Multimed Tools Appl
DOI 10.1007/s11042-015-2934-5

Hand landmarks detection and localization in color images

Tomasz Grzejszczak[1] · **Michal Kawulok**[2] · **Adam Galuszka**[1]

Received: 18 February 2015 / Revised: 17 July 2015 / Accepted: 1 September 2015

Abstract This paper introduces a new method for detecting and localizing hand landmarks in 2D color images. Location of the hand landmarks is an important source of information for recognizing hand gestures, effectively exploited in a number of recent methods which operate from the depth maps. However, this problem has not yet been satisfactorily solved for 2D color images. Here, we propose to analyze the skin-presence masks, as well as the directional image of a hand using the distance transform and template matching. This makes it possible to detect the landmarks located both at the contour and inside the hand masks. Moreover, we performed an extensive experimental study to compare the proposed method with a number of state-of-the-art algorithms. The obtained quantitative and qualitative results clearly indicate that our approach outperforms other methods, which may help improve the existing gesture recognition systems.

Keywords Gesture recognition · Hand pose estimation · Hand landmarks detection · Human-computer interaction · Directional image

1 Introduction

It can be easily observed that the development of new human-computer interaction (HCI) techniques is primarily aimed at increasing the intuitiveness of control. In the last few years,

✉ Tomasz Grzejszczak
tomasz.grzejszczak@polsl.pl

Michal Kawulok
michal.kawulok@polsl.pl

Adam Galuszka
adam.galuszka@polsl.pl

[1] Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland

[2] Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Published online: 22 September 2015

the touch screens have become widely popular, and recently the first touchless, gesture-based interfaces are emerging as commercial solutions. Introduction of the Kinect sensor has greatly improved the capacities of gesture recognition [29–31, 42, 52], however in order to increase the accessibility of touchless interfaces, more effective solutions based on computer vision would help in taking advantage of large popularity and low price inherent to standard 2D cameras. The vision-based systems [6, 8, 14, 17, 25, 32, 37, 45, 47, 49, 51, 54, 56, 57, 59] still do not allow for recognizing the gestures with sufficient accuracy, which brings the need for developing new image analysis algorithms for gesture recognition. One of the major problems here, addressed in this paper, is concerned with detecting and localizing hand landmarks, such as fingertips, joints, or phalanges. This is a crucial step towards estimating a hand pose from digital images, which could decrease the gap in the effectiveness between existing vision-based solutions and those relying on the depth data.

1.1 Overview of vision-based gesture recognition

A gesture is an intentional action of the entire human body or its parts, aimed at communicating a certain message. Static or dynamic features may be relevant here, and the messages are usually conveyed using hand poses, facial expressions, or arm movements. Gesture recognition from digital images and videos is an important topic in computer vision that has been intensively studied over the years. Different methods have been developed for recognizing static [8, 58] or dynamic [34] hand gestures, facial expressions [53] or body actions [40].

Hand gesture recognition requires solving a number of challenging computer vision and pattern recognition tasks, including (i) human skin segmentation [19, 20] to extract hand regions from color images, (ii) hand pose estimation [8], (iii) hand tracking [10], and (iv) hand motion analysis and recognition [28, 41]. Among the methods for estimating a hand pose, there are solutions based on localizing hand landmarks [6, 17, 45, 51, 54], extracting hand shape features [36, 37, 56], or fitting the parameters of a 3D hand model [15, 49, 59]. In the last case, subspace learning (linear [15] or non-linear [59]) is applied so as to improve the searching process in a large database of hand images obtained from the model.

Hand landmarks localization is performed in many gesture recognition systems, but usually it is limited to detecting fingertips of extended digits [6, 14, 17, 25, 45, 54, 57], as well as finding the wrist [11, 32] and palm region [6, 47, 51]. This is certainly helpful for estimating a hand pose, when combined with the analysis of some shape features extracted from the hand silhouette or contour. Importantly, retrieving more information on hand landmarks location may improve the accuracy of the state-of-the-art techniques, especially if it were possible to find the position of the landmarks located inside the hand silhouettes. It has been achieved only for the depth maps [29–31, 52], acquired using the Kinect sensor or time-of-flight (ToF) cameras, and this has substantially improved the accuracy and reliability of gesture-controlled interfaces.

1.2 Contribution

There is a substantial difference in the effectiveness that can be achieved using the solutions which exploit the depth maps, compared with the vision-based systems utilizing only 2D images. It is worth noting that the main benefit of the former consists in the accuracy of detecting the region and landmarks of a hand. Therefore, we addressed this problem in the research reported in this paper. At first, we transform an input image of a hand into the directional image, and subsequently, we perform the distance transform from the rendered

directional image and from the hand contour. Local maxima in the distance map form a set of landmark candidates, which is filtered relying on the data extracted using the template matching. Finally, we label the landmarks using a set of heuristic rules.

Overall, our contribution consists in using the distance tranform performed in the directional image to detect the hand landmarks, which makes it possible to find them regardless of whether they are located at the hand contour or inside the hand silhouette. This is in contrast to the alternative existing techniques, which are focused exclusively on the contour, and they fail to detect the landmarks of the folded digits. Furthermore, we elaborate on the procedure for comparing different algorithms for hand landmark localization, which was presented in our earlier work [12]. The results of an extensive experimental study clearly demonstrate the advantages of the proposed method compared with a number of alternative state-of-the-art approaches. Also, the results identified the different aspects of the analyzed algorithms, which may be helpful in designing a new hybrid method.

1.3 Paper structure

The paper is organized as follows. In Section 2, we present the related literature with particular attention given to the existing methods for detecting hand landmarks. The proposed algorithm is described in Section 4, and the results of experimental validation are reported and discussed in Section 5. The paper is concluded with Section 6.

2 Related literature

The process of hand gesture recognition can be divided into three main stages [16, 48], namely: (i) segmentation of a hand region from the background, (ii) feature extraction, and (iii) gesture or hand pose classification. Regarding the last stage, the recognition may be performed using the features extracted from still images or video sequences, and it may consist in classifying the feature vector to a certain gesture class or estimating the hand pose. Completing the whole process is necessary in the HCI systems [5, 26] or for sign language recognition [61], but as a number of challenging image processing and pattern recognition tasks are involved here, many works are focused on improving particular processing steps, assuming some simplifying conditions for the others. The methods applied at different stages of the gesture recognition process are outlined in Section 2.1, which gives the context for the work reported here. Existing methods used for detecting hand landmarks are reported in Section 2.2.

2.1 Gesture recognition process

There have been a number of methods for skin detection and segmentation proposed [19–21] that are based on skin color modeling [18], supported with the analysis of the texture [22] as well as spatial distribution of skin pixels [44]. Also, adapting the skin model [21, 62] to a presented scene increases the precision of segmenting skin regions. These techniques are utilized in a number of gesture recognition systems [4, 51]. The difficulty of hand region extraction heavily depends on the lighting and background conditions imposed during image acquisition, and in many works on gesture recognition a controlled background is assumed [54] or the hand region extraction is simplified using some markers [2]. This step can also be handled using infrared thermal imaging in order to segment a region

characterized with the human body temperature [45, 46]. Furthermore, a depth sensor or a stereoscopy system may be helpful here, as the skin region is supposed to be positioned close to the camera in the depth map [1, 13, 43, 55].

After segmenting the region of interest from the background, the next step consists in extracting the features that can be used for proper gesture classification. There are mainly two general approaches towards extracting the features [3, 41]: (i) appearance-based and (ii) model-based. The former methods involve analysis of the hand silhouette shape [36, 37, 56] and detecting the landmarks [6, 17, 45, 51, 54], which is the main scope of the work reported here and are given more attention in the subsequent section. The model-based methods [15, 49, 59] are focused on fitting a predefined 3D hand model to an image subject to the analysis—the 3D model parameters form the feature vector which may be further processed.

The gesture classification stage constitutes the last step of the whole process. The extracted appearance features are compared and classified in order to recognize and identify the presented gesture. The most common procedures for classification are based on clustering, support vector machines [23], hidden Markov models and neural networks [41]. 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 [41]. However, it is worth noting that many solutions described in the literature propose novel feature extraction algorithms without focusing on a particular practical application.

2.2 Hand landmarks detection

There have been many approaches proposed to detect and localize the hand landmarks, usually as a feature extraction step in estimating the hand pose. The majority of existing techniques are limited to finding the fingertips of extended digits, which is a relatively easy task that can be successfully accomplished in most cases. When combined with some shape-based features, this may be sufficient to discriminate between a limited number of hand gestures. Seldom has it been attempted to detect other landmarks associated with particular knuckles [45, 51]. Existing approaches may be categorized into those that exploit (i) template matching [17, 25, 39, 45], (ii) distance transform [6, 30] and (iii) contour analysis [9, 14, 42, 54].

