Journal of Real-Time Image Processing (2020) 17:981–992
https://doi.org/10.1007/s11554-019-00850-5

ORIGINAL RESEARCH PAPER

Fast background removal of JPEG images based on HSV polygonal cuts for a foot scanner device

T. Trigano[1] · Y. Bechor[2]

Received: 25 May 2017 / Accepted: 4 January 2019 / Published online: 11 January 2019

Abstract
Foot scanning devices aim to provide information on a patient's foot and to help in diagnosing issues to be corrected with orthoses. The Galaxy foot scanner developed by the Aetrex company aims to provide a computer-aided framework to help physicians in their diagnoses. As numerous embedded devices, used for image processing and 3D-reconstruction, it includes cameras which provide JPEG pictures of the object to reconstruct. In this framework, an important step is the segmentation of the image, to isolate the object of interest, but the JPEG compression introduces artifacts which can lower the performance of any segmentation procedure. In this paper, we suggest a model which takes the artifacts stemming from the JPEG compression into account. The pixels are first sorted into layers of pixels with similar value V in the HSV color space, and the background is modeled by a polygon from an additional picture. Segmentation based on the knowledge of the background and the layer to be processed is then performed. Results obtained with the Galaxy foot scanner illustrate that this method provides good results for segmentation, while being sufficiently fast to be implemented for near real-time applications.

Keywords Image segmentation · JPEG compression · Embedded devices · HSV color space · Foot scanners

1 Introduction

Foot orthoses are molded pieces of rubber, leather, plastic or any other soft synthetic material which are inserted into a shoe. They aim to correct some defect in the foot, ankle or hip biomechanics. They also aim to attenuate pain symptoms (mostly back pains or articulation stress), by balancing the foot in a neutral position. Finding the best orthotic device shape for a specific symptom is still a complex problem and under medical research, as seen in Telfer et al. [14]. However, it is known that well-designed, custom-made foot orthoses may control pain for specific problems, such as an unusually shaped foot or foot rolling towards the arch (over-pronation). To determine the correct type of orthoses, practitioners are increasingly utilizing computer-aided foot scanners. These scanners provide information both on pressure points of the foot and on its morphology, thus helping in diagnosis.

In this line of work, the Galaxy device developed by the Aetrex company is a foot scanning device which measures human feet for the purpose of determining shoe size and insole type. It performs all measurements using electronic and optical means, and does not include any lasers, motors or moving parts. The Galaxy scanner includes 16 cameras in the perimeter of the scanner to capture the view of the foot, as well as two additional cameras to capture the alignment of the foot and to determine whether the foot is over-pronated or supinated (outward roll of the foot during normal motion). The Galaxy foot scanner also includes 16 white light LEDs, positioned all around the device, to illuminate the scene and to remove shadows introduced by the foot and lessen light reflections. It is presented in Fig. 1.

When measurements are performed, each camera of the Galaxy device takes two kind of shots: during the calibration stage, background images (denoted by B in the rest of the paper) without foot are recorded, and during the measurement stage, foot images (denoted by I in the rest of the paper) are captured, as displayed for example in Fig. 2. As seen from this figure of foot images, a foot stands in the middle of the picture, either bare with an unknown skin color,

✉ T. Trigano
thomast@sce.ac.il

[1] Department of Electrical Engineering, Shamoon College of Engineering, Ashdod, Israel

[2] Aetrex Israel, Nes Ziona, Israel

Fig. 1 The Galaxy scanner

or wearing a sock with unknown colors. To reconstruct a reliable 3D-model of the foot, we must isolate the foot from the background, by means of an algorithm sufficiently fast for the application in mind (the overall procedure, include taking the picture and performing the segmentation, cannot exceed 30 s).

The problem of efficient background separation from still images is of great practical importance in numerous applications. One of them, which is becoming increasingly present in the industry world, is the three-dimensional reconstruction of an object, given a set of still images (a thorough discussion in three-dimensional image processing applications can be found in Pears et al. [9]). Common methods used for background removal imply a thresholding operation (see, e.g., Gonzalez and Woods [4]), and usually perform poorly with complex backgrounds. State-of-the-art background removal methods focus more on good segmentation performances for still pictures in complex backgrounds. In such cases, segmentation can be seen as an optimization problem in Markov fields as in Li [6], or as an optimization of the flow in graphs as in the Grabcut algorithm presented in Rother et al. [10, 11], Boykov and Jolly [1], or the use of Bayesian techniques Haines and Xiang [5]. The obtained algorithms provide extremely good segmentation performances, but are usually partially supervised, in the sense that the result must often be refined by the user during the execution. Furthermore, the computational complexity makes them less suited for embedded applications, e.g., Sigal et al. [12], Li et al. [7], Faro et al. [3], Liu and Payeur [8]. Moreover, many of the cited approaches apply to video processing, and exploit the redundancy of the video frames to infer a statistical model of the background. This is not applicable in our case, where we only have access to two still images.

Another problem is that most background removal algorithms are usually tested on pictures encoded in a loss-less format, such as TIFF or PNG. The foot scanner device used in our study is based on cameras which encode recorded pictures in JPEG with a high, lossy, compression rate. This compression step done before the image analysis introduces compression artifacts, and creates distortion in the colors of the image (see the example in the next section). Therefore, the background removal method in our framework should be as robust as possible to these distortions.

The objective of this paper is to propose a fast segmentation method suited for our foot scanning application, while taking into consideration the distortion brought by the JPEG compression. The proposed algorithm splits the data into layers related to different values, and deals with them one by one. For each layer, a background model is learned from a set of pictures without foot. The learning is done by building a convex hull around the points of each layer separately, which can be done using fast algorithms. This model is then compared with the points of a picture which includes a foot for segmentation. The rest of the paper is organized as follows: the following section describes the mathematical model used and the segmentation procedure suggested, which takes into account the JPEG distortion. We present results of simulations performed on a picture database built from the Galaxy foot scanner, and discuss the setting of the parameters on which the proposed algorithm depends. The obtained results

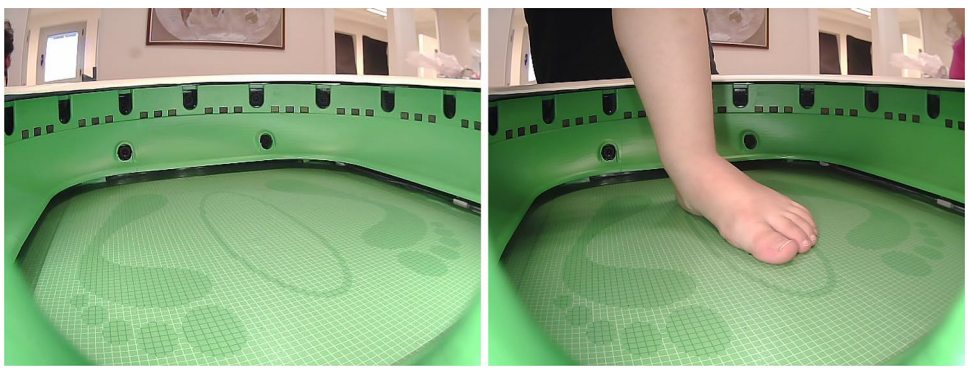


Fig. 2 Examples of a background image B (left) and foot image I (right)

show that, with a correct choice of inner parameters, the proposed methods can provide excellent segmentation results combined with fast execution times.

