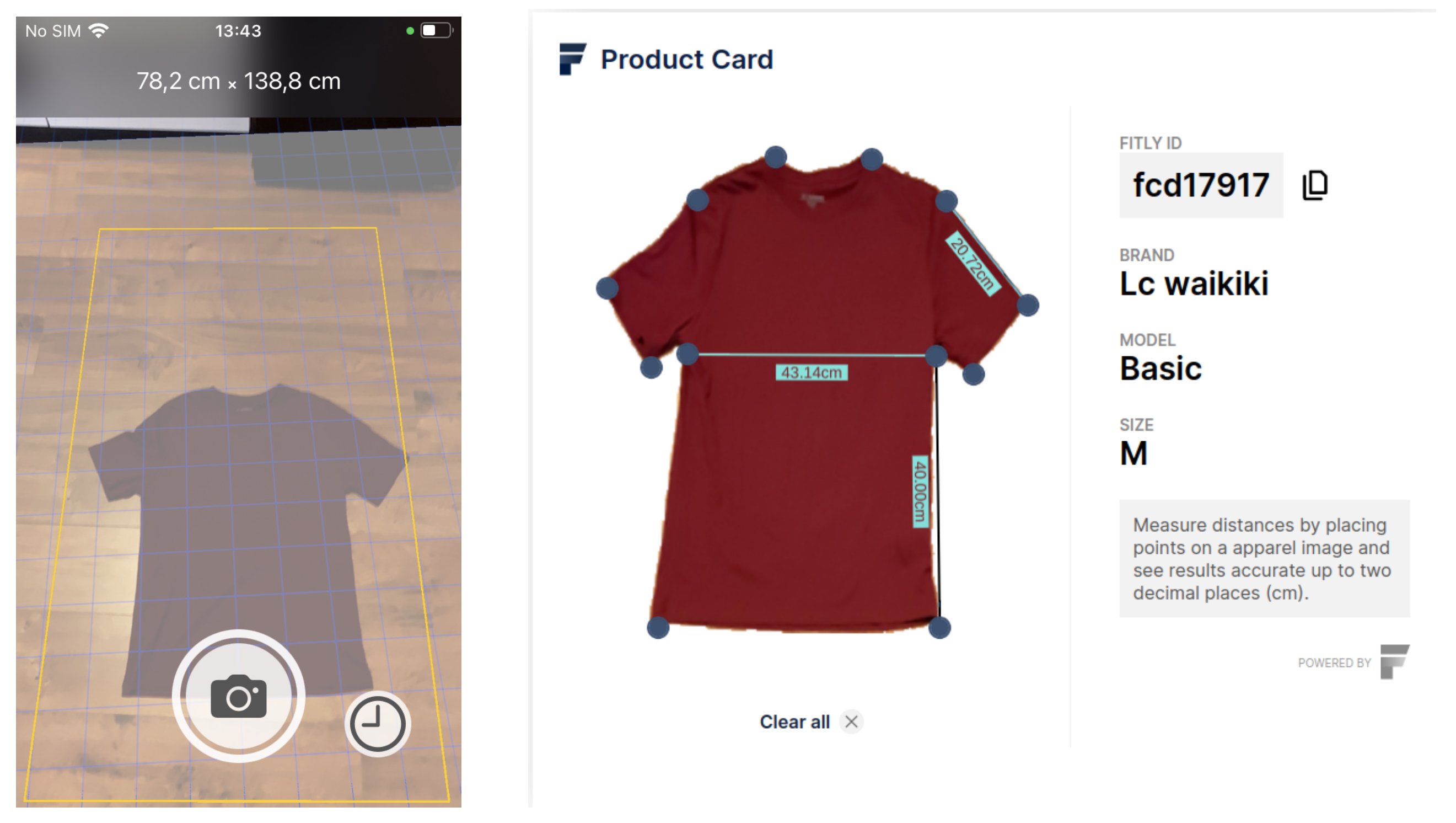
Finding perfectly fitting clothes

Konrad Czarnota, Magdalena Kalbarczyk, Jakub Cieślik
konrad@fitly.ai, magda@fitly.ai, kuba@fitly.ai
Fitly.ai

Introduction

This poster introduces a novel approach to finding the best fitting clothes based on images. The problem is widely known for online fashion shoppers and e-commerce retailers. The way fashion is designed and made production-ready is inconsistent with the way manufactures execute production of clothes and there's no standard for how specific sizes should differ and what physical dimensions they should be coherent with. Because of that, shoppers experience uncertainty, they are often disappointed with their purchases and sellers are exposed to copious revenue inefficiencies and high return handling costs.