Fingertips and knuckles of the folded digits form curvatures in the hand silhouette, which may be extracted by matching a circular template to the hand mask. This observation underpins a number of methods. Sato et al. [45] applied a template with a fixed size of 15×15 pixels to detect the fingertips in hand masks normalized to 80×80 pixels. Local maxima exceeding a certain threshold are considered the fingertip candidates, each of which is verified using simple heuristic rules. From the detected landmarks, the wrist line is estimated, and the hand region is subject to iterative erosion to determine the palm center. These landmarks may be tracked in video sequences, which allowed the authors to create a simple gesture-based interface. This method was later extended with a set of heuristic rules [39] to detect a thumb and digit bases. A very similar technique was utilized by Infantino et al. [17] to detect the fingertips—it was demonstrated that based on their relative locations, it is possible to discriminate between several hand poses, which was used for controlling a robotic hand. Recently, Kerdvibulvech [25] presented an embedded system for tracking the fingertips, which are detected by matching a semicircular template to the detected skin region.

Overall, the template matching methods enable fast fingertip detection of extended digits, which makes it possible to distinguish between simple gestures. However, these methods do not detect the landmarks positioned inside the hand mask, and may produce false positives for folded digits—in such cases, the knuckles are likely to be determined as fingertips.

Another approach towards detecting the landmarks was reported by Dung and Mizukawa [6]. In their system for extracting hand features, the palm center is determined at first throughout a heuristic process employing morphology, connected component labeling and image moments. Also, the distance transform is proceeded from the boundary of the hand mask and local maxima that fulfill certain conditions are grouped using the Hough transform to determine the digit segments. In this way, the fingertips and the digit bases are localized as the extreme points of these segments. This approach makes it possible to estimate the finger direction with high precision and allows for detecting the landmarks of extended fingers, but it is less effective for folded digits. A similar approach was applied by Liang et al. [30] to detect the fingertips from depth images acquired using the Kinect sensor. It is assumed that the hand is formed by the largest skin-color blob, and the palm center is determined from the circle inscribed into the blob. Afterwards, the geodesic distance transform is proceeded in the 3D domain from the palm center to find the fingertips of extended and folded digits. This method was later enhanced by the authors with a 3D model of a hand [31]. The geodesic distance transform has been also utilized by Krejov and Bowden [29] to detect the fingertips in depth images.

Hand landmarks can also be detected from the hand silhouette. Tanibata et al. [54] developed a system for sign language recognition, whose operation requires a number of simplifying conditions, however given the wrist position, the fingertips are detected without any further restrictions. A function of the distance between the wrist location and subsequent points at the hand contour is created, and its local maxima are considered as fingertips candidates. Similarly as in other aforementioned works, this makes it possible to detect the fingertips of extended digits, but it is unlikely to work for the folded ones—such landmarks would not be detected or the knuckles would be mistaken as the fingertips. Ren et al. [42] proposed to exploit polar representation of the contour vertices (with the palm center as the origin) to analyze the depth images. A relatively complex algorithm for detecting hand contour landmarks has been proposed by Feng et al. [9]. The method exploits a multi-scale space to analyze the local curvatures of the contour, which makes it possible to detect the fingertips and digit bases located at the contour. The curvatures are also extracted from the mutual relations between the contour pixels and used to detect the fingertips in the virtual keyboard interface developed by Hagara and Pucik [14].

An interesting method for hand shape analysis to detect the landmarks was proposed by Stergiopoulou and Papamarkos [51]. The self-growing and self-organized neural gas (SGONG) network is established within the hand silhouette—after an iterative optimization procedure, particular neurons are expected to be positioned at the landmark locations. This makes it possible to detect the landmarks lying on the contour as well as those positioned inside the hand region. Another implementation of the neural gas was proposed later by Shi et al. [47].

The existing methods for detecting the hand landmarks are often exploited as a part of a complex system for gesture recognition, and quantitative evaluation is usually focused on assessing the higher level performance rather than the detection outcome. In most cases, the landmark detection is limited to the fingertips of extended digits, which allows for discriminating between a few different hand poses.

3 Theoretical background

In this section, we present two algorithms that are essential to explain the details of our proposed method. In Section 3.1, we describe the operations required to obtain a directional image, and in Section 3.2 we outline our earlier method for localizing the wrist line.

3.1 Directional image

The directional image is a set of oriented segments, and it was used in our earlier work for face detection [24].

In order to obtain the directional image, averaged tangent directions for every 3×3 group of gradient vectors are approximated. The gradient vector $\mathbf{g}_{m,n}$ is computed for every pixel $I_{m,n}$ using Sobel operator:

$$\mathbf{g}_{m,n} = \begin{bmatrix} I_{m+1,n+1} + I_{m+1,n-1} - I_{m-1,n+1} - I_{m-1,n-1} + 2I_{m+1,n} - 2I_{m-1,n} \\ I_{m+1,n-1} + I_{m-1,n-1} - I_{m-1,n+1} - I_{m+1,n+1} + 2I_{m,n-1} - 2I_{m,n+1} \end{bmatrix}. \quad (1)$$

After that, for every third pixel in every third row (i.e., for every ninth pixel in the image) a single tangent vector ($\mathbf{u}_{m,n}$) is determined based on 25 gradients in 5×5 neighborhood. The procedure is repeated for every third pixel in every third row.

The tangent vector is a unit vector which minimizes the given error function:

$$\delta(\mathbf{u}_{m,n}) = \sum_{i=m-2}^{m+2} \sum_{j=n-2}^{n+2} (\mathbf{g}_{i,j} \cdot \mathbf{u}_{m,n})^2. \quad (2)$$

The function is minimized analytically to obtain the tangent angle. If the variance of the gradient directions is high within the group of pixels, then the normalized error of averaging defined as:

$$\rho_{m,n} = \delta_{min}(\mathbf{u}_{m,n}) \Big/ \sum_{i=m-2}^{m+2} \sum_{j=n-2}^{n+2} (\mathbf{g}_{i,j})^2 \quad (3)$$

is high as well, and the detected tangent direction is not reliable. The directions, for which the error is greater than a certain threshold value ($\mathcal{T}_\rho = 0.25$), are considered unreliable and are rejected. Every detected tangent direction is characterized by its location, angle, and magnitude defined as a sum of squared contributing gradients. The magnitude quantifies how strong the direction is, and only the directions with high magnitudes (over a certain threshold $\mathcal{T}_M$) are considered further.

The process of creating the directional image is illustrated in Fig. 1. For each pixel of the input image (a), averaged tangent directions are calculated for every pixel in the 3×3 neighborhood—green arrows in (b) and (c). The directional lines are rendered in the final image (d), if in the 5×5 neighborhood (c) all tangent directions are characterized with small variance. This condition is met for the red and blue directions in (c) and (d), while for the yellow question mark in (c), the tangent directions in the 5 × 5 neighborhood are too diverse to determine the direction.

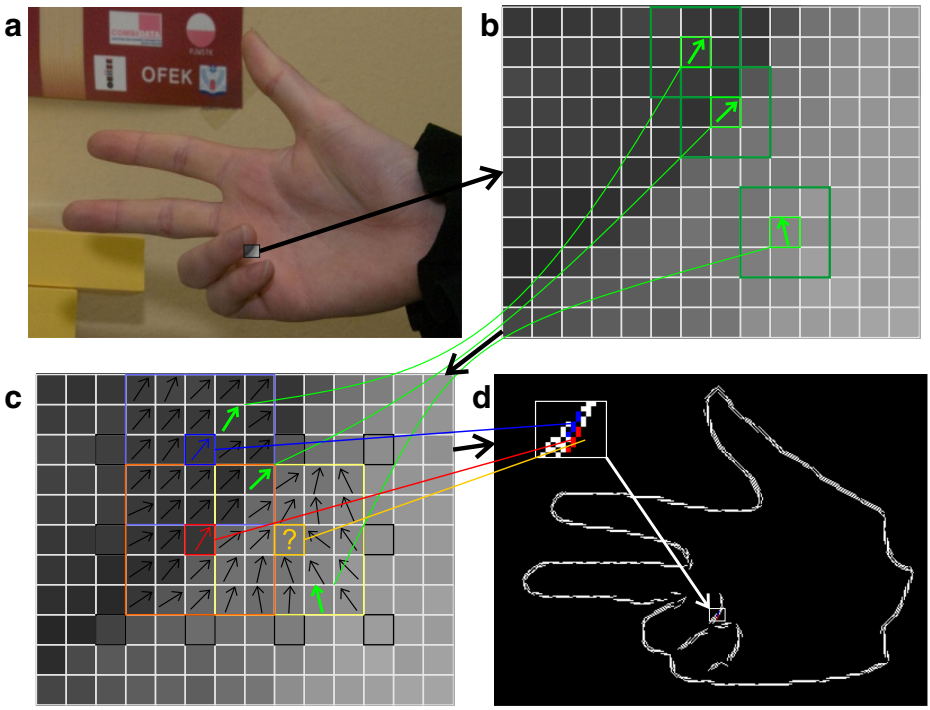

Fig. 1 Process of creating a directional image

3.2 Wrist line localization

Wrist detection is an important step in recognizing hand gestures, as it ensures proper hand region estimation from a skin-presence mask. The procedure is based on the observation that the wrist forms a local minimum in the width profile of the hand silhouette. The aim of this step is to calculate the line that divides the mask of a hand into two parts: hand region and forearm region. The algorithm consists of the following operations [11]:

1. Prepare the input data: calculate the contour.
2. Determine the longest chord of the contour.
3. Rotate the image by an angle of the chord's slope.
4. Find the local extrema, assuming local minimum to be the wrist point.
5. Compute the final wrist point in the original image.