2 Segmentation based on HSV polygonal cuts

In this section, we present both the model and the novel segmentation algorithm used from background removal, from two images taken from the device. This new method takes into account the distortion stemming from the JPEG compression. Before describing the method itself, we present a preliminary discussion on the problems encountered in the Hue-Saturation-Value (HSV) space when working on JPEG-encoded pictures.

2.1 Problems induced by lossy compression in the HSV decomposition

For segmentation tasks, choosing a suitable color space is very important, since the accuracy of color detection affects segmentation results. The HSV color space (Hue, Saturation, Value) is one of the most used color spaces in the field. The HSV color space appears frequently in numerous applications ranging from image enhancement (see [17]) to feature-based classification as in Chen et al. [2], or in addition to existing segmentation frameworks Silva et al. [13], Wei et al. [16]. Given a pixel $p = (R, G, B)$ described in the RGB coordinate system, coded with L bits, with a maximum component $M = \max\{R, G, B\}$ and minimum component $m = \min\{R, G, B\}$, it can be decomposed into the HSV color space accordingly to the following equations:

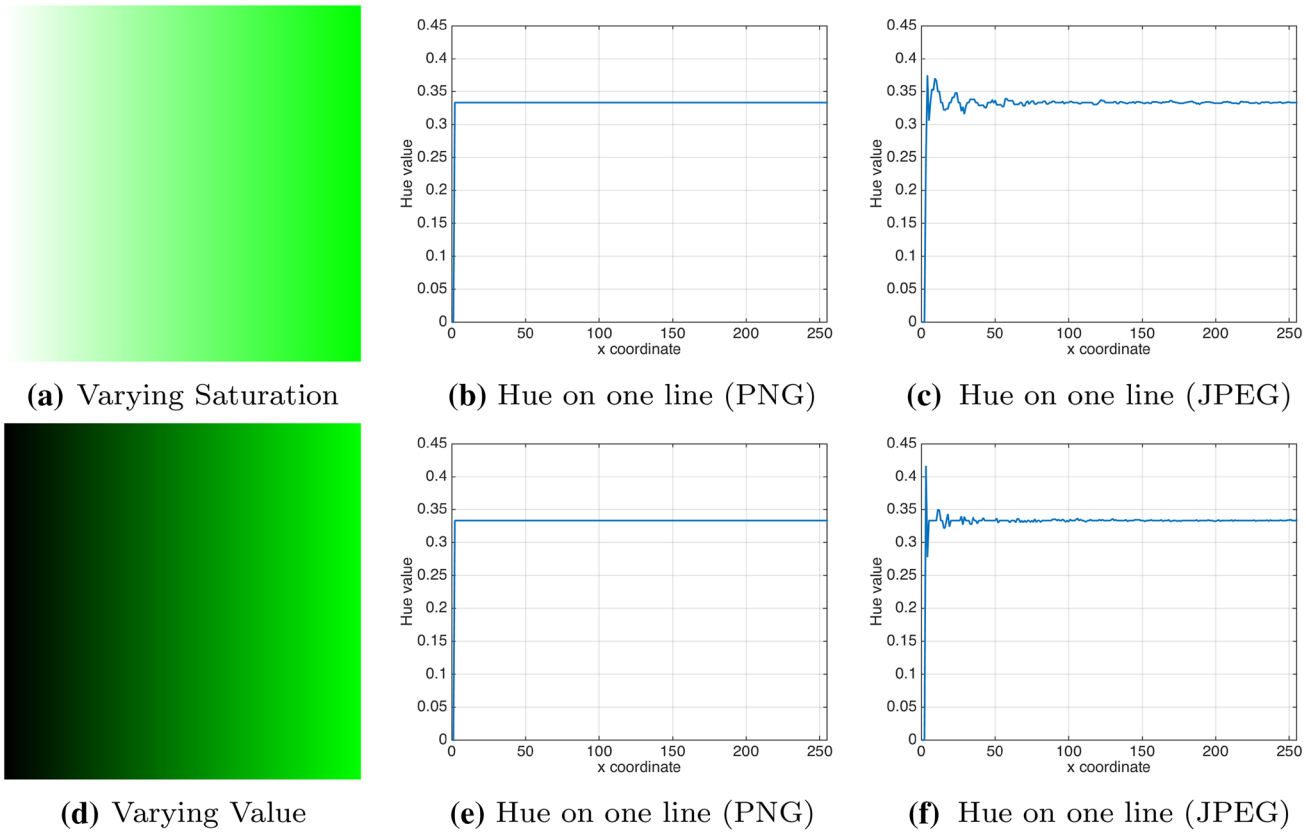
$$H = \begin{cases} 0 & \text{if } M = m. \\ 60^o \times \dfrac{G-B}{M-m} \mod 360^o & \text{if } M = R. \\ 60^o \times \left(\dfrac{B-R}{M-m} + 2\right) & \text{if } M = G. \\ 60^o \times \left(\dfrac{R-G}{M-m} + 4\right) & \text{if } M = B. \end{cases} \tag{1}$$

$$S = \begin{cases} 0 & \text{if } M = m \\ \dfrac{M-m}{M} & \text{otherwise} \end{cases} \tag{2}$$

