AR Contents Superimposition on Walls and Persons

João M. F. Rodrigues[1(✉)], Ricardo J. M. Veiga[1], Roman Bajireanu[1], Roberto Lam[1], Pedro J. S. Cardoso[1], and Paulo Bica[2]

[1] LARSyS (ISR-Lisbon) and ISE, University of the Algarve, 8005-139 Faro, Portugal
{jrodrig,rjveiga,rlam,pcardoso}@ualg.pt, romanmsn.com@hotmail.com
[2] SPIC - Creative Solutions, Faro, Portugal

Abstract. When it comes to visitors' experiences at museums and heritage attractions, objects speak for themselves. With the aim of enhancing a traditional museum visit, a mobile Augmented Reality (AR) framework was developed during the M5SAR project. This paper presents two modules, the wall and human shape segmentation with AR content superimposition. The first, wall segmentation, is achieved by using a BRISK descriptor and geometric information, having the wall delimited, and the AR contents superposed over the detected wall contours. The second module, person segmentation, is achieved by using an OpenPose model, which computes the body joints. These joints are then combined with volumes to achieve AR clothes content superimposition. This paper shows the usage of both methods in a real museum environment.

Keywords: Augmented Reality · Wall detection · Human detection · Wall overlapping · Clothes overlapping · HCI

1 Introduction

Augmented Reality (AR) [3] is no longer an emergent technology, thanks mainly to the mobile devices increasing hardware capabilities and new algorithms. As cornerstone, AR empowers a higher level of interaction between the user and real world objects, extending the experience on how the user sees and feels those objects, by creating a new level of edutainment that was not available before. While many mobile applications (App) already regard museums [31,56], the use of AR in those spaces is much less common, albeit not new, see e.g. [21,46,49,58].

The Mobile Image Recognition based Augmented Reality (MIRAR) framework [45] (developed under M5SAR[1] project [49]) focuses on the development of mobile multi-platform AR systems. One of the MIRAR's requirements is to only use the mobile devices RGB cameras to achieve its goals. A framework that

[1] Mobile Five Senses Augmented Reality System for Museums, financed by CRESC ALGARVE2020, PORTUGAL2020 and FEDER.

M. Antona and C. Stephanidis (Eds.): HCII 2019, LNCS 11572, pp. 628–645, 2019.
https://doi.org/10.1007/978-3-030-23560-4_46

integrates our presented goals is completely different from the existing AR software development kits – SDK, frameworks, content management systems, etc. [2,11,37].

This paper focuses on two particular modules of MIRAR, namely: (a) the recognition of walls, and (b) the segmentation of human shapes. While the first module intends to project AR contents onto the walls (e.g., to project text or media), the second contemplates the overlap of clothes onto persons. The wall detection and recognition is supported upon the same principles of the object's recognition (BRISK descriptor) but uses images from the environment to achieve it. On the other hand, the human detection and segmentation uses Convolutional Neural Networks (CNN) for the detection (namely, the OpenPose model [9]). The overlapping of contents in the museum environment is done over the area limited by the wall or using the body joints along with clothes volumes to put contents over the persons. The main contribution of this paper is the integration of AR contents in walls and persons in real environments.

The paper is structured as follows. The contextualization and a brief state of the art is presented in Sect. 2, followed by the wall segmentation and content overlapping sub-module in Sect. 3, and the human shape segmentation and content overlapping in Sect. 4. The paper concludes with a final discussion and future work, Sect. 5.

2 Contextualization and State of the Art

AR image-based markers [12] allow adding in any environment easily detectable pre-set signals (e.g. paintings and statues), and then use computer vision techniques to sense them. In the AR context, there are some image-based commercial and open source SDK and content management systems, such as Catchoom [11], ARtoolKit [2] or Layar [37]. Each of the above solutions has pros and cons and, to the best of your knowledge, none has implemented wall and person segmentation with information overlapping.