The algorithm analyzes the mask shape and its margin points, thus a contour of a hand blob is calculated as the first step. Next, the longest chord is determined and the silhouette is rotated, so the segment is positioned horizontally. After rotation, the width at every position of the rotated chord is obtained. A sum of skin pixels in each column of the rotated image forms an image profile, which is analyzed to determine the wrist position at a specified local minimum coordinates. The margin points are projected back to the original image afterwards.

It is also important to note that at this step, only the wrist points are located. A line passing through those points divides the mask into two regions, and it is unknown which one is a hand region and which is a forearm region.

4 Hand landmarks detection and localization

In this section, we present the details of our proposed method for localizing landmarks of a hand. Compared with the existing methods, our proposed approach benefits from the analysis of the directional image, which makes it possible to determine locations of the landmarks positioned inside the skin-presence masks (SPMs). Taking into account the categorization presented earlier in Section 2.2, our method falls into two categories, as it benefits from the template matching and the distance transform.

4.1 Hand landmarks

First of all, we have defined a set of hand landmark points, which should be detected to define the hand pose. They have been selected based on the anatomy of a human hand, presented in Fig. 2. A hand consists of 27 bones, among which 14 are the phalanges that form the hand digits (i.e., fingers and a thumb), 5 form the palm, and the remaining 8 bones are positioned within the wrist. In order to identify the hand pose, it is necessary to determine the positions of the digits, and therefore we intent to localize the fingertips (5 orange dots in the figure) and the digit joints (14 green dots). In addition, the set of landmarks includes 4 points (blue dots) that are positioned between the digits. Finally, the wrist line is defined using two extreme points positioned at the hand contour (red dots).

Overall, the generally defined task is to localize 25 landmarks within the hand area. Here, our aim is to localize the wrist line and the digits (i.e., a fingertip and the first visible knuckle), which limits our set to 10 landmarks of the digits and two of the wrist. It is worth noting that usually some landmarks are occluded in the image by the hand itself, and they should not be detected in such cases.

4.2 Algorithm outline

A general flowchart of the algorithm is presented in Fig. 3, and its subsequent steps with some exemplary results are visualized in Fig. 4. An input color image (Fig. 4a) along with an SPM (Fig. 4b) are processed in three chains in order to: (i) localize the wrist line (red arrows

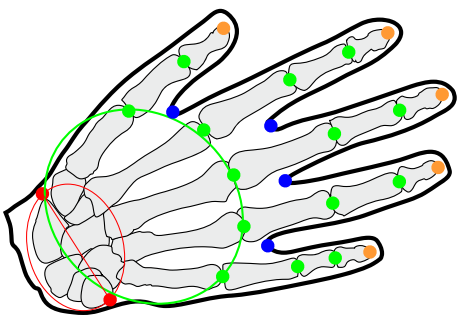

Fig. 2 Hand landmarks: fingertips (*orange*), knuckles (*green*), wrist points (*red*), and the concavities between the digits (*blue*). The green ellipse indicates the palm region and the red one shows the wrist region

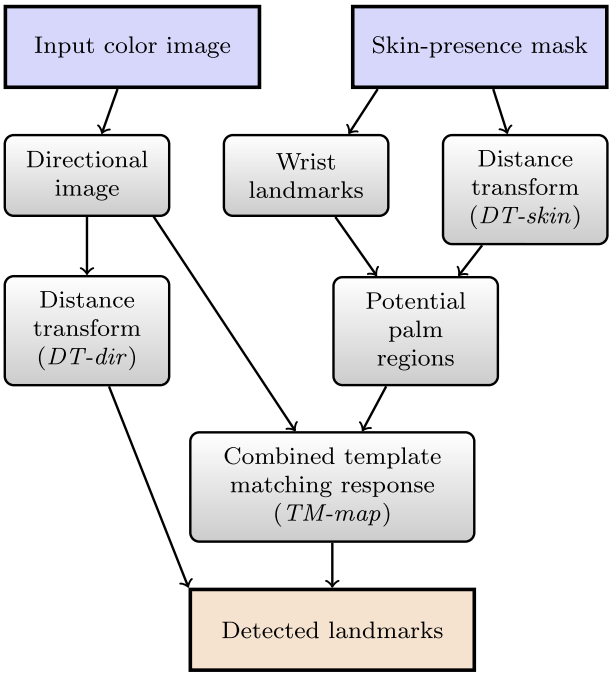

Fig. 3 Flowchart of the proposed algorithm for localizing hand landmarks

in Fig. 4), (ii) find potential palm regions (green arrows), and (iii) detect the landmark candidates (blue and purple arrows).

As outlined earlier in Section 3.2, in order to locate the wrist line (i.e., the points termed W_1 and W_2), at first the hand orientation angle is estimated (Fig. 4i). The horizontal profile of the rotated SPM (Fig. 4j) is analyzed to determine the wrist position (Fig. 4k). The detected wrist line splits the entire mask into two parts, and for each part a global maximum is found in the distance map obtained after applying the distance transform to the hand contour (Fig. 4h). Each maximum (termed P_1 and P_2) is considered as the center of a potential palm region, and it is characterized by a radius r of the incircle of the SPM (i.e., the value of the global maximum).

The landmark candidates are determined as local curvatures detected in the directional image of a hand (Fig. 4c) as well as in the contour of the SPM. The former is subject to the circle-based template matching (Fig. 4e), while the latter is processed with the disc-based template matching (Fig. 4f), and these two template matching responses are combined with each other (Fig. 4g). The directional image presents the hand contour along with the inner contours of digits inside the SPM. The circle-shaped template applied to the directional image makes it possible to locate the fingertips of folded digits, while the disc-shaped template matching map applied to the SPM highlights the fingertips of extended digits. The template radius is set based on the detected palm region size. Subsequently, each local maximum in the combined template matching response is considered as a landmark candidate. Using the distance transform map (Fig. 4d) obtained from the directional image, the whole potential digit is localized from the landmark candidate and it is verified based on the potential palm regions. The procedure for detecting the landmark candidates is presented in detail in the next subsection.

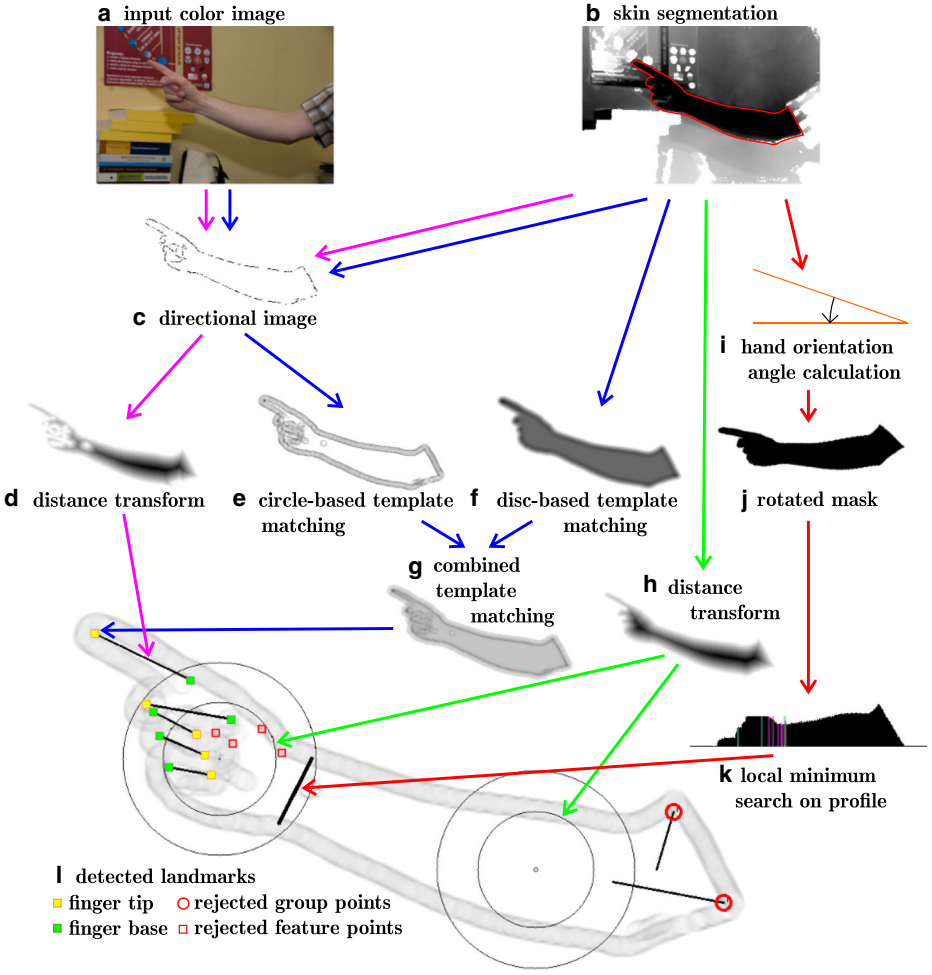

Fig. 4 Examples of the results obtained at subsequent steps of detecting the landmarks

The palm region, in which more landmarks are detected, is considered as the correct one, while the other one is rejected. The landmark candidates associated with the accepted palm region are considered as the final detected landmarks (Fig. 4l).