$$V = \frac{\max\{R, G, B\}}{2^L - 1} \tag{3}$$

The HSV color space is more fitted for color-based segmentation tasks, as it corresponds more closely to human perception of color. In the HSV coordinate system, Saturation is a measure of the lack of whiteness in the color, whereas Hue is defined as the angle from the red color axis, and Value refers to the brightness. However, the cameras used in our foot scanning device, as it is also the case in numerous embedded applications, usually provide pictures in JPEG format, that is after a lossy compression. This compression step cannot usually be circumvented, and yields a significant distortion in the data available on the HSV space. Therefore, it can significantly lower the performances of segmentation algorithms afterwards. This can be intuitively understood from Eq. (1): due to the well-known JPEG compression artifacts, the ratio $M - m$ is susceptible to slight variation (and varies more around the edges of the objects), which causes a large variability in the $1/(M - m)$ of the Hue term, particularly when M is close to m (which is the case for black, white and gray tones). For the sake of argument, Fig. 3 illustrates the distortion brought by the lossy compression on a simple, synthetic, example. In this figure, a synthetic image is obtained by fixing the Hue to $\frac{1}{3}$. In the upper figures, the Value is fixed to 1, while the Saturation varies from 0 to 1. In the lower figures, the Saturation is fixed to 1, while the Value varies from 0 to 1. Both images are saved in JPEG format after moving back to the RGB color space. After performing the image compression, the compressed image is reloaded and transformed to the HSV color space. For simplicity, and due to the image structure, we display the variation of the Hue on one line of the picture only, and make the y coordinate vary. It can be observed that the JPEG compression introduces a distortion of the Hue, which is particularly visible for dark and white tones.

Though the level of distortion can be controlled by changing the compression level, this solution cannot be retained in practice. Indeed, a lower level of compression increases the execution time required both for the transmission of the raw data to the computer which performs the segmentation as well as for the segmentation task itself. In practice, there is a trade-off between the level of compression, the quality of the segmentation of the images and the near real-time aspects of the application itself. This distortion can be extremely problematic, in particular for the development of a foot scanner based on pictures. Indeed, most measurements are done with the patient wearing either black or white socks, which are the most common sock colors. Consequently, segmentation algorithms perform poorly in that case. With these considerations in mind, it is clear that any segmentation method based on the HSV space should take into account the fact that compression artifacts introduce a distortion in the Hue for clear and dark colors. The next section details



(a) Varying Saturation **(b)** Hue on one line (PNG) **(c)** Hue on one line (JPEG)

(d) Varying Value **(e)** Hue on one line (PNG) **(f)** Hue on one line (JPEG)

Fig. 3 Influence of the compression rate on the Hue of a synthetic image. The Hue of clear and dark tones is noisy due to the compression artifacts

the mathematical model used for our algorithm, and the proposed method to circumvent the distortion issue.

2.2 Model and notations

In our framework, an image is modeled by a sequence of three-dimensional vectors:

$$I = \left\{ I_k \in \mathbb{R}^3, \quad 1 \leq k \leq N \right\}. \tag{4}$$

In (4), each vector I_k is related to the k-th pixel of the image. More specifically, if H_k, S_k, V_k represent, respectively, the Hue, Saturation and Value of the k-th pixel, we define

$$I_k \stackrel{\Delta}{=} \left[S_k \cos(H_k);\, S_k \sin(H_k);\, V_k \right]^T; \tag{5}$$

that is, I_k in (5) is a Cartesian representation of the HSV components of the k-th pixel. From (2) and (3), it is straightforward that $0 \leq S_k \leq 1$ and $0 \leq V_k \leq 1$. As discussed before, the JPEG compression of the images at hand introduces color distortion, which is more crucial as the Saturation decreases or as the Value increases.

From the figures obtained on synthetic data in Fig. 3, we can notice that the distortion observed in the Saturation and Value components is not uniformly distributed. Consequently, we define N layers of pixels, sorted accordingly to their Value levels. More precisely, given a sequence $0 = a_0 < a_1 < \cdots < a_N = 1$, the n-th layer of pixels is defined as

$$\mathcal{V}_n(I) \stackrel{\Delta}{=} \left\{ I_k \in I \,;\, a_{n-1} \leq V_k \leq a_n \right\}, \quad 1 \leq n \leq N. \tag{6}$$

Recall that the background used for separation is as uniformly light green as possible, and that the LEDs provide uniform lighting on the scene. Therefore, the pixels in $\mathcal{V}_n$ are most likely to be well separated when n is high, whereas small n indicates that $\mathcal{V}_n$ is less reliable for good segmentation. The choice of the number of Value layers N in (6) is also important. If N is too large, the polygonal representations introduced later on are not statistically representative enough. In addition, the processing time increases, which is problematic in our case. On the other hand, too small a Value of N will give poorer segmentation results. It can be noticed that the layers may not necessarily be uniform, since the distortion introduced by the JPEG encoding is not uniform.

However, the optimal choice of the layer depths a_n is not within the scope of the present paper, and shall be discussed in further contributions.

2.3 Image segmentation by means of polygonal cuts

The problem at hand can be summarized as follows: given one background image $B = \{B_k \in \mathbb{R}^3,\ 1 \leq k \leq N\}$ and one foot image $I = \{I_k \in \mathbb{R}^3,\ 1 \leq k \leq N\}$, with associated layer sets $\mathcal{V}_n(B)$ and $\mathcal{V}_n(I)$, respectively, we must associate a label L_k to each pixel I_k, which is equal to 1 if this pixel belongs to the sock and 0 otherwise.

We now detail the background removal procedure. As a preliminary processing, the upper part of the picture, whose background is not uniformly green, is discarded using a fixed mask. Note that though this step seems hard to perform, it is not problematic, since the green background of interest is made of rigid plastic. Therefore, the dimensions of the lower green part are perfectly known, and the mask keeping only the lower part of the pictures can be designed manually on calibration pictures. The two main steps of the algorithm include: a training phase on the background pictures B and the segmentation itself performed on I.