The proposed solution uses a mobile application, which takes photos of apparel lying on flat surfaces. Computer vision engine processes the input, segments the object and detects points of interests. This information is later used to help the user make a better choice and is expected to improve the overall quality of purchase decisions.



FITLY ID
fcd17917
BRAND
Lc waikiki
MODEL
Basic
SIZE
M

Measure distances by placing points on a apparel image and see results accurate up to two decimal places (cm).

Figure 1: Left side: app screenshot while taking photo. Right side: an example of apparel with all collected information.

Data

To fully solve the problem, images should contain the same clothes in different sizes. Those should be additionally measured and annotated with segmentation masks and key points' locations. There is a lack of datasets with all this information, so the authors have decided to create one themselves. Acquisition of images and their measurements in centimetres was achieved by using ARKit [1] technology in the mobile app. Collected dataset consists of 62 lower body clothes (pants and shorts) and 165 upper body clothes (t-shirts, shirts, long-sleeves, blouses). Each manual annotation contains a category, mask and a list of coordinates for points of interest.

Segmentation

Due to the relatively small dataset and assumptions about clothes (one object per image with a relatively plain background) made authors decided to first test computer vision methods not requiring data to train, postponing deep learning approaches for later, when more data would be available.

Classical computer vision approaches to segmentation can be divided into two groups: first, work only with pixels, second, require additional initialisation.

Algorithms like edge detection with contour filling or different thresholdings take an image and produce a mask. Those are very sensitive for any noise and produce acceptable results only when the object is clearly visible and easily distinguishable from the background. However, they can be used as a starting point for more sophisticated methods.

Algorithms requiring initialisation are definitely more robust, but need that additional input to make them work. This might be done manually (i.e. by providing a bounding box selected by a user) or automatically. Those hints don't have to be perfect, just more or less pointing to the probable location of the object.

During tests, no single method without initialisation worked well enough. Algorithms requiring hints were performing better, but the results depended greatly on hints used. In easy cases, some random initialisation with checkerboard or ellipse patterns were enough. However, they were not able to cope with a cluttered background. In these cases more sophisticated approaches as Otsu's thresholding [4] or structured edge detections [3] proved to behave better.

Final results presented in Table 1 show the best pairs and their performance, divided by a category.

Method	Upper body mIoU	Lower body mIoU
Grabcut [2] with structured edge detection [3]	0.96	0.97
ACWE [5] with random checkerboard	0.86	0.68
Watershed [6] with otsu thresholding [4]	0.52	0.77