4.3 Detecting landmark candidates

The candidates for the landmarks are detected using three images generated from the original color image and the SPM, namely: (i) the distance map obtained after applying the distance transform to the directional image (termed *DT-dir*), (ii) the distance map obtained after applying the distance transform to the contour of the SPM (termed *DT-skin*), and (iii) the template matching map (termed *TM-map*, which is the sum of the circle-shape

template matching applied to the directional image and the disc-shape template matching applied to the SPM. The procedure for generating the directional image has been presented earlier in Section 3.1, and the segments in the directional image are found only inside the SPM. Radius of the template (r_T) used for the template matching is set based on the detected palm regions size ($r_T = r/(2\sigma)$, where σ is the template size factor). The values in all these three maps are normalized within the range [0; 1].

The procedure for detecting the landmarks is presented in Algorithm 1. It requires the position of the wrist (W_1 and W_2) and two potential palm regions defined by their centers (P_1 and P_2) and radii (r_1 and r_2). The detection itself is an iterative process. At first, the global maximum in the template matching response is located (line 4) and it is masked by rendering a disc over the response map (line 5). The rendered disc has the same radius as the template (r_T). A new landmark is added to the set $\mathcal{D}_1$ or $\mathcal{D}_2$ (line 8), only if it is positioned at a sufficient distance from the skin mask boundary—the value of *DT-skin* must be high enough, larger than a certain threshold controlled with the $\mathcal{T}_{DT}$ parameter (line 6). The set ($k \in \{1, 2\}$) is selected based on the D_i location related to the line passing through the wrist points W_1 and W_2 (line 7).

Algorithm 1 Landmark candidates detection

Require: *DT-dir*, *DT-skin*, *TM-map*
Output: Two sets of detected landmarks: $\mathcal{D}_1$, $\mathcal{D}_2$

1: **function** DetectLandmarks(W_1, W_2, P_1, r_1, P_2, r_2)
2: $i \leftarrow 1$;
3: **for** $c := 1$ **to** N **do**
4: $D_i \leftarrow \text{argmax}(TM\text{-}map)$;
5: RenderDisc($TM\text{-}map$, D_i, r_T);
6: **if** $DT\text{-}skin(D_i) \geq \mathcal{T}_{DT} \cdot r_T$ **then**
7: $k \leftarrow$ DetermineRegion(W_1, W_2, D_i); ▷ $k \in \{1, 2\}$
8: AddElement($\mathcal{D}_k$, D_i);
9: **if** $\|D_i P\| > \lambda \cdot r_k$ **then** ▷ The digit is extended
10: $D_{i+1} \leftarrow$ Follow($D_i, inside$, r_k, P_k);
11: **else** ▷ The digit is folded
12: $D_{i+1} \leftarrow$ Follow($D_i, outside$, r_k);
13: **end if**
14: **if** $D_i \neq D_{i+1}$ **then**
15: AddElement($\mathcal{D}_k$, D_{i+1});
16: $i \leftarrow i + 1$; ▷ D_{i+1} added to the candidates
17: **end if**
18: **end if**
19: $i \leftarrow i + 1$;
20: **end for**
21: **return** $(\mathcal{D}_1, \mathcal{D}_2)$;
22: **end function**

Depending on the distance of D_i from the palm center, it is determined whether the digit is folded or extended (line 9). For every new landmark D_i, its corresponding landmark D'_i is localized by following the maxima path in the *DT-dir* map. These two corresponding points D_i and D'_i are supposed to belong to the same digit, hence they are expected to be connected with a path. If the new point is localized, then it is added to the set (line 15). When a digit is folded, the maximum may appear at the knuckle rather than at the fingertip. Hence, if it is determined that the finger is folded, then we assume that the fingertip is located closer to the palm center P.

Algorithm 2 Following the local maximum

Require: *DT-dir*
Output: Location of the corresponding landmark D'

```
1: function FOLLOW(D, direction, r, P)
2:     stop ← false;
3:     RENDERDISC(DT-dir, D, DT-dir(D));
4:     repeat
5:         D' ← argmax(GETDISC(DT-dir, D', 2·DT-dir(D')));
6:         RENDERDISC(DT-dir, D', DT-dir(D'));
7:         if direction = inside then
8:             if ||D'P|| < r then
9:                 stop ← true;
10:            end if
11:        else
12:            if ||DD'|| > r then
13:                stop ← true;
14:            end if
15:        end if
16:        if DT-dir(D') = 0 then
17:            stop ← true;
18:        end if
19:    until stop
20: end function
```

Algorithm 2 presents how the maxima path is followed from D in order to locate D'. First, a disc is rendered in the *DT-dir* map at the current location ($D' = D$) to mask the distance map. The disc's radius is equal to the value at the current position in the distance transform map (lines 3 and 6). Subsequently, a maximum is found in the *DT-dir* map within the range of $2 \cdot DT\text{-}dir(D')$ from the current position, and the latter is updated (line 5). This process is iteratively repeated as long as there are non-zero values in the neighborhood of the current location (line 17). Also, depending on whether the digit is supposed to be folded or extended, different distance conditions must be met to stop the searching process (lines 7–15).

5 Experimental validation

In order to evaluate our algorithm, we have compared it with four state-of-the-art methods [6, 45, 51, 54], enlisted in Table 1. As reported earlier in Section 2.2, they represent different approaches towards localizing the landmarks, namely they are based on template matching [45], distance transform [6] and contour analysis [54]. Moreover, we report the

Table 1 Methods investigated on the experimental basis and average processing times for 400×600 images

Abbreviation	Full name	Processing time
C2W	Contour-to-wrist distance [54]	186 ms
DT	Distance transform-based approach [6]	75 ms
TM	Circular template matching [45]	138 ms
SGONG	Self-growing and self-organized neural gas [51]	162 ms
DIB	Proposed, directional image based algorithm	217 ms

results obtained with the SGONG method [51], which does not fall into any of these categories. All the algorithms were implemented in C++ language, and the tests were conducted using a computer equipped with an Intel Core i7 2.3 GHz (4 GB RAM) processor.

5.1 Data sets

There have been a number of databases created to evaluate the algorithms for gesture recognition and some of them include the ground-truth annotations on the landmark locations. The summary of available sets is presented in Table 2. For each set, we quote the number of samples, classes and subjects, provided that they are explicitly given in the database description. It may be noticed that besides our two hand gesture recognition (HGR) sets (marked as bold), there are only two data sets (ColorTip [52] and Dexter 1 [50]) with landmarks localization annotated. Moreover, in these two cases, the metadata include only the fingertips locations.

The databases were created for various purposes and they contain 2D images or video sequences, in some cases extended with the depth data acquired using a Kinect sensor or ToF cameras. The Bosphorus set [7] is used to evaluate biometric identification from hand shapes, hence the images come from many different individuals presenting a hand with extended digits. The Dexter 1 set was created to measure accuracy of fingertip tracking in video sequences. Here, different data modalities are available, including RGB 2D images, ToF and Kinect depth data.

In order to properly evaluate the algorithms for hand landmark localization, we used our HGR set of hand images (available at http://sun.aei.polsl.pl/~mkawulok/gestures). It consists of two parts, namely HGR1 which contains 899 low-quality images (of resolution between 174×131 and 640×480 pixels) acquired in uncontrolled lighting conditions and HGR2 with 659 high-quality images acquired in controlled lighting conditions, organized

Table 2 Publicly available data sets of hand images (our sets are marked as bold)

Name	Landmarks	Data type	Samples	Classes	Subjects
Bosphorus [7][a]	None	2D (still)	ca. 6000	918*	918
CHGD [27][b]	None	2D (seq.)	900	9	2
ColorTip [52][c]	Fingertips	Kinect (seq.)	7**	9	7
Dexter 1 [50][d]	Fingertips	Multi (seq.)	7	—	1
HGds [35][e]	None	ToF (still)	—	—	11
HGR1 [38]	25 landmarks	2D (still)	899	25	12
HGR2	25 landmarks	2D (still)	659	32	18
SKIG [33][f]	None	Kinect (seq.)	2160	10	6