The first training step is summarized in the block diagram in Fig. 4. It aims to learn a relevant model to characterize the background. As mentioned before, most (but not all) of the background pixels have a high Value due to LED lighting and are green, which corresponds to a Hue approximately equal to 0.4. We first classify the pixels of B accordingly to the layer $\mathcal{V}_n(B)$ they belong to. The motivation is to group together the pixels related to the same lighting condition. That way, pixels related to dark or clear tones can be processed differently from the others. Given $\mathcal{V}_n(B)$, the background's contribution to $\mathcal{V}_n(B)$ is modeled by a polygon, namely the convex hull generated by the points of $\mathcal{V}_n(B)$.

$$\mathcal{P}_n = \text{Hull}\big\{\text{diag}(1,1,0) \times B_k, B_k \in \mathcal{V}_n(B)\big\},\ n = 1 \ldots N.$$

The proposed approach has statistical significance. Indeed, it is common to model the background's contribution to the layer $\mathcal{V}_n(B)$ with a bi-dimensional Gaussian mixture density. The convex hull generated by the points can be considered as a sub-optimal approximation of the previous model, more suited to small algorithmic complexity requirements. Furthermore, finding the convex hull in the plane for the points in the layer $\mathcal{V}_n(B)$ can be done using fast algorithms such as the quickhull, with $O(M_n \log_2 M_n)$ complexity, where M_n is the number of pixels related to the layer $\mathcal{V}_n(B)$. Figure 5 represents the 3D shape generated by the polygons $\mathcal{P}_n$, $n = 1 \ldots N$ obtained for the background image from Fig. 2. In this figure, the two blue circles represent the limit of the HSV cylinder.

The second step of the proposed method is the classification of the pixels of I, and is summarized in the block diagram presented in Fig. 6. The underlying idea is that pixels from I, whose HSV decomposition falls outside the polyhedron representing the background (as in Fig. 5), most likely belong to the foot.

We now detail Fig. 6 mathematically. For each layer $\mathcal{V}_n(I)$, we then investigate whether the pixel $I_k = [S_k \cos(H_k); S_k \sin(H_k); V_k]^T$ belongs to the sock or not as follows: if the 2D-point $[S_k \cos(H_k); S_k \sin(H_k)]^T$ has either a low Saturation or lies outside a scaled version of $\mathcal{P}_n$, say $\alpha_n \mathcal{P}_n$ we set the label $L_k = 1$. Otherwise, we set $L_k = 0$. The motivation for this labeling is twofold: if $[S_k \cos(H_k); S_k \sin(H_k)]^T$ is inside $\alpha_n \mathcal{P}_n$, then the pixel most likely belongs to the background. Furthermore, since the

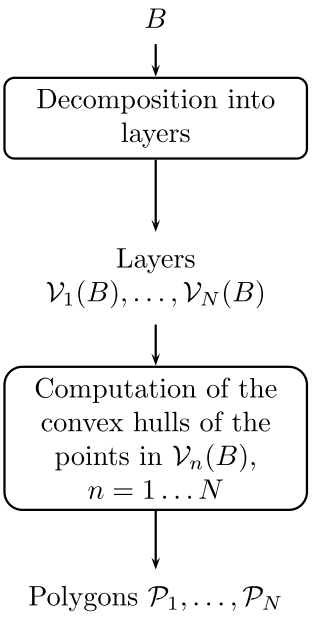


Fig. 4 Block diagram of the training procedure

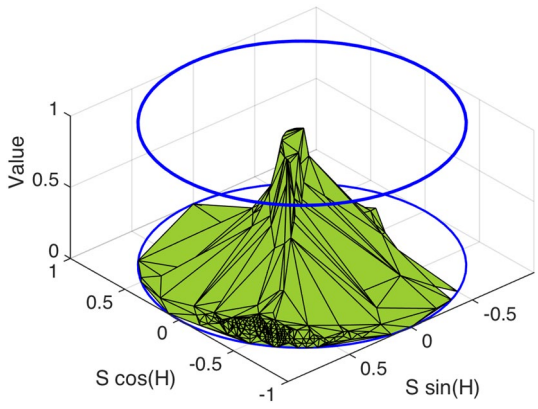


Fig. 5 Example of generated volume for one background image B with $N = 32$ layers

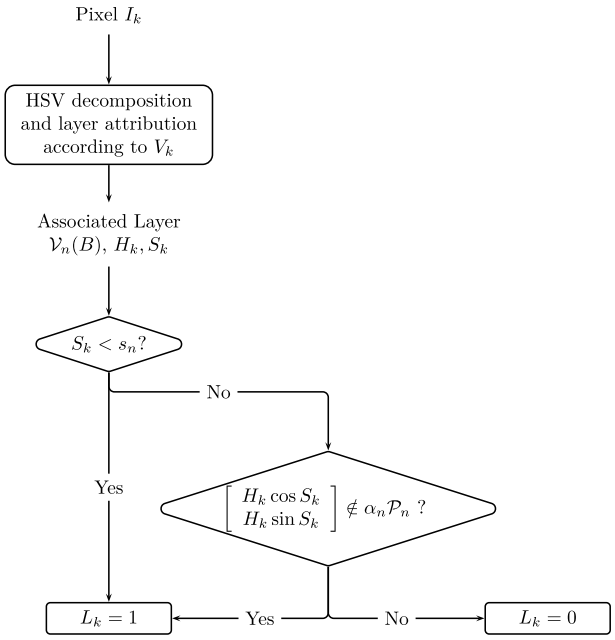


Fig. 6 Block diagram of the segmentation step

background is green and well lit because of the LEDs, any pixel with a low Saturation most likely belongs to a dark sock. We therefore summarize the segmentation as

$$L_k = \begin{cases} 1 & \text{if } \mathrm{diag}(1,1,0) \times I_k \notin \alpha_n \mathcal{P}_n \text{ or } S_k < s_n \\ 0 & \text{otherwise} \end{cases}, \quad \text{for all } I_k \in \mathcal{V}_n(I)\,, \tag{7}$$

where the parameter α_n is a scaling factor depending on the layer considered. It should be emphasized that testing whether a given pixel lies in $\mathcal{P}_n$ can be done quickly in our case, since the polygon to investigate is convex. Therefore, this can be addressed using binary search methods, with logarithmic complexity.

The parameters s_n (defined later on a Saturation threshold) and α_n in (7) aim to attenuate two different kinds of error in the pixel classification. First, it is important to remind that pixels with a low Saturation are the most likely to suffer from the distortion introduced by the JPEG compression, as discussed earlier. However, the background of the Galaxy device is light green with a high Saturation. It is therefore likely that pixels whose Saturation is below the Saturation threshold belong to either a black, gray or white sock. A discussion on a good empirical choice of s_n is detailed in the applications section. The parameter α_n, on the other hand, is necessary to compensate the changes in lighting conditions between background pictures and pictures with a foot. Indeed, since the pictures are taken in near-field conditions, the background in pictures including a foot may slightly differ in terms of Saturation and Value from the model inferred on background pictures. To compensate this discrepancy, we suggest to scale each polygon $\mathcal{P}_n$ based on its layer B_n. In practice, in the applications presented in the paper, the scaling parameter α_n is chosen slightly greater than 1, based on the sequence $\alpha_n = 1.2 + \frac{N-n}{N}$. Such a choice provides good results in practice, and illustrates the fact that we put less confidence in our background model as the Value decreases.