The ability of segmenting the planar surfaces of any environment continues to be a challenge in computer vision, mainly if only a monocular camera is used. One of the directest approach to an environment's scanning is the use of RGB-D cameras [25] or LiDaR devices [30] to directly acquire a 3D scan of the cameras' reach. A more indirect approach – more based on computation than hardware – is the Simultaneous Localization and Asynchronous Mapping (SLAM) [13]. SLAM's methods for indoor and outdoor navigation has shown new advances either by using the Direct Sparse Odometry [15], or with a feature matching method like the ORB SLAM [42] or even a Semi-Dense [17] or Large-Scale Direct Monocular SLAM [16].

Another usual approach is the cloud of points method or the structure from motion, which is part of the SLAM's universe, relying on multiple frames to be able to calculate a relation in-between the features – 3D points – and the camera's position. There have been developments in the outdoor, or landmark, recognition [4], an also simple objects detection and its layout prediction using

the cloud of oriented gradients [48]. Another example, proving the possibilities of a proper environment's layout analysis, is the use of a structure from motion algorithm using the natural straight lines in an environment, through representation, triangulation and bundle adjustment [6].

One of the main novelties is the use of CNN to solve any complex computer vision challenge, including environment's layout prediction [54], although the current state is not useful in runtime. On the other hand, in every common human-based construction there can be found the presence of lines or edges in its geometric perspective. These vanishing lines allows us to predict the orientation and position of planes [26]. It is even possible to compute a relative pose estimation using the present lines in the environment [14]. These techniques, applied to the indoor layouts' prediction, allows us to compute the existence of natural planar surfaces [51], even by using the edges of maps available on any indoor layout [39]. One major advance in the outdoor camera localization is the PoseNet [33], which also uses a CNN. It is important to stress that none of those methods presents the superimposing of contents over an environment know *a priori*, on a monocular mobile device and in runtime.

The second module to be presented in this work focuses on human segmentation and pose estimation, which is also a challenging problem due to several factors, such as body parts occlusions, different viewpoints, or human motion [20]. In the majority of models based on monocular cameras, the estimation of occluded limbs is not reliable. Nevertheless, good results for a single person's pose estimation can be achieved [20]. Conversely, pose estimation for multiple people is a more difficult task because humans occlude and interact between them. To deal with this task, two types of approaches are commonly used: (a) top-down approach [27], where a human detector is used to find each person and then running the pose estimation on every detection. However, top-down approach does not work if the detector fails to detect a person, or if a limb from other people appears in a single person's bounding box. Moreover, the runtime needed for these approaches is affected by the number of people in the image, i.e., more people means greater computational cost. (b) The bottom-up approach [10,20] estimates human poses individually using pixel information. The bottom-up approach can solve both problems cited above: the information from the entire picture can distinguish between the people's body parts, and the efficiency is maintained even as the number of persons in the image increases.

As in the wall detection, the best results for pose estimation are achieved using R-CNN (Regions - CNN) [23] or evolutions, such as the Fast R-CNN [22], Faster R-CNN [47] or the Single Shot MultiBox Detector (SSD) [29]. A comparison between those methods can be found in [29]. The results show that SSD has the highest mAP (mean average precision) and speed. With good results, OpenPose [10] can also be used for pose estimation, being based on Part Affinity Fields (PAFs) and confidence maps (or heatmaps). The method's overall process can be divided in two steps: estimate the body parts (ankles, shoulders, etc.) and connect body parts to form limbs that result in a pose. In more detail, the method takes an input image and then it simultaneously infers heatmaps and

PAFs. Next, a bipartite matching algorithm is used to associate body parts and, at last, the body parts are grouped to form poses. The OpenPose can be used with a monocular camera and run in "real-time" on mobile devices. Additionally, the estimated 2D poses can be used to predict 3D poses using a "lifting" system, that does not need additional cameras [55].