[a] http://bosphorus.ee.boun.edu.tr/hand

[b] http://www.iis.ee.ic.ac.uk/icvl/ges_db.htm

[c] https://imatge.upc.edu/web/res/colortip

[d] http://handtracker.mpi-inf.mpg.de/projects/handtracker_iccv2013/dexter1.htm

[e] http://www-vpu.eps.uam.es/DS/HGds

[f] http://lshao.staff.shef.ac.uk/data/SheffieldKinectGesture.htm

* – each subject (rather than a gesture) forms a separate class

** – each sequence consists of 600-2000 frames presenting different gestures

into two series: (i) HGR2A containing 85 images of 3 individuals (3104×4672 pixels), and (ii) HGR2B containing 574 images of 18 individuals (3264×4928 pixels). As the images present mainly hand regions, we have downsampled them to the dimensions not larger than 400×600 pixels—we have observed that with such resolution, the detection scores are not affected, while keeping the processing times at an acceptable level. In addition, we have selected 65 images of hands with extended digits (mostly the gestures presenting "four" and "five" in American Sign Language [60]), in which the landmarks should be easily detected—we refer to this subset as HGR2B-easy. Every image in the set presents a single-hand gesture, and the ground-truth data encompass the SPM and locations of 25 landmarks. Our data sets (HGR1 and HGR2) contain images acquired with controlled and uncontrolled background, which makes skin segmentation challenging [21, 22]. In the database, we provide ground-truth SPMs—this makes it possible to validate skin segmentation algorithms, and also enables evaluating hand shape recognition [36] as well as landmark detection and localization relying on the ground-truth data, so as to prevent error propagation. Therefore, all of the results reported in this paper (for each method from Table 1) have been obtained using ground-truth SPMs. Naturally, if the human skin regions are segmented automatically, then the landmark detection scores are worse than relying on the ground-truth masks, which we demonstrated in our earlier work for the wrist localization case [11].

5.2 Evaluation procedure

The algorithms were evaluated in order to examine the accuracy of the landmarks localization. After detection and localization, a set of detected landmarks $\{D\}$ is obtained, and it is compared against the set of ground-truth landmarks $\{G\}$. First of all, the mutual distances

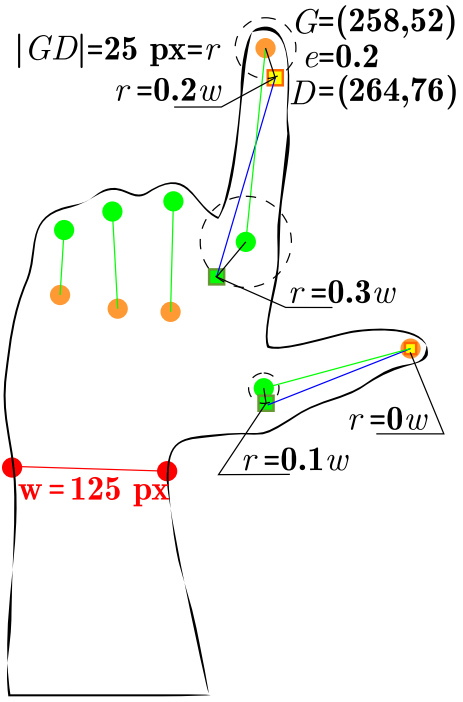

Fig. 5 Example of localization errors

between each detected landmark D and every ground-truth landmark G are computed and sorted in the ascending order. Subsequently, the pairs (G, D) are created starting from the shortest distance $\|GD\|$, and the points G and D are removed from both sets, so that each point is matched with a single point from the other set, or none, if the cardinality of both sets is not equal. The distances are initially measured in pixels, which basically depends on the input image size. Therefore, the localization error is normalized using the hand size estimated based on the ground-truth wrist width w. The localization error (e) of each detected landmark D, in relation to the closest ground-truth landmark G from the ground-truth set is calculated as $e = \|GD\|/w$. In fact, the wrist width w is derived from the annotated ground-truth locations (W_1 and W_2), hence actually this is the length of the hand silhouette diagonal measured at the wrist level. This means that w is an estimation of the real wrist width, whose precision depends on the hand rotation—it is slightly shorter for hand profiles than for hands frontally oriented to the camera. We have also considered using the hand silhouette area to normalize the error, but here the difference would be even more significant.

An example showing different localization errors is presented in Fig. 5. The annotated ground-truth points include: two wrist points (red circles), fingertips (orange circles) and digit knuckles (green circles), while the detected landmarks are annotated using the squares of corresponding color. In the case of the fingertip of the index finger, the distance between the ground-truth and detected point is 25 pixels. The width of the wrist is 125 pixels, hence the calculated localization error is $e = 0.2$. In the figure, we show a circle with the radius equal to the error value $e = 0.2$ to present the possible displacement of the detected point, assuming the same localization error. In the figure, two other detected points are presented with the localization error $e = 0$, and $e = 0.3$.

For every ground-truth landmark G, it is necessary to determine whether it has been correctly detected with a certain localization error e, or whether it has not been detected at all. The latter may happen in two cases, namely: (i) there is no detected landmark matched with the ground-truth landmark, or (ii) the localization error between the matched points is too large to consider it as a correct detection. Hence, we introduced the error acceptance threshold (E)—if the error e is over the threshold, then the landmark is not considered as correctly detected. For the validated data sets, we show cumulative error curves—the horizontal axis presents the error value (i.e., the acceptance threshold E), and the vertical one—the percentage of correctly detected landmarks.

In order to evaluate the quantity of correctly detected points, it is necessary to set appropriate value of the acceptance threshold E. Basically, each landmark corresponds to a certain

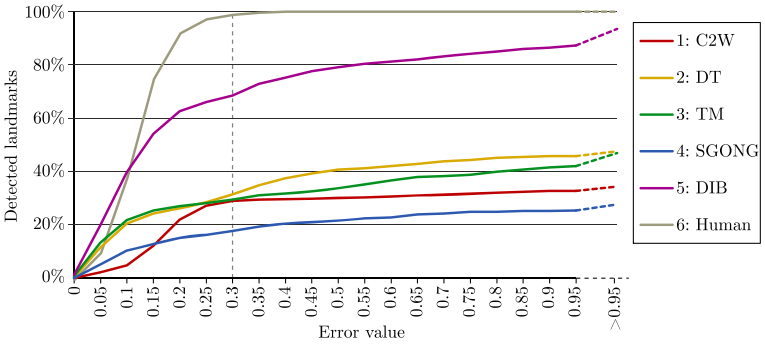

Fig. 6 Cumulative error curves obtained for the HGR2A set

Table 3 The parameters controlling the proposed method, their optimal values, and the ranges of low sensitivity

Symbol	Value	Range	Description
$\mathcal{T}_M$	0.002	$(1 \cdot 10^{-4}; 1 \cdot 10^{-2})$	Direction magnitude threshold
σ	3	(2.5; 3.5)	Template size factor
$\mathcal{T}_{DT}$	0.1	(0.0; 0.3)	Distance threshold factor for landmark candidates
λ	1.6	(1.0; 1.8)	Palm radius factor
N	10	(7; 15)	Maximal number of iterations for searching the landmark candidates

region rather than to a point, so we have decided to take into account the discrepancy between the locations indicated by different human experts. Therefore, we have asked two experts to locate the landmarks (the wrist, finger bases and fingertips) in the images from the HGR2A set. The first expert indicated 826 landmarks, while the second one—760. Subsequently, we measured the errors between the matched landmarks, and the cumulative error curve is presented in Fig. 6 (Human). The first expert was treated as the ground-truth here, and the second one was considered as the "detector". It can be seen that the curve stabilizes for $e = 0.3$, and we used this value to distinguish between correct and incorrect localization in our study.

If a ground-truth landmark is not detected (i.e., no detected landmark is within the radius equal to $w \cdot E$), then the landmark is regarded as false negative (FN). On the other hand, if a detected landmark is not matched with any ground-truth landmark, then it is regarded as false positive (FP). The correctly detected landmarks form the set of true positives (TP). Based on these numbers cumulated for the entire test set, we compute the recall:

$$rec = \frac{TP}{TP + FN}, \tag{4}$$

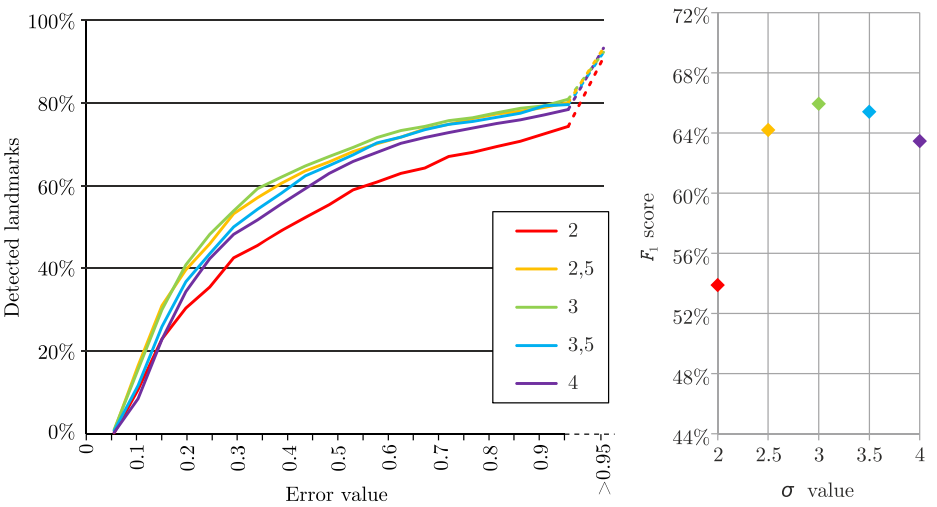

Fig. 7 Cumulative error curves (*left*) and F_1 scores (*right*) obtained for different values of the template size factor (σ)