3 Applications

In this section, we present the results with pictures taken from the Galaxy foot scanner. The presented algorithm was implemented in *C♯* using the EMGU computer vision library (a *C♯* wrapper of the OpenCV library), and the execution time for one image on an i7-computer was of the order of magnitude of 100 ms, making it relevant for near real-time implementation.

3.1 Experimental settings

We investigate results obtained on nine types of feet:

- *The Pink and Flower Power datasets* A model of the foot wearing, respectively, a pink sock and a pink sock with green patterns; this can be considered as an easy case, for the sock's Hue is very distinguishable from the light green background;
- *The Funky Black and Black datasets* A model of the foot wearing, respectively, a black sock with gray spots, and a black sock; this kind of sock is hard to isolate from the background with a Hue-based segmentation procedure, due to low Saturation and Value;
- *The White and Pinkie Pie datasets* A model of the foot wearing, respectively, a white sock and a light pink sock with patterns; as in the Funky Black case, this kind of sock is hard to isolate from the background with a Hue-based segmentation procedure;
- *The Light Blue dataset* A model of the foot wearing a clear blue sock; this kind of sock is hard to isolate from the background with a Hue-based segmentation procedure, since this specific tone of blue has a Hue close to the background's.
- *The Duboni and Nuni datasets* Two children's bare feet.

Samples from these nine datasets are displayed in Fig. 7. Recall that one dataset consists of a foot captured from 16 angles by different cameras. For each picture, a mask is

Fig. 7 Samples of the investigated datasets

applied to get rid of the upper part of the picture, and the results obtained with our segmentation procedure are compared to an ideal segmentation performed manually. The error rate is defined as the number of falsely classified pixels divided by the overall number of pixels in the picture.

In our experiments, we investigated the influence of two parameters of interest in our algorithm, that is, the number of layers N and s_n, the Saturation threshold under which a pixel is systematically classified as a foreground pixel. For each dataset, in the first experiment, we perform the segmentation procedure with N ranging from 1 to 256, while s_n is kept constant and equal to 0.2. These extreme values correspond, respectively, to no decomposition into layers, that is, a standard segmentation based on the Hue, and to a layer attributed to each possible grayscale. Regarding the second experiment on the Saturation threshold s_n, we let it vary from 0 (no point is automatically retained as a foreground pixel) to 1 (all the pixels are retained as background), while the number of layers is constant $N = 32$. In both cases, for each dataset, we computed the average segmentation error obtained on the 16 pictures and the associated estimated 90% -confidence interval. By doing so, we aimed to find a good practical choice of the parameters N and s_n, which guaranteed both low segmentation error and the smallest possible variance. Our last experiment presents the results obtained with values of N and s_n set up according to the two previous experiments.

3.2 Results on the choice of the input parameters

Results on the choice of N, the number of layers used in our algorithm, are presented in Fig. 8. A quick examination of the graphs from Fig. 8a–i shows that choosing a N between 20 and 50 provides excellent segmentation results, with an error rate under 5%. For a small number of layers (under 10), we get in all cases the worst results in terms of error rate. After attaining a minimum, the error rate increases regularly for higher number of layers.

Figure 9a–i illustrates the influence of the Saturation threshold s_n. They show that, in practice, choosing $s_n = 0.2$