Table 1: Segmentation results

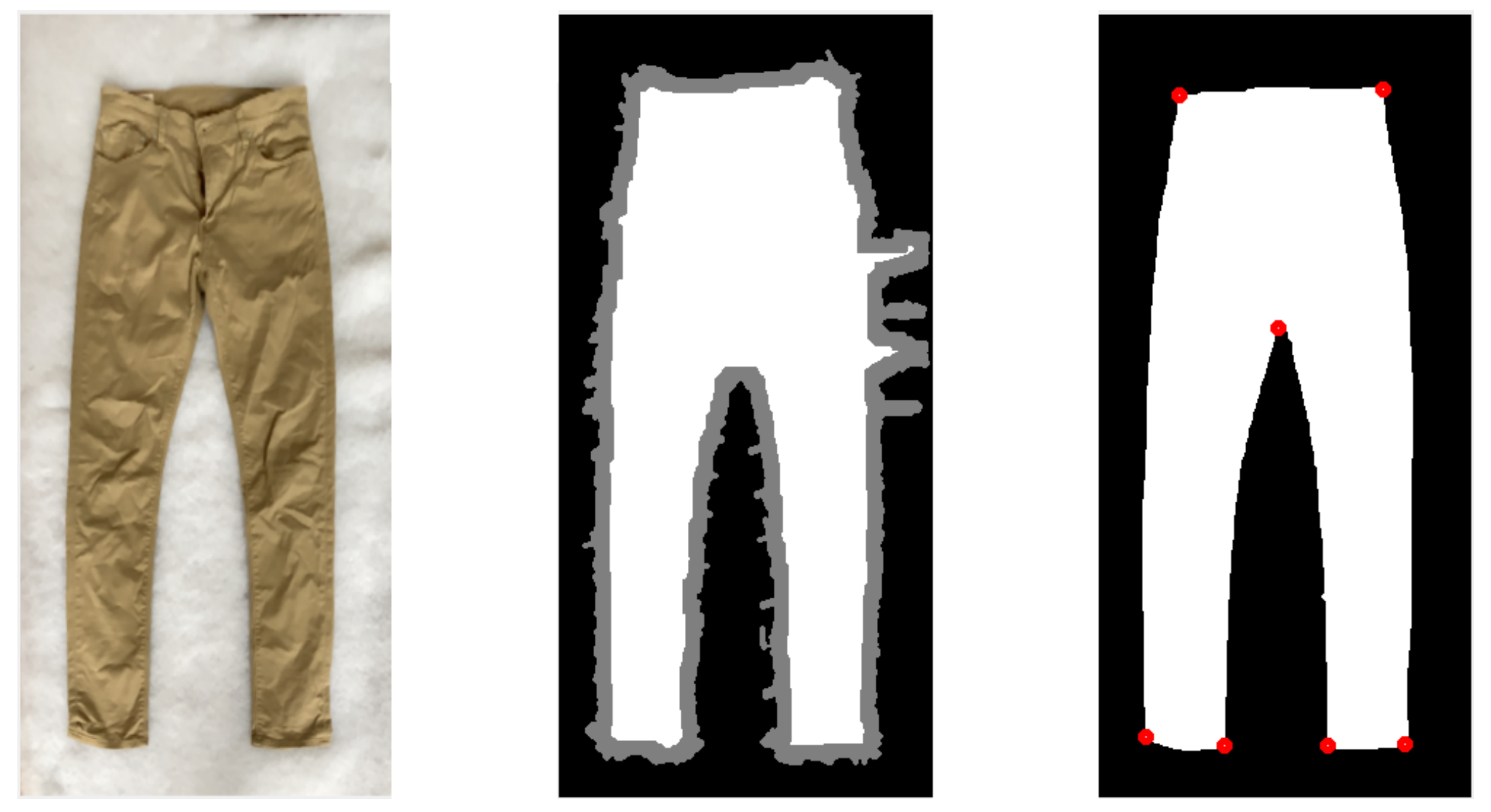


Figure 2: Left: original image. Middle: initialisation mask obtained with edge detection, contour filling and erosion (black means background, gray probable background, white probable forground). Right: final segmentation with Grabcut and keypoints.

Keypoints detection

Detecting the location of characteristic points on clothes is necessary for automatic measurements. These can later be used as one of the inputs to the comparison module. The proposed approach is to first obtain a contour around the whole mask. In the next step, the contour is simplified and the least informative points are omitted. Moving further, the most outlying points can be found and work as anchors for further selections. Using a divide and conquer approach, all necessary points can be found with low mean squared error in pixels. Example for pants: find top left, top right, bottom right and bottom left points - those are the points from the contour closest to four edges of the image. Next, get the points between bottom left and bottom right. The one which is the highest of those is a crotch. Use it to divide to the right and left leg. Now find inner points of legs by picking the lowest points in corresponding groups.

Upper body MSE	Lower body MSE
5.04	2.88

Table 2: Keypoints detection results

Comparing sizes

Comparing different clothes sizes is crucial to help the user make a good decision. There are two possible cases:

- there is a shop with different sizes available - the user should get the closest match for the item presented to him,
- there is only one size available(e.g. someone is selling something that is too large for him) - user should get information about how it differs from his item.

There are numerous approaches for object comparison, usually based on geometric moments. In this described case they don't work, because of invariance to size changes. This proposed solution uses the area of masks and measurements taken between selected pairs of corresponding points. This way the shape (e.g. baggy or slim fit) is taken into consideration as well. It was tested on different types of clothing from the same set in a way that the distance between closer sizes should be smaller then the distance for those further apart (e.g. difference between S and M should be smaller then between S and XL). The proposed algorithm was able to obtain **100%** accuracy.