Table 4 Evaluation scores obtained using various methods

Algorithm	#GT	#Det	FN	TP	FP	rec	$prec$	F_1 score
HGR1:								
1: C2W	3588	2662	1624	1964	698	54.74%	73.78%	0.6285
2: DT	7222	3016	5319	1903	1113	26.35%	63.10%	0.3718
3: TM	3588	2102	2109	1479	623	41.22%	70.36%	0.5199
4: SGONG	7222	2479	5928	1294	1185	17.92%	52.20%	0.2668
5: DIB	7222	6819	3507	3715	3104	51.44%	54.48%	0.5292
HGR2A:								
1: C2W	414	286	172	242	44	58.45%	84.61%	0.6914
2: DT	826	395	566	260	135	31.48%	65.82%	0.4259
3: TM	414	387	173	241	146	58.21%	62.27%	0.6017
4: SGONG	826	230	679	147	83	17.79%	63.91%	0.2784
5: DIB	826	772	264	562	210	68.04%	72.79%	0.7034
6: Human	826	760*	75	751	9	90.92%	98.81%	0.9470
HGR2B:								
1: C2W	2423	1600	1290	1133	467	46.76%	70.81%	0.5633
2: DT	5200	1858	4043	1157	701	22.25%	62.27%	0.3279
3: TM	2423	2308	1097	1326	982	54.72%	57.45%	0.5606
4: SGONG	5200	929	4626	574	355	11.04%	61.78%	0.1873
5: DIB	5200	4662	2525	2675	1987	51.44%	57.38%	0.5425
HGR2-easy:								
1: C2W	295	289	14	281	8	95.25%	97.23%	0.9623
2: DT	295	278	26	269	9	91.19%	96.76%	0.9389
3: TM	295	294	14	281	13	95.25%	95.58%	0.9542
4: SGONG	295	283	94	201	82	68.13%	71.02%	0.6955
5: DIB	295	286	35	260	26	88.14%	90.90%	0.8950

#GT– number of ground-truth landmarks annotated by an expert

#Det– number of detected landmarks

*– number of landmarks annotated by the second expert

the precision:

$$prec = \frac{TP}{TP + FP}, \tag{5}$$

and the F_1 score (a harmonic mean of the both):

$$F_1 = \frac{2 \cdot rec \cdot prec}{rec + prec}. \tag{6}$$

5.3 Sensitivity analysis

There are several parameters that control the behavior of the proposed algorithm, and their values are reported in Table 3. We have run a number of tests to find an optimal value of each parameter, as well as we analyzed the sensitivity of the algorithm. The results obtained

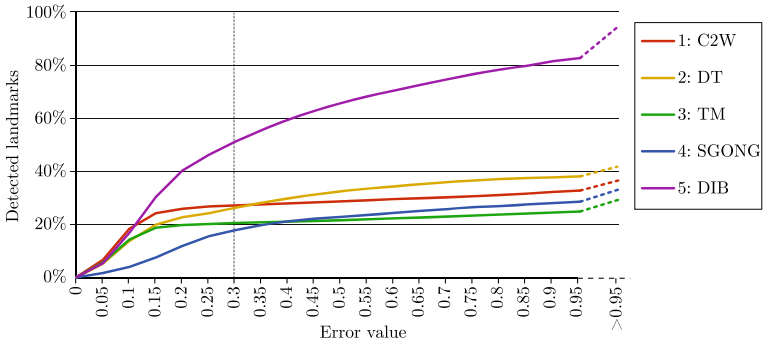

Fig. 8 Cumulative error curves obtained for the HGR1 test set

for different values of the template size factor σ are reported in Fig. 7. It can be seen from the figure that the sensitivity is low within the range $2.5 \geq \sigma \geq 3.5$, and it increases for $\sigma < 2.5$. Following the same approach, we tuned all the parameters and the ranges of their low sensitivity are reported in Table 3.

5.4 Comparison with the state of the art

The scores obtained using several state-of-the-art methods are reported in Table 4. The results are presented for HGR1, HGR2A and HGR2B sets, as well as for the HGR2B-easy series. Also, as mentioned earlier in Section 5.2, the discrepancy between two human experts is reported for the HGR2A set. The cumulative error curves for each test subset are presented in Figs. 6, 8, 9 and 10. In Table 4, the number of ground-truth landmarks is different in case of C2W and TM methods for HGR1, HGR2A and HGR2B. As these algorithms do not make it possible to detect the landmarks inside the hand masks, it would not be fair to evaluate them taking into account such points. Therefore, in the table we quote the scores obtained exclusively using the landmarks located at the contour.

For HGR1, HGR2A and HGR2B sets it may be seen that the proposed method allows for detecting more landmarks than the existing approaches, however many of them are false positive. As a result, the recall is much better here at the cost of the precision. Overall,

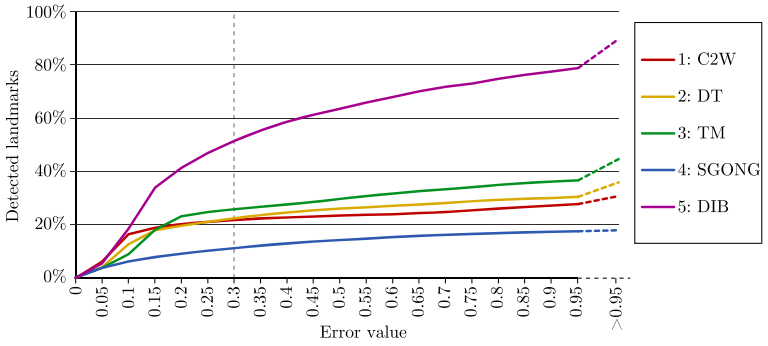

Fig. 9 Cumulative error curves obtained for the HGR2B test set

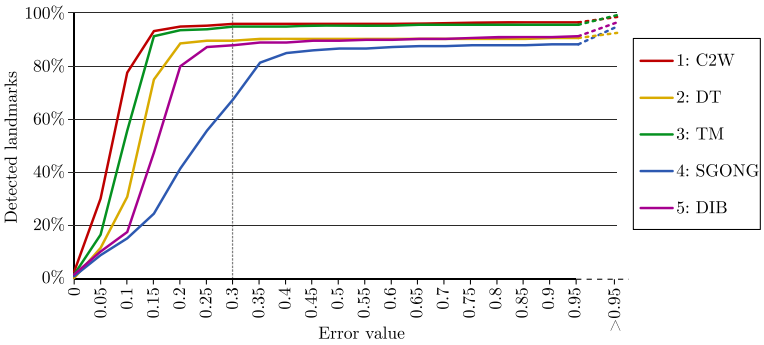

Fig. 10 Cumulative error curves obtained for the HGR2B-easy images

it may be seen that the F_1 score is the highest using our method for the HGR2A set, and for HGR1 and HGR2B it is close to the scores obtained with C2W and TM (which are measured only for the landmarks located at the contour for these methods). It is worth noting that the obtained scores are rather low for all the investigated algorithms, lower than those reported in many other works. The reason is that our data set contains very difficult cases, and to support this argument we have selected the images (i.e., the HGR2B-easy series),