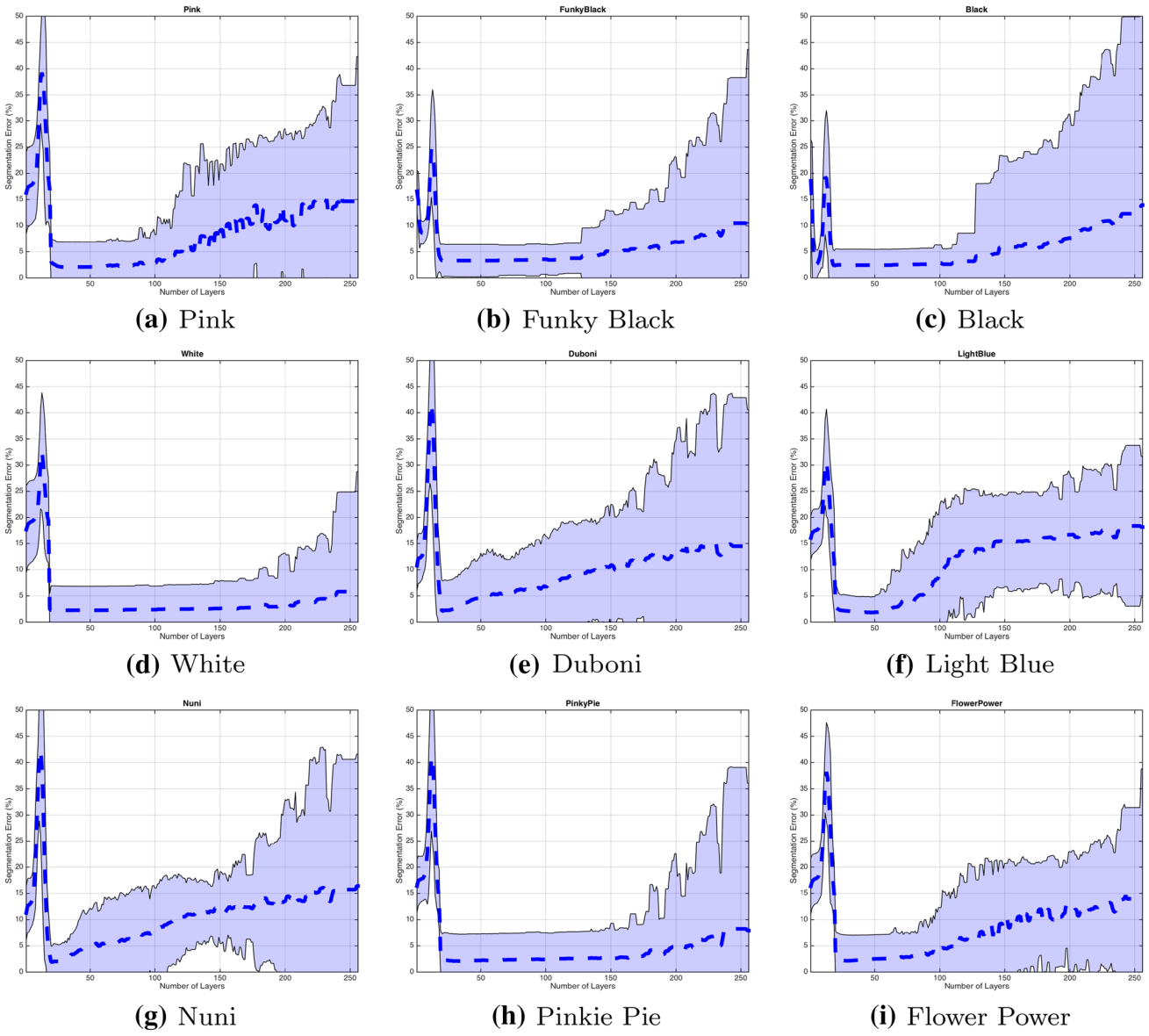


Fig. 8 Influence of the number of layers (x axis) used on the segmentation error rate (y axis)—average result (dotted blue) and associated 90% confidence interval in light blue

can provide very good segmentation results for all the types of socks and bare feet. We notice, in the mean error rate, a change of behavior between color socks and black socks. This is not surprising, however; for black socks, the pixels belonging to the socks have low Saturation as well, so discarding too many pixels with a low Saturation increases the segmentation error rate.

3.3 Segmentation results

Examples of results obtained with the proposed segmentation method are displayed in Fig. 10, with $N = 30$ and $s_n = 0.2$. We chose to present these results without any mathematical morphology post-processing involved. It can be observed that the proposed approach is quite robust to the color of the sock, though the results obtained in favorable cases (pink sock and bare foot) are obviously better. On the whole database, the average segmentation error lies between 2% and 4%, which is quite good and sufficient for the purpose of the Galaxy apparatus.

Among the results, it can be noticed that the proposed algorithm behaves very well for the light blue database, even if the Hue of the sock is close to that of the background. The worst error rate was obtained in that case for Black and FunkyBlack datasets, as appears clearly in Fig. 10b and c. This can be understood easily, since in these cases the most

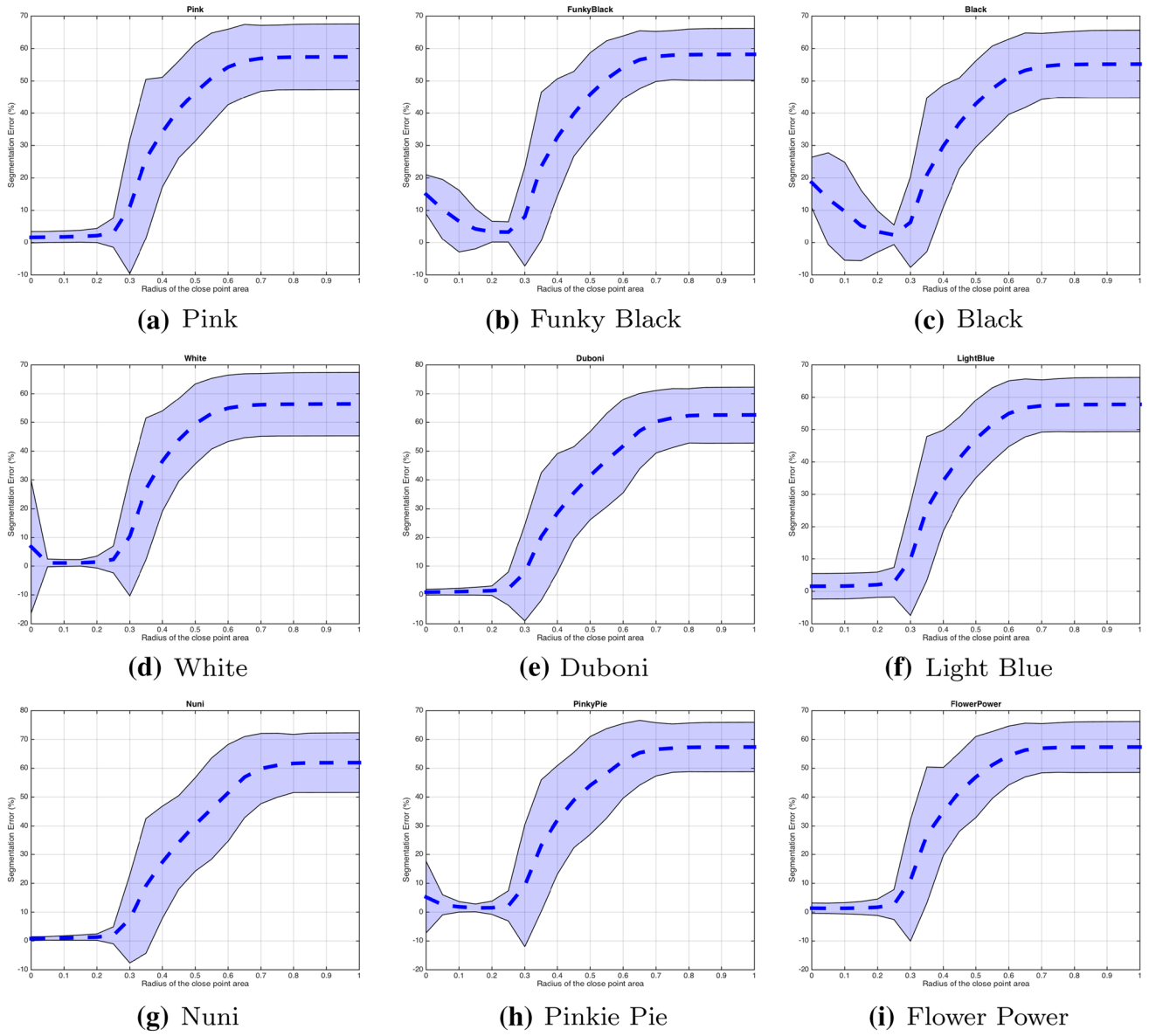


Fig. 9 Influence of the Saturation threshold s_n (x axis) used on the segmentation error rate (y axis)—average result (dotted blue) and associated 90% confidence interval in light blue

critical operation in our algorithm is the comparison to s_n. Obviously, this comparison to a single threshold provides poorer results than a more refined, layer-by-layer analysis of the pictures.

