenching gesture

Finger	$\overline{d_i}$	σ_i
Thumb	9.96	12.22
Index	13.16	15.48
Middle	2.36	1.37
Ring	2.37	6.29
Pinky	5.83	6.21

The last test, with all fingers folded in a fist (Table 4), is the most difficult. The error and standard deviation for all points are at the same level, confirming the conclusions from Fig. 6c, that the detected plots are heavily scattered. The mean error of all $\overline{d} = 4.17$, so the mean error range thickness is equal 8.34 while the approximate finger thickness is equal 30 pixels. This means that points are not detected in the exact center of the finger, however the error is acceptable.

Table 4: Results of dynamic fist movement gesture

Finger	$\overline{d_i}$	σ_i
Thumb	5.29	2.96
Index	3.97	3.55
Middle	4.16	2.86
Ring	3.17	3.75
Pinky	4.30	4.47

The overall results of the tests are proving the concept of using the described technique in dynamic gesture finger tracking. The best results are for the gestures with risen fingers, however this approach is not novel and was solved with numerous methods. While folding and folded fingers tracking, the error increases, however it is still acceptable. Gross errors have the biggest influence on the average error, thus the next step in the future work will be application of filtering and other optimization methods.

4. Comparison to state of the art

At the moment, the most accurate hand tracking device is Leap Motion. The test of accuracy and repeatability are compared with the outcome from Leap Motion tests [38], that are presented in first row of Table 5. The results of vision based approach presented in this article are summed and presented in second row of Table 5. The repeatability and standard deviation are calculated with reference to a static point in a stationary gesture (from Table 1) while the average accuracy is calculated with regards to path following with dynamic gesture (from Tables 2–4).

Table 5: Comparison of obtained results with state of the art

Method	Measurand	Repeatability	σ_i	Accuracy error
Leap Motion [38]	reference pen 10.0 (mm)	0.1276 (1.27%)	0.0268 (0.26%)	1.2 (12%)
Vision Based	finger 30(px)	1.194 (3.98%)	0.502 (1.67%)	5.95 (19.8%)

One important remark is that Leap Motion controller was tested with use of different width reference pens. The measurements were taken in millimeters, while in this article, the measurements are presented in pixels. Thus all presented results are shown in percentage of a measurand. Among presented results, a pen with 10 mm diameter is chosen to comparison, because it is most similar to finger width. From the video recordings, the average finger width is calculated to be equal to 30 pixels (px).

The presented comparison shows that the described vision based approach is nearly as good as Leap Motion controller in case of dynamic gestures. In case of repeatability, calculated from a static gesture, the error is much higher with comparison to Leap Motion. The probable influence is that Leap Motion was tested with a manipulator, that was able to withstand micro movement. Moreover, in the presented tests, the static error was on average equal to 1 pixel, which is the measurement scale, thus no better results could be obtained in this environment.

5. Conclusions

The presented research and test results prove the concept of fingertip tracking basing on the hand landmarks detection and localization algorithm. In the presented dynamic gestures, fingertips are located with sufficient accuracy. Risen fingertips are detected with higher accuracy than the folded fingers, however in both cases the mean error is smaller than the finger diameter. Gross errors that are results of wrong processing of video frame have the highest influence on the output points position. In future work the optimization and filtering methods would be developed to reduce this type of error.

References

[1] Washef Ahmed, Kunal Chanda, and Soma Mitra: Vision based hand gesture recognition using dynamic time warping for indian sign language, In *2016 International Conference on Information Science (ICIS)*, pages 120–125. IEEE, 2016.

[2] Z. Cao, G. Hidalgo, T. Simon, S.E. Wei, and Y. Sheikh: Openpose: realtime multi-person 2d pose estimation using part affinity fields, *arXiv preprint arXiv:1812.08008*, 2018.

[3] Ankit Chaudhary, J.L. Raheja, Karen Das, and Sonia Raheja: Intelligent approaches to interact with machines using hand gesture recognition in natural way: A survey, *CoRR*, abs/1303.2292, 2013.

[4] Ming Jin Cheok, Zaid Omar, and Mohamed Hisham Jaward: A review of hand gesture and sign language recognition techniques, *International Journal of Machine Learning and Cybernetics*, **10**(1) (2019), 131–153.

[5] Zhiquan Feng, Bo Yang, Yuehui Chen, Yanwei Zheng, Tao Xu, Yi Li, Ting Xu, and Deliang Zhu: Features extraction from hand images based on new detection operators, *Pattern Recognition*, **44**(5) (2011), 1089–1105.