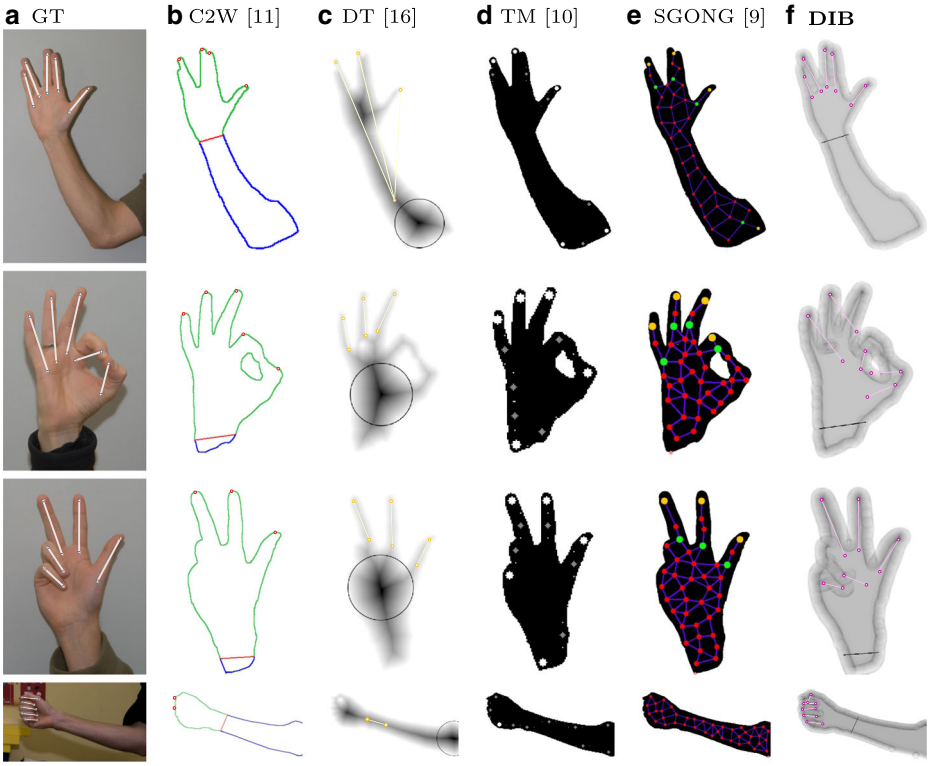

Fig. 11 Examples of the landmarks detected using different algorithms

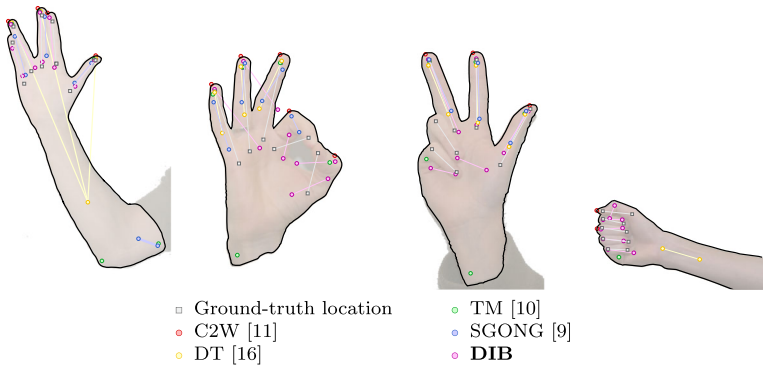

Fig. 12 Comparison of the detected landmarks using different algorithms

which theoretically should be easy to analyze. Moreover, we have limited the validation to the fingertips. From Table 4 and Fig. 10, it may be seen that all of the methods are quite effective here, including our method, and the numbers are similar to those reported in other papers.

In Table 1, we report the average processing times per 400×600 image. It may be seen that our algorithm is slightly slower than the alternative methods, but still it may process the images at ca. 5 images per second, which allows for real time applications. The alternative methods may analyze between 5 (for C2W [54]) and 13 images per second (for DT [6]). As we have not observed any improvement when processing images of a larger resolution, the only cost of applying the algorithm to larger images would be concerned with scaling the hand region down to 400×600.

Examples of the landmarks detected using the investigated algorithms along with their ground-truth locations (GT) are presented in Fig. 11 and they are illustrated altogether in Fig. 12. The following conclusions may be drawn for each algorithm:

1. C2W—analysis of the contour makes it possible to localize the landmarks being the local maxima of the distance from the wrist. This method does not detect the digit bases, and also it fails when two digits are joined with each other—in such cases, only a single landmark is detected instead of two. Overall, this makes the method quite effective for detecting fingertips in "easy" images, even if their quality is low (which may be seen for the scores obtained for HGR1).
2. DT—the method often fails, if the SPM includes the forearm region. Moreover, the joined digits are detected as a single one as for the C2W method.
3. TM—the TM-based analysis may detect the joined fingertips, however it often produces false positives in the contour curvatures. Also, this method does not detect the knuckles.
4. SGONG—the landmarks are detected by expanding the neural gas, and in most cases the joined digits are not distinguished from each other.
5. DIB—the proposed method makes it possible to detect the joined digits (both the fingertips and the knuckles), which results in the highest recall taking into account all the ground-truth landmarks. The main drawback here lies in detecting more false positives compared to the alternative approaches.

6 Conclusions

In this paper, a new method for detecting hand landmarks has been presented. The main contribution lies in analyzing the directional image, which makes it possible to localize the landmarks positioned inside the hand region. The existing methods do not have the capacity to deal with such cases, which has been confirmed during extensive experimental validation, whose results we reported and discussed in the paper.

Our ongoing research is aimed at taking advantage of local features extracted from the distance maps. This may slow down the algorithm, but would be helpful in reducing the false positives. Furthermore, we want to localize all the knuckles along the maxima paths that are currently used to detect the digits. Finally, we intent to take advantage of the extracted information in our shape-based gesture recognition system [36, 37].

Acknowledgments The work of TG and AG has been supported by Institute of Automatic Control BK-227/RAu1/2015/1 funds in the year 2015. The work of MK has been supported by Institute of Informatics internal funds no. BKM-515/RAu2/2015. The research was performed on the infrastructure supported by POIG.02.03.01-24-099/13 grant: "GeCONiI–Upper Silesian Center for Computational Science and Engineering.

References

1. Appenrodt J, Al-Hamadi A, Michaelis B (2010) Data gathering for gesture recognition systems based on single color-, stereo color-and thermal cameras. International Journal of Signal Processing. Image Processing and Pattern Recognition 3(1):37–50
2. Bellarbi A, Benbelkacem S, Zenati-Henda N, Belhocine M (2011) Hand gesture interaction using color-based method for tabletop interfaces. In: Proceedings IEEE International Symposium on Intelligent Signal Processing (WISP), pp. 1–6
3. Chaudhary A, Raheja JL, Das K, Raheja S (2013) Intelligent approaches to interact with machines using hand gesture recognition in natural way: A survey. CoRR abs/1303.2292
4. Cooper H, Bowden R (2007) Large lexicon detection of sign language. In: Human–Computer Interaction, pp. 88–97. Springer
5. Czupryna M, Kawulok M (2012) Real-time vision pointer interface. In: Proceedings International Symposium ELMAR 2012, pp. 49–52
6. Dung L, Mizukawa M (2009) Fast hand feature extraction based on connected component labeling, distance transform and hough transform. J Rob Mechatronics 21:726–738
7. Dutağacı H, Sankur B, Yörük E (2008) Comparative analysis of global hand appearance-based person recognition. J Electron Imaging 17(1):011018–011018
8. Erol A, Bebis G, Nicolescu M, Boyle RD, Twombly X (2007) Vision-based hand pose estimation: A review. Comput Vis Image Underst 108(1–2):52–73
9. Feng Z, Yang B, Chen Y, Zheng Y, Xu T, Li Y, Xu T, Zhu D (2011) Features extraction from hand images based on new detection operators. Pattern Recogn 44(5):1089–1105. doi:10.1016/j.patcog.2010.08.007
10. Ge SS, Yang Y, Lee TH (2008) Hand gesture recognition and tracking based on distributed locally linear embedding. Image Vis Comput 26(12):1607–1620
11. Grzejszczak T, Nalepa J, Kawulok M (2013) Real-time wrist localization in hand silhouettes. In: Burduk R, Jackowski K, Kurzynski M, Wozniak M, Zolnierek A (eds) Proceedings International Conference