3.4 Discussion

From the two first experiments, we observe that the datasets can be classified into three distinct groups: the first group includes the datasets Pink, FlowerPower and LightBlue, the second includes the White, PinkyPie, FunkyBlack and Black datasets, and the third the Nuni and Duboni datasets. For pictures of the first group, when observing the influence of the number of layers, we see a decrease of performance until 20 layers (this degradation can be explained by the oversmoothing phenomenon, as described later). Above 100 layers, though the average error remains below a 15% threshold, the variability increases. Therefore, the level of confidence we put into the presented method decreases as well. The Saturation threshold s_n, on the other hand, has little influence on the results, provided $s_n < 0.3$. Considering the fact that the socks of the Pink and FlowerPower have mostly a red Hue, and that it is the furthest from the green background's Hue, it is clear that the segmentation performed depends strongly on the Hue parameter, and that a decomposition into layers $\mathcal{V}_n$ can greatly decrease the error rate. The LightBlue

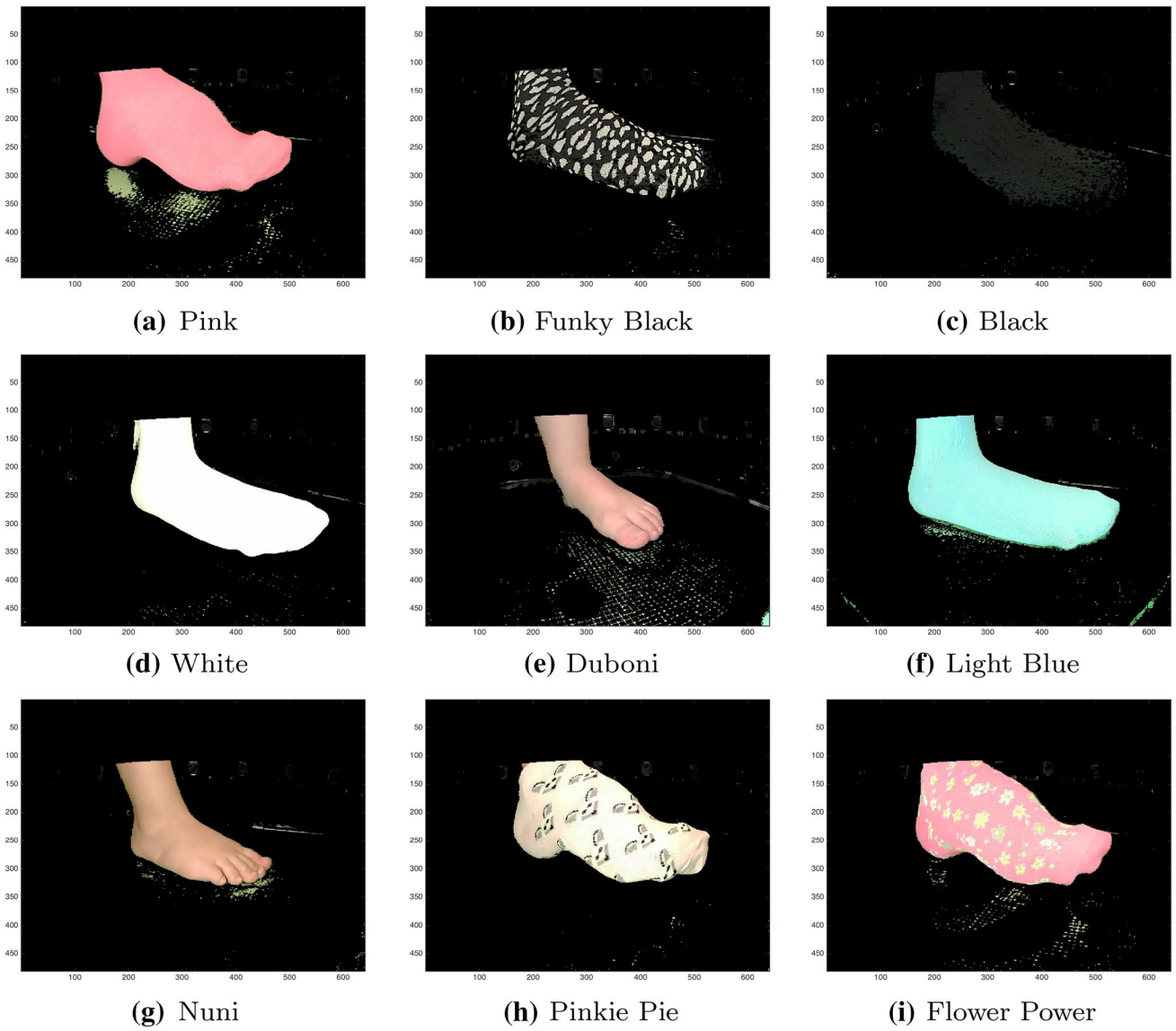


Fig. 10 Results on the sample

dataset behaves similarly, though results have a larger confidence interval due to the similarities between blue and green Hues. For datasets of the second group, the error rate remains below 10% for all of them, even for large values of N. On the other hand, too small a value of s_n yields a larger error rate, since we discard more pixels with low Saturation as understood from Fig. 5. Notice that this decrease of performances is less critical for white socks since in that case the socks' pixels have a high Value V_k. Therefore, they have a larger chance to be identified as foreground pixels than black pixels from dark socks. For this group, we can understand that the Saturation threshold s_n is the main bottleneck of the segmentation procedure.

Finally, the third group illustrates a characteristic of the human skin whose Hue is close to 0. Therefore, this group behaves similarly to the first one, even if the changes of tones of the human skin (when compared to a red uniform sock) introduce more variability in the results, as seen for example from the comparison between Fig. 8e and a. For these measurements, due to the bigger discrepancy of the human skin when compared to socks, the number of layers is a sensible parameter to set, as now detailed.

As aforementioned, the number of layers must be carefully chosen. Splitting the data into few layers increases the error rate of the segmentation procedure. This is because large layers tend to include clusters of both background and sock pixels, and are difficult to separate (this phenomenon is known as oversmoothing). In that case, results have a small variance but a large bias, as shown from the results. On the other hand, an overly large number of layers reduces the

average performances of the segmentation procedure, but at the cost of a large variability in the results. This is mainly due to the fact that each layer contains few pixels to build a background model with any statistical significance. The trade-off to be attained illustrates the common issue of over-fitting (as detailed in the statistical literature, for example in Wasserman [15]). From the results displayed in Fig. 8 obtained on the presented database, it appears that a good numerical rule-of-thumb consists of choosing between 20 and 50 layers to obtain near to optimal performances. Consequently, the number of layers chosen was equal to 20, in practice, for our application. This choice speeds up the execution time while guaranteeing good segmentation results. It shall be noticed that the results presented use numbers a_n (defining the layers $\mathcal{V}_n(I)$) uniformly distributed on the Value scale. We conjecture that the optimal layers subdivision depends on the JPEG compression rate used, and will investigate this aspects in future contributions.