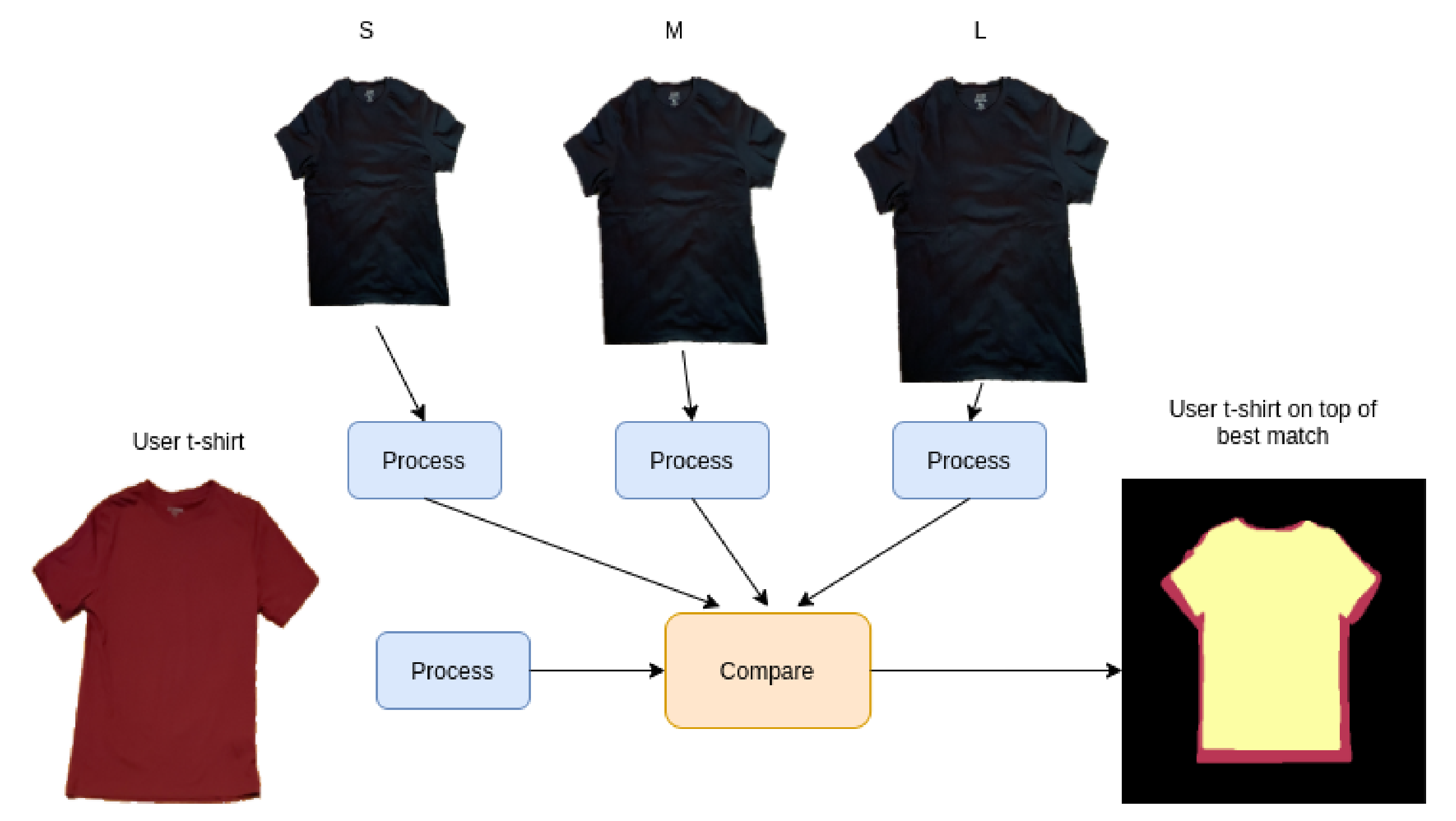


Figure 3: The process of finding best fitting clothes. Images are processed to obtain their masks and measurements. Those are passed to comparing module. It returns the closest match and provides the user with an image outlying his item on top of the matched one.

Summary

This paper describes the whole process of finding the best fitting clothes based on images. It uses only classical computer vision methods, which doesn't require data to train. Those are still useful and worth knowing even in the era of deep learning. They might be used as final solutions in more constrained use cases or as a way of automatic ground truth generation in more general problems.

References

[1] `https://developer.apple.com/documentation/arkit`. [Online; accessed 08.12.2020].

[2] V. Kolmogorov C. Rother and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph*, 2004.

[3] Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. *IEEE International Conference*, 2013.

[4] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979.

[5] Luis Álvarez Pablo Márquez-Neila, Luis Baumela. A morphological approach to curvature-based evolution of curves and surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[6] Peer Neubert & Peter Protzel. Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. *ICPR*, 2014.