Several methods exist for clothes overlapping. A popular one is Virtual Fitting Room (VFR) [18], which combines AR technologies with depth and colour data in order to provide strong body recognition functionality and effectively address the clothes overlapping process. Most of these VFR applications overlap 3D models or pictures of a clothing within the live video feed and then track the movements of the user. In the past, markers were used to capture the person [1]. In that case, specific joints are used to place the markers, which differ in colours according to the actual position on the body. From a consumer's point of view, a general disadvantage is the time consumed placing the markers and the discomfort of using them. Isikdogan and Kara [32] use the distance between the Kinect sensor and the user to scale a 2D model over the detected person, only depicting the treatment of t-shirts. Another similar approach, presented in [18], uses 3D clothing with skeleton animation. Two examples of the several commercial applications are FaceCake [19] and Fitnect [34].

3 Wall Detection and Information Overlapping

Previously, the authors followed two distinct approaches to solve the environments' surfaces detection [45,50,59,60]. A first approach assumes that the vanishing lines present in the environment follow an expected geometric shape; and a second approach focuses on retrieving the walls' proportions using the features extraction and matching method, followed by the homographies' computation. The methods were then combined in order to achieve a harmonious detection, recognition and localization of the environment, allowing to dynamically superimpose different types of content over the walls, such as images, video, animations, or 3D objects.

As detailed next, the present algorithm is designed to work over regular plane walls, which are known *a priori* through a previously bundle creation phase. Being the purpose of this AR application the ability to run seamlessly on any current monocular smartphones, from which only a RGB image is provided by the camera (i.e., without any additional depth information), it is important to assure an ideal performance using less computational' eager algorithms.

Our current algorithm divides itself in five different stages: (a) the bundle creation, (b) the recognition and localization, (c) corners' adjustment, (d) tracking, and (e) superimposition.

The first stage of the algorithm – the bundle creation (a) – is pre-executed, i.e., not performed during runtime. For this task two distinct types of bundles are generated: a FLANN (Fast Library for Approximate Nearest Neighbours) [41] Index (FI) bundle, and a FLANN Based Matcher (FBM) bundle. This odd combination is due to a better performance being obtained by a hybrid version

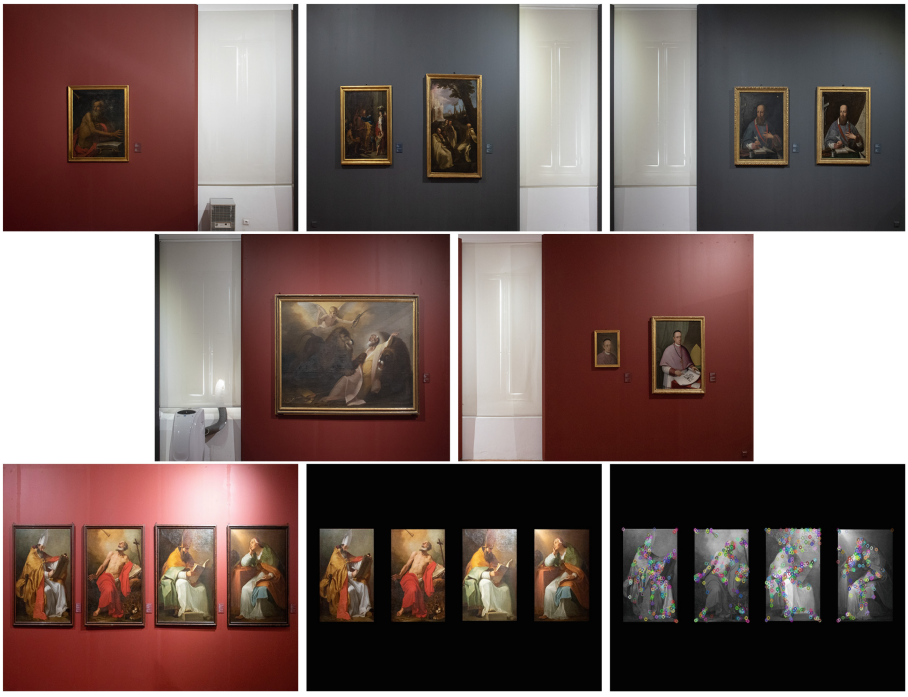

Fig. 1. Top two rows, from top left to bottom right: Example of five templates following of the same wall. Bottom row, pre-processing of the templates. Left to right, input image with the complete desired height of the wall, mask applied over removing the wooden frames, features retrieved and computed.

of both FLANN matchers instead of only one, as presented in [60]. Reasons for this choices will be better detailed during the recognition and localization (b) phase.