A parameter of interest is also the Saturation threshold s_n, below which a point is systematically chosen as belonging to the sock. From the results obtained in Fig. 9, it can been observed that this parameter is of less importance for colored socks or human skin, while being critical for black, white and gray socks. This observation is not surprising, however, since black and gray tones are related to the smallest Saturations or Values. This happens, when the distortion introduced by the JPEG compression used is the most disturbing. A good value for overall performance is to set uniformly $s_n = 0.25$. We also notice that the proposed approach is relatively steady for colored and bright socks as well as for skin color (as can be seen from the small confidence interval obtained for such pictures in Fig. 9a, d and e). Not surprisingly, its performances naturally decreases for darker socks and gray tones as shown in Fig. 9b.

The final segmentation results presented in Fig. 10 can be improved using standard mathematical morphology operations on the resulting mask, since the small artifacts remaining can be easily discarded this way. From the experiments performed, such post-processing operations (involving a morphological close, finding the biggest element in the mask and applying median filtering to smooth the results) increase the required processing time up to 800 ms for one picture, which remains relevant for our application. We emphasize that these morphological operations must be performed with caution, since they can combine background pixels and foreground pixels altogether. A more uniform background on the device, or the use of more refined algorithms, may improve the obtained results, and will be investigated in future contributions.

4 Conclusion

In this paper, we have presented a generic algorithm for uniform background removal, which is independent of the color of the object of interest, and takes into account the color distortion inherent to the JPEG compression. When applied to our specific application, a foot scanning device, we observed that the obtained performances are quite good, even for foreground objects considered as difficult, with approximately a 2–4% error rate. Further work in that direction will include background removal from the upper part of the image, extraction of features of interest from the foreground objects and full 3D reconstruction of the foot given the segmented images. Details on these aspects will appear in future contributions.

References

1. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: Proceedings of the 8th IEEE international conference on computer vision (ICCV). IEEE, Vancouver, 7-14 July 2001
2. Chen, J.J., Su, C.R., Grimson, W.E.G., Liu, J.L., Shiue, D.H.: Object segmentation of database images by dual multiscale morphological reconstructions and retrieval applications. IEEE Trans. Image Process. **21**, 828–843 (2012). https://doi.org/10.1109/TIP.2011.2166558
3. Faro, A., Giordano, D., Spampinato, C., Ullo, S., Di Stefano, A.: Basal ganglia activity measurement by automatic 3-D striatum segmentation in SPECT images. IEEE Trans. Instr. Measure. **60**(10), 3269–3280 (2011)
4. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 4th edn. Pearson, New York (2017)
5. Haines, T.S.F., Xiang, T.: Background Subtraction with Dirichlet Processes, pp. 99–113. Springer, Berlin Heidelberg (2012)
6. Li, C.T.: Multiresolution image segmentation integrating gibbs sampler and region merging algorithm. Signal Process. **83**(1), 67–78 (2003)
7. Li, T.H.S., Wang, Y.H., Chen, C.C., Lin, C.J.: A fast color information setup using EP-like PSO for manipulator grasping color objects. IEEE Trans. Indus. Inform. **10**(1), 645–654 (2014)
8. Liu, Y., Payeur, P.: Robust Image-based detection of activity for traffic control. Can. J. Electr. Comput. Eng. **28**(2), 63–67 (2003)
9. Pears, N., Liu, Y., Bunting, P.: 3D Imaging, Analysis and Applications. Springer, New York (2012)
10. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph **23**(3), 309–314 (2004a)
11. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In: ACM SIGGRAPH 2004 Papers, ACM, New York, NY, USA, SIGGRAPH '04, pp 309–314 (2004b)
12. Sigal, L., Sclaroff, S., Athitsos, V.: Skin color-based video segmentation under time-varying illumination. IEEE Trans. Pattern Anal. Mach. Intell. **26**(7), 862–877 (2004)
13. Silva, A.S., Quintao Severgnini, F.M., Oliveira, M.L., Santiago Mendes, V.M., Assis Peixoto, Z.M.: Object tracking by color and active contour models segmentation. IEEE Latin Am. Trans. **14**, 1488–1493 (2016). https://doi.org/10.1109/TLA.2016.7459639

14. Telfer, S., Gibson, K.S., Hennessy, K., Steultjens, M.P., Woodburn, J.: Computer-aided design of customized foot orthoses: reproducibility and effect of method used to obtain foot shape. Arch. Phys. Med. Rehab. **93**(5), 863–870 (2012)
15. Wasserman, L.: All of Nonparametric Statistics (Springer Texts in Statistics). Springer, New York (2006)
16. Wei, K., Jing, Z.L., Li, Y.X., Tuo, H.Y.: Extended scheme of chan-vese models for color image segmentation. IET Image Process. **5**, 583–597 (2011). https://doi.org/10.1049/iet-ipr.2009.0387
17. Yoon, I., Kim, S., Kim, D., Hayes, M.H., Paik, J.: Adaptive defogging with color correction in the HSV color space for consumer surveillance system. IEEE Trans. Consum. Electr. **58**, 111–116 (2012). https://doi.org/10.1109/TCE.2012.6170062

Thomas Trigano was born in Paris, France, in 1978. He received the M.Sc. degree in engineering from the Télécom Paris Tech, Paris, France, and the M.Sc. degree in applied probability from Paris VI University, Paris, France, in 2001. He received the Ph.D. degree in signal processing from the Télécom Paris Tech in 2005. From 2006 to 2008, he received a Postdoctoral Fellowship from the Department of Statistics, Hebrew University of Jerusalem. Since 2008, he has been a Senior Lecturer in the Department of Electrical Engineering, Shamoon College of Engineering, Ashdod, Israel. His main research interests include applied statistics, statistical signal processing, and pattern recognition and communications.

Yuval Bechor is Aetrex's Software Development Manager. With over 30 years of combined hardware and software development experience, Yuval has been intricately involved in procuring several unique patents related to foot sole measuring. His research interests include image processing and computer vision applications to orthotics. His current work focus on iStep software, scanner interface and new products development.