[6] T. Grzejszczak, A. Gałuszka, M. Niezabitowski, and K. Radlak: Comparison of hand feature points detection methods, In Luis M. Camarinha-Matos, Nuno S. Barrento, and Ricardo Mendonça, editors, *Technological Innovation for Collective Awareness Systems*, pages 167–174, Berlin, Heidelberg, 2014, Springer Berlin Heidelberg.

[7] T. Grzejszczak, M. Kawulok, and A. Galuszka: Hand landmarks detection and localization in color images, *Multimedia Tools and Applications*, **75**(23) (2016), 16363–16387.

[8] T. Grzejszczak, J. Nalepa, and M. Kawulok: Real-time wrist localization in hand silhouettes, In Robert Burduk, Konrad Jackowski, Marek Kurzynski, Michal Wozniak, and Andrzej Zolnierek, editors, *Proc. International Conference on Computer Recognition Systems CORES 2013*, volume 226 of *Advances in Intelligent Systems and Computing*, pages 439–449, Springer International Publishing, 2013.

[9] M. Hagara and J. Pucik: Fingertip detection for virtual keyboard based on camera, In *Radioelektronika (RADIOELEKTRONIKA), 2013 23rd International Conference*, pages 356–360, April 2013.

[10] G.J. Iddan and G. Yahav: Three-dimensional imaging in the studio and elsewhere, In *Three-Dimensional Image Capture and Applications IV*, volume 4298, pages 48–56, International Society for Optics and Photonics, 2001.

[11] Feng Jiang, Shengping Zhang, Shen Wu, Yang Gao, and Debin Zhao: Multi-layered gesture recognition with kinect, In *Gesture Recognition*, pages 387–416, Springer, 2017.

[12] M. Kawulok, J. Kawulok, and J. Nalepa: Spatial-based skin detection using discriminative skin-presence features, *Pattern Recognition Letters*, **41** (2014), 3–13.

[13] M. Kawulok, J.a Kawulok, J. Nalepa, and B. Smolka: Self-adaptive algorithm for segmenting skin regions, *EURASIP Journal on Advances in Signal Processing*, **2014**(170) (2014).

[14] M. Kawulok, J. Kawulok, J. Nalepa, and B. Smolka: Hybrid adaptation for detecting skin in color images, *Intelligent Data Analysis*, **20**(s1) (2016), S121–S139.

[15] M. Kawulok and J. Nalepa: Support vector machines training data selection using a genetic algorithm, In Georgy Gimel'farb, Edwin Hancock, Atsushi Imiya, Arjan Kuijper, Mineichi Kudo, Shinichiro Omachi, Terry Windeatt, and Keiji Yamada, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 7626 of *Lecture Notes in Computer Science*, pages 557–565, Springer Berlin Heidelberg, 2012.

[16] M. Kawulok, J. Nalepa, and J. Kawulok: Skin detection and segmentation in color images, In M. Emre Celebi and Bogdan Smolka, editors, *Advances in Low-Level Color Image Processing*, volume 11 of *Lecture Notes in Computational Vision and Biomechanics*, pages 329–366, Springer Netherlands, 2014.

[17] M. Kawulok and J. Szymanek: Precise multi-level face detector for advanced analysis of facial images, *IET image processing*, **6**(2) (2012), 95–103.

[18] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat, and J. Jatakia: Human skin detection using rgb, hsv and ycbcr color models, *arXiv preprint arXiv:1708.02694*, 2017.

[19] Ngoc Le Ba, Sechang Oh, Dennis Sylvester, and Tony Tae-Hyoung Kim: A 256 pixel, 21.6 μw infrared gesture recognition processor for smart devices, *Microelectronics Journal*, **86** (2019), 49–56.

[20] Bei Li, Ying Sun, Gongfa Li, Jianyi Kong, Guozhang Jiang, Du Jiang, Bo Tao, Shuang Xu, and Honghai Liu: Gesture recognition based on modified adaptive orthogonal matching pursuit algorithm, *Cluster Computing*, **22**(1) (2019), 503–512, Jan.

[21] Wen-Jeng Li, Chia-Yeh Hsieh, Li-Fong Lin, and Woei-Chyn Chu: Hand gesture recognition for post-stroke rehabilitation using leap motion. In *2017 International Conference on Applied System Innovation (ICASI)*, pages 386–388. IEEE, 2017.