Museums' environments are full of detail and some of its areas gather enough significant information to be considered keypoints, which can be detected and define by computing its descriptors. In this approach, the BRISK keypoint detector and descriptor extractor [38] is used, due to its capabilities of performing well with image scaling. Images of continuous walls, as can be seen in Fig. 1 top two rows, allow not only to project content, but also retrieve the users' localization through the sparse unique keypoints inside the artworks. The retrieved features are stored during the bundle creation, allowing the comparison during runtime with the ones obtained from the smartphones' cameras.

As observed in [60], the paintings' wooden frames are rich in similar features, which often would lead to cross-matched in between them. To prevent this false matches, the templates are pre-processed before training the FLANN indexes, defining masks where only the features from the artworks could be obtained, as it can be seen on Fig. 1 bottom row. Additional final templates examples can be

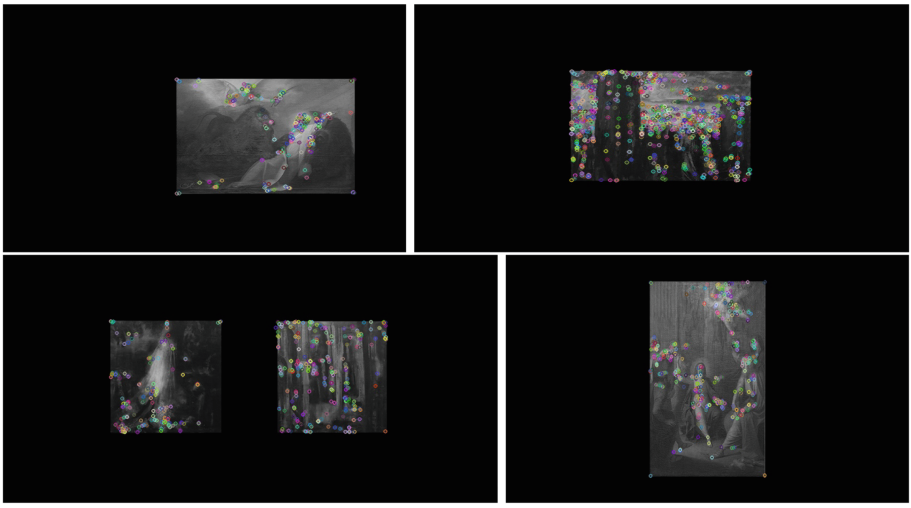

Fig. 2. Example of some of the templates used during the bundle creation stage.

observe on Fig. 2. The motive behind the shape and form of the templates will be explained in detail during the next phases.

Although FBM is built upon FI, previous performance tests showed that the bare FI returns results similar to the ones obtained with FBM, but with an average of 60.66% less processing time [60], which justifies the choice of building an FI bundle. While both methods retrieve the same template index, the FBM also retrieves the matching between features, which is essential for the computation of the homography. Following this necessity, a bundle is created for each matching method, which allows to generate a hybrid FLANN matching method. This method, starts by searching across our templates with the FI bundle and then only process the top retrieved results with FBM, which was proved to be a faster matching method, when compared to using exclusively FBM [60].

Both methods – FI and FBM – used the same index parameters, and the same searching algorithm, the Locality-Sensitive Hashing (LSH), which performs extremely well with non-patent binary descriptors. The LSH used a single hash table with a key size of 12, and only 1 multiprobe level. The addition of a multiprobe to the LSH, allows to reduce the number of hash tables, obtaining a better computational performance without affecting precision. As presented in [60], it was noted an average reduction of 76.56% of processing time across different binary features detectors and descriptors (AKAZE, BRISK, ORB) [53] while using only 1 hash table, versus the 6 hash tables originally recommended.

The runtime computation starts with the recognition and localization stage (b). While no localization information or previous match is available, the retrieved frame from the camera is resized to a resolution of 640×480 pixels (px), and processed with the BRISK feature detector through the FI feature matcher, returning a list of probabilities for the index of each template, as can

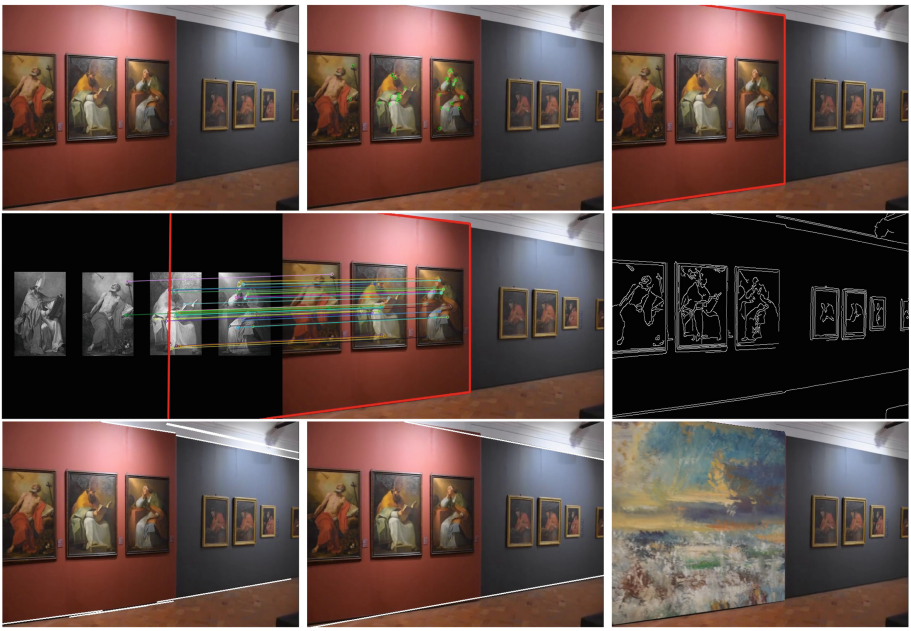

Fig. 3. Pipeline of the environments' superimposition algorithm. Left to right, top to bottom: input frame, keypoints and descriptor computation, homography's calculation, demonstration of the relation between matches and the homography, Canny edge detection, Probabilistic Hough Transform, vanishing lines post-processing, content superimposed.

be seen on the top-left and top-centre of Fig. 3. Similar to the top-5 rank in CNN, the image with highest probability is not occasionally matched, although one within the top-5 is used. Then the FBM is applied through the top-5 indexes and the results are subjected to the Lowe's ratio test, where only the matches with distances to each other with a relation between 55% and 80% are considered. If at least 20 of these matches are obtained, then the algorithm continues, otherwise it skips this frame's processing. It is also important to stress that in order to achieve a near real-time performance, the previous frame's processing time is correlated with the total amount of descriptors for the current frame, with all being firstly sorted by their response parameter, which correlates the level of similarity between the templates and frames' descriptors.

With the computation of the homography's matrix between the correlated matches of the template and the camera's frame, the perspective transformation of the 2D template can be computed as an object within the 3D world, which can be observe in Fig. 3 – top-right. Normally, the homography only requires 4 matches to be able to calculate but, with the user navigating through out the museum, steady results for this AR application were obtained only when the minimum limit of matches was increased to 20 points. We also discard the

bad homographies verifying if the computed matrix presents a possible solution which could match our desired output: direction, proportion, and perspective. A demonstration of this process can be seen in Fig. 3 – centre-left and right.

During the bundle creation stage (a) the templates' shape form where made for a specific purpose: the ability to find the upper and bottom margins of any specific wall, as well the left and right limits when necessary. The current arrangement of templates is divided between two rooms, one regular – cuboid – and one irregular. The aim for the regular room is to be able to localize the exact position and angle that the user is pointing. Furthermore, using the continuous templates from the same wall, as shown in Fig. 1 – top two rows, an automated mixed 3D layout of the museum's room is being designed, with the objetive of further exploring the AR applications without the need for advanced 3D calculations. In the irregular room, the walls are used to project any desired content, e.g., a video-documentary related to the artwork exposed on that specific wall without the ability to project over the entire environment's layout.

With the homography already known, the next step is the corner's adjustment stage (c), which is the result of the combination of several methods [45,50,59,60]. The frame's edges are computed by applying a Gaussian filter to blur the frame, followed by a dynamic Canny edge detection [8] using the Otsu threshold [43] to replace the high Canny's threshold, which decides if a pixel is accepted as an edge, while the lower threshold, which decides if a pixel is rejected, varies with a direct proportion of 10% to the higher. The computed edges can be seen on the centre-right of Fig. 3. Afterwards, the Probabilistic Hough Transform [36] is applied in order to retrieve the lines present in the frame, as seen in the bottom-left of Fig. 3.

Next, the obtained lines are filtered by discarding the extremely uneven lines in relation to the horizon line, followed by the calculation of the similar ones, resulting only in the expected environment's vanishing lines. The lines' intersecting points were clustered using a K-means clustering method, where the densest cluster is chosen, and its centroid is considered as the vanishing point of said lines. Considering the original location of the homography's corner points, with the known vanishing point, these corners can be adjusted to existing lines in the environment – upper and lower limit of the wall –, as observed in the bottom-centre of the Fig. 3.

Previously, the application of Kalman filters to the vanishing point and its corresponding corner's coordinates was introduced in [60], allowing for a better perception of the user's movement, and consequently smoothing the transitions of the superimposed content. Although the current state of the present algorithm retains this step, Kalman filters are no longer used for tracking, with its main function being the validation of a proper template's perspective found on the processed frames. More precisely, the Kalman filtering of the coordinates allows to predict their next position and estimate if the ones retrieved behaved as noise or valid inputs. This favors the obtention of more precise coordinates in time with more harmonious trajectories – it is important to refer that the obtained homographies are not perfect and their perspective fluctuates significantly, which

leads to noisy coordinates. This probably is due to the recursiveness of the Kalman filters but, there was only the need to adjust the uncertainty matrix to our specific application and no additional past information is required to be able to process in real time. Before advancing to the last two stages of the algorithm, the previous steps are computed again using a mask retrieved from the calculated coordinates. When the Kalman filters stabilizes, the process proceeds to the next stage.

Regarding the tracking stage (d), with the corresponding template's coordinates found, the good features to track within our current frame's mask are computed, using the Shi-Tomasi method [52]. Afterwards, the optical flow between the previous and the current frame is calculated using the iterative Lucas-Kanade method with pyramids [7]. Using this method, a more accurate homography between frames can be computed, which results in a more fluid and smooth tracking using even less computation than our previous approach. It should be noticed some important aspect of this approach such as the fact that the smartphones' cameras are different between brands and models, sometimes even between the operating system versions, which results in different features match across the devices. Through this method, a lighter computational tracking in any device and in multiple conditions was possible. The Shi-Tomasi corners continues to be obtained through the tracking, which enables the visitor to continue walking through the museum without the AR experience – which enables the visitor to explore the content in a higher detail.

Following the previous stage, the superimposition stage (e) can finally processed. With the improved tracking stage, the overlay of content over the environments' previous known walls, allowing the visitors' movement, is possible, without affecting the projected content. The result can be seen in the bottom-right of Fig. 3. Although, it is only presented the projection of content over the corresponding template's shape, it is also possible to use the template's mask and re-purpose the artwork's surrounding empty walls with content without covering the artwork. With the different templates, specific content can be projected on different walls throughout the museum's divisions.

4 Person Detection and Clothes Overlapping

As mention, the goal of the Person Detection and Clothes Overlapping module is to use a mobile device to project AR content (clothes) over persons that are in a museum. On other words, the goal is "to dress" museums' users with clothes from the epoch of the museums' objects. The module has two main steps: (i) the person detection and pose estimation, and the (ii) clothes overlapping. Those steps will be explained in detail in the following sections.

The implementation was done in Unity [57] using the OpenCV library (Asset for Unity). In order to verify the implementation's reliability, computational tests were done in a desktop computer and in a mobile device, namely using a Windows 10 desktop with an Intel i7-6700 running at 3.40 GHz and an ASUS Zenpad 3S 10″ tablet.

Fig. 4. Left to right, example of confusion between left and right ankle, the correct detected pose, and the pose estimation with spatial size of the CNN equal to 368×368 px and 192×192 px. (Color figure online)

The method used for pose estimation was OpenPose (see Sect. 2 and [10, 35]). OpenPose was implemented on TensorFlow [24] and the CNN architecture for feature extraction is MobileNets [28]. The extracted features serve as input for the OpenPose algorithm, that produces confidence maps (or heatmaps) and PAFs maps which are concatenated. The concatenation consists of 57 parts: 18 keypoint confidence maps plus 1 background, and 38 ($= 19 \times 2$) PAFs. The component *joint/body part* of the body, e.g., the right knee, the right hip, or the left shoulder, are shown in Fig. 4, where red and blue circles indicate the person's left and right body parts. A pair of connected parts, *limb*, e.g., the right shoulder connection with the neck are shown in the same figure, the green line segments.

A total amount of 90 frames of expected user navigation were the input to the CNN. Furthermore, two input sizes images for the CNN were tested: 368×368 and 192×192 px. Depending on the size of the input, the average process time for each frame was 236 ms/2031 ms (milliseconds) and 70 ms/588 ms, respectively in the desktop and tablet. As expected, reducing the input size images of the CNN allow attaining improvements on the execution time, but the accuracy of the results dropped. The pose is always estimated, but the confidence map for a body part to be considered valid must be above 25% of the maximum value estimated in the confidence map (this value was empirically chosen), otherwise is not considered. A missing body part example for a 192×192px image which was detected in the 368×368 px image is shown in Fig. 4, right most image. The same figure also shows an example of error that sometimes occurs in the identification of the right and left hands/legs (left most image).

Besided the presented cases, a stabilization method was needed because pose estimated (body part) can wrongly "change" position, for instance due to light changes. The stabilization is done using groups of body parts from the estimated pose. The body parts selection for each group is based on the change that body parts do when any single one moves, see Fig. 5.

Groups	1st	2nd	3rd	4th	5th
Parts	Neck	Right Hip	Left Hip	Right Elbow	Left Elbow
	Right Shoulder	Right Ankle	Left Ankle	Right Wrist	Left Wrist
	Left Shoulder	Right Knee	Left Knee		

Fig. 5. Pose estimation stabilization groups.

Fig. 6. Examples of volume 2D views.

The stabilization algorithm is as follows: (a) for each one of the 5 groups present in Fig. 5, a group of RoIs (one for each body part), with 2% of the width and height of the frame (value chosen empirically), is used to validate if all the body parts from the group have changed position or not. (b) To allow a body part to change position, all the other group body parts must change, i.e., they must have a position change bigger than the RoIs mentioned before. (c) Depending of the group, if one or two body part(s) have a value bigger than the predefined RoIs, this wrong body part(s) is/are replaced by the correct ones, that was/were estimated in a previous frame.

To solve the incorrect detection of the body parts problem, the estimated pose view is used, i.e., to distinguish between right and left body parts it is necessary to validate if the body is in a front or in a back view. (d) This is done by observing that in a front view, the x coordinates of the right side body parts should be smaller than the ones from the left side. To replace a missing body part from a pose is used the previously estimated pose.

In the second phase, the clothes overlapping methods has as input the estimated body parts. For clothes overlapping, three methods were tested: (i) segments, (ii) textures, and (iii) volumes. The first two methods were presented in [5], showing some lack precision and the limitation of only working in frontal view.

For the third method (volumes), the two main steps are: (a) rotate and resize the volume, (b) project the (clothes) volume over the person.

In the first step, (a.1) a clothe volume was developed in 3DS MAX [40] and (a.2) imported to Unity. Then, (a.3) the volume was rotated horizontally