on Computer Recognition Systems CORES 2013. Advances in Intelligent Systems and Computing, vol. 226, pp. 439–449. Springer
12. Grzejszczak T, Galuszka A, Niezabitowski M, Radlak K (2014) Comparison of hand feature points detection methods. In: Proceedings DoCEIS, pp. 167–174
13. Grzejszczak T, Mikulski M, Szkodny T, Jedrasiak K (2012) Gesture based robot control. In: Bolc L, Tadeusiewicz R, Chmielewski LJ, Wojciechowski KW (eds) ICCVG. Lecture Notes in Computer Science, vol. 7594, pp. 407–413. Springer
14. Hagara M, Pucik J (2013) Fingertip detection for virtual keyboard based on camera. In: Proceedings 23rd International Conference Radioelektronika, pp. 356–360
15. Hoshino K, Tomida M (2010) Quick and accurate estimation of human 3D hand posture. Intell Serv Robot 3(1):11–19
16. Ibraheem NA, Khan R (2012) Survey on various gesture recognition technologies and techniques. Int J Comput Appl 50(7):38–44
17. Infantino I, Chella A, Macaluso HDI (2003) Visual control of a robotic hand. In: Proceedings International Conference on Intelligent Robots and Systems, pp. 1266–1271
18. Jones MJ, Rehg JM (2002) Statistical color models with application to skin detection. Int J Comput Vis 46:81–96
19. Kakumanu P, Makrogiannis S, Bourbakis NG (2007) A survey of skin-color modeling and detection methods. Pattern Recogn 40(3):1106–1122
20. Kawulok M, Nalepa J, Kawulok J (2014) Skin detection and segmentation in color images. In: Celebi ME, Smolka B (eds) Advances in Low-Level Color Image Processing. Lecture Notes in Computational Vision and Biomechanics, vol. 11, pp. 329–366. Springer
21. Kawulok M, Kawulok J, Nalepa J, Smolka B (2014) Self-adaptive algorithm for segmenting skin regions. EURASIP Journal on Advances in Signal Processing 2014(170)
22. Kawulok M, Kawulok J, Nalepa J (2014) Spatial-based skin detection using discriminative skin-presence features. Pattern Recogn Lett 41:3–13
23. Kawulok M, Nalepa J (2012) Support vector machines training data selection using a genetic algorithm. In: Gimel'farb G, Hancock E, Imiya A, Kuijper A, Kudo M, Omachi S, Windeatt T, Yamada K (eds) Structural, Syntactic, and Statistical Pattern Recognition. Lecture Notes in Computer Science, vol. 7626, pp. 557–565. Springer
24. Kawulok M, Szymanek J (2012) Precise multi-level face detector for advanced analysis of facial images. IET Image Process 6(2):95–103
25. Kerdvibulvech C (2014) A methodology for hand and finger motion analysis using adaptive probabilistic models. EURASIP J Embed Syst 2014(1)
26. Kim D, Lee J, Yoon H-S, Kim J, Sohn J (2013) Vision-based arm gesture recognition for a long-range human-robot interaction. J Supercomput 65(1):336–352
27. Kim T-K, Cipolla R (2009) Canonical correlation analysis of video volume tensors for action categorization and detection. IEEE Trans Pattern Anal Mach Intell 31(8):1415–1428
28. Koller O, Ney H, Bowden R (2013) May the force be with you: Force-aligned signwriting for automatic subunit annotation of corpora. In: Proceedings IEEE International Conference on Automatic Face and Gesture Recognition, pp. 1–6
29. Krejov P, Bowden R (2013) Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima. In: Proceedings IEEE International Conference on Automatic Face and Gesture Recognition (FG), pp 1–7
30. Liang H, Yuan J, Thalmann D (2012) 3D fingertip and palm tracking in depth image sequences. In: Proceedings 20th ACM International Conference on Multimedia, MM '12, pp. 785–788, ACM, New York
31. Liang H, Yuan J, Thalmann D, Zhang Z (2013) Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization. Vis Comput 29(6–8):837–848
32. Licsar A, Sziranyi T (2004) Hand gesture recognition in camera-projector system. In: Sebe N, Lew MS, Huang TS (eds) Proceedings ECCV Workshop on HCI. Lecture Notes in Computer Science, vol. 3058, pp. 83–93. Springer
33. Liu L, Shao L (2013) Learning discriminative representations from RGB-D video data. In: Proceedings International Joint Conference on Artificial Intelligence, pp 1493–1500. AAAI Press
34. Mitra S, Acharya T (2007) Gesture recognition: A survey. IEEE Transactions on Systems, Man, and Cybernetics. Part C 37(3):311–324
35. Molina J, Escudero-Vinolo M, Signoriello A, Pardas M, Ferran C, Bescos J, Marques F, Martinez J (2013) Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models. Mach Vis Appl 24(1):187–204. doi:10.1007/s00138-011-0364-6

36. Nalepa J, Kawulok M (2014) Fast and accurate hand shape classification. In: Kozielski S, Mrozek D, Kasprowski P, Malysiak-Mrozek B, Kostrzewa D (eds) Beyond Databases, Architectures, and Structures. Communications in Computer and Information Science, vol. 424, pp. 364–373. Springer
37. Nalepa J, Kawulok M (2013) Parallel hand shape classification. In: Proceedings IEEE International Symposium on Multimedia (ISM 2013), pp. 401–402
38. Nalepa J, Grzejszczak T, Kawulok M (2014) Wrist localization in color images for hand gesture recognition. In: Gruca DA, Czachórski T, Kozielski S (eds) Man-Machine Interactions 3. Advances in Intelligent Systems and Computing, vol. 242, pp. 79–86. Springer
39. Oka K, Sato Y, Koike H (2002) Real-time fingertip tracking and gesture recognition. IEEE Comput Graph Appl 22(6):64–71. doi:10.1109/MCG.2002.1046630
40. Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28(6):976–990
41. Rautaray S, Agrawal A (2012) Vision based hand gesture recognition for human computer interaction: a survey. Artif Intell Rev:1–54
42. Ren Z, Yuan J, Meng J, Zhang Z (2013) Robust part-based hand gesture recognition using kinect sensor. IEEE Trans Multimedia 15(5):1110–1120
43. Ren Z, Meng J, Yuan J, Zhang Z (2011) Robust hand gesture recognition with kinect sensor. In: Proceedings ACM International Conference on Multimedia. MM '11 ACM, New York, pp 759–760
44. Ruiz-del-Solar J, Verschae R (2004) Skin detection using neighborhood information. In: Proceedings IEEE International Conference on Automatic Face and Gesture Recognition, pp. 463–468
45. Sato Y, Kobayashi Y, Koike H (2000) Fast tracking of hands and fingertips in infrared images for augmented desk interface. In: Proceedings IEEE International Conference on Automatic Face and Gesture Recognition, pp. 462–467
46. Sahami Shirazi A, Abdelrahman Y, Henze N, Schneegass S, Khalilbeigi M, Schmidt A (2014) Exploiting thermal reflection for interactive systems. In: Proceedings SIGCHI Conference on Human Factors in Computing Systems. CHI '14, pp. 3483–3492 ACM, New York
47. Shi Y, Chen X, Wang K, Fang Y, Xu L (2010) Real-time hand posture analysis based on neural network. In: Proceedings IEEE International Conference on Signal Processing (ICSP), pp. 893–896
48. Sonka M, Hlavac V, Boyle R (2014) Image Processing, Analysis, and Machine Vision. Cengage Learning
49. Srivastava A, Turaga P, Kurtek S (2012) On advances in differential-geometric approaches for 2D and 3D shape analyses and activity recognition. Image Vis Comput 30(6–7):398–416
50. Sridhar S, Oulasvirta A, Theobalt C (2013) Interactive markerless articulated hand motion tracking using RGB and depth data. In: Proceedings IEEE International Conference on Computer Vision (ICCV), pp. 2456–2463
51. Stergiopoulou E, Papamarkos N (2009) Hand gesture recognition using a neural network shape fitting technique. Eng Appl Artif Intell 22(8):1141–1158
52. Suau X, Alcoverro M, Lopez-Mendez A, Ruiz-Hidalgo J, Casas JR (2014) Real-time fingertip localization conditioned on hand gesture classification. Image Vis Comput 32(8):522–532
53. Sun Y, Reale M, Yin L (2008) Recognizing partial facial action units based on 3D dynamic range data for facial expression recognition
54. Tanibata N, Shimada N, Shirai Y (2002) Extraction of hand features for recognition of sign language words. In: Proceedings International Conference on Vision Interface, pp. 391–398
55. Tang M (2011) Recognizing hand gestures with Microsoft's Kinect. Palo Alto: Department of Electrical Engineering of Stanford University:[sn]
56. Thippur A, Ek CH, Kjellstrom H (2013) Inferring hand pose: A comparative study of visual shape features. In: Proceedings IEEE International Conference on Automatic Face and Gesture Recognition, pp. 1–8
57. Tsagaris A, Manitsaris S, Hatzikos E, Manitsaris A (2012) Methodology for finger gesture control of mechatronic systems. In: Proceedings International Symposium MECHATRONIKA, pp. 1–6
58. Vanco M, Minarik I, Rozinaj G (2014) Evaluation of static hand gesture algorithms. In: Proceedings International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 83–86
59. Wang Y, Luo Z, Liu J, Fan X, Li H, Wu Y (2013) Real-time estimation of hand gestures based on manifold learning from monocular videos. Multimedia Tools and Applications 71(2):555–574
60. Wilcox S (1992) The Phonetics of Fingerspelling vol. 4. John Benjamins Publishing
61. Yang H-D, Lee S-W (2013) Robust sign language recognition by combining manual and non-manual features based on conditional random field and support vector machine. Pattern Recogn Lett 34(16):2051–2056
62. Yogarajah P, Condell J, Curran K, Cheddad A, McKevitt P (2010) A dynamic threshold approach for skin segmentation in color images. In: Proceedings IEEE International Conference on Image Processing, pp. 2225–2228

Tomasz Grzejszczak was born in 1986 in Zabrze, Poland. He graduated and received his M.Sc. degree in 2005 from the Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Poland, with degree in automatic control. In 2010 he started PhD studies at the Institute of Automatic Control, Silesian University of Technology. His research interests are focused on issues related image processing, optimization and heuristic algorithms.

Michal Kawulok is working as an assistant professor at Institute of Informatics, Silesian University of Technology, Gliwice, Poland. He graduated and received his M.Sc. degree in 2003 and the Ph.D. in 2007, both from Silesian University of Technology. He is an IEEE member. His general research interests are concerned with image analysis and pattern recognition, with particular attention given to human skin segmentation, image colorization, gesture recognition, and facial image analysis.

Adam Galuszka was born on 06.06.1972 in Ruda Slaska, Poland. In 1996 graduated from the Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Poland, with degree in robotics. In 1996 he started PhD studies at the Institute of Automatic Control, Silesian University of Technology. In 2001 he received PhD degree in the field of automation and robotics. In 2014 he received DSc degree. His research interests are focused on issues related to artificial intelligence, in particular to planning algorithms and their computational complexity.