[22] Yi Li: Hand gesture recognition using kinect, In *2012 IEEE International Conference on Computer Science and Automation Engineering*, pages 196–199, June 2012.

[23] Hui Liang, Junsong Yuan, and D. Thalmann: 3D fingertip and palm tracking in depth image sequences, In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 785–788, New York, NY, USA, 2012. ACM.

[24] C. Lugaresi, Jiuqiang Tang, Hadon Nash, C. McClanahan, E. Uboweja, M. Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, M. Georg, and M. Grundmann: Mediapipe: A framework for building perception pipelines, *arXiv preprint arXiv:1906.08172*, 2019.

[25] G. Marin, F. Dominio, and P. Zanuttigh: Hand gesture recognition with leap motion and kinect devices, In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 1565–1569. IEEE, 2014.

[26] J. Mazumder, L.N. Nahar, and M.U. Atique: Finger gesture detection and application using hue saturation value, *International Journal of Image, Graphics & Signal Processing*, **10**(8) (2018).

[27] A. Memo and P. Zanuttigh: Head-mounted gesture controlled interface for human-computer interaction, *Multimedia Tools and Applications*, **77**(1) (2018) 27–53.

[28] J. Molina, J.A. Pajuelo, and J.M. Martínez: Real-time motion-based hand gestures recognition from time-of-flight video, *Journal of Signal Processing Systems*, **86**(1) (2017) 17–25.

[29] M. Monisha and P.S. Mohan: A real-time embedded system for human action recognition using template matching, In *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*, pages 1–5. IEEE, 2017.

[30] S.S. Rautaray and A. Agrawal: Vision based hand gesture recognition for human computer interaction: a survey, *Artificial Intelligence Review*, pages 1–54, 2012.

[31] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang: Robust hand gesture recognition with kinect sensor, In *Proceedings of the 19th ACM international conference on Multimedia*, pages 759–760, ACM, 2011.

[32] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang: Robust part-based hand gesture recognition using kinect sensor, *Multimedia, IEEE Transactions on*, **15**(5) (2013), 1110–1120, Aug.

[33] A.S. Shirazi, Y. Abdelrahman, N. Henze, S. Schneegass, M. Khalilbeigi, and A. Schmidt: Exploiting thermal reflection for interactive systems, In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 3483–3492, New York, NY, USA, 2014. ACM.

[34] Y. Sato, Y. Kobayashi, and H. Koike: Fast tracking of hands and fingertips in infrared images for augmented desk interface, In *Automatic Face and Gesture Recognition, 2000, Proceedings, Fourth IEEE International Conference on*, pages 462–467, 2000.

[35] Kabid Hassan Shibly, Samrat Kumar Dey, Md Aminul Islam, and Shahriar Iftekhar Showrav: Design and development of hand gesture based virtual mouse, In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–5. IEEE, 2019.

[36] M. Sonka, V. Hlavac, and R. Boyle: *Image processing, analysis, and machine vision*, Cengage Learning, 2014.

[37] Qinghui Wang, Ying Wang, Fenglin Liu, and Wei Zeng: Hand gesture recognition of arabic numbers using leap motion via deterministic learning, In *2017 36th Chinese Control Conference (CCC)*, pages 10823–10828, July 2017.

[38] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler: Analysis of the accuracy and robustness of the leap motion controller, *Sensors*, **13**(5) (2013), 6380–6393.

[39] G. Wu and W. Kang: Vision-based fingertip tracking utilizing curvature points clustering and hash model representation, *IEEE Transactions on Multimedia*, **19**(8) (2017), 1730–1741.

[40] K. Yadav, L.P. Saxena, B. Ahmed, and Y.K. Krishnan: Hand gesture recognition using improved skin and wrist detection algorithms for indian sign, *Journal of Network Communications and Emerging Technologies (JNCET)*, **9**(2) (2019), www. jncet. org.

[41] Qingrui Zhang, Mingqiang Yang, Kidiyo Kpalma, Qinghe Zheng, and Xinxin Zhang: Segmentation of hand posture against complex backgrounds based on saliency and skin colour detection, *IAENG International Journal of Computer Science*, **45**(3) (2018), 435–444.

[42] Lian Deng and Shuhua Xu: Adaptation of human skin color in various populations, *Hereditas*, **155**(1) (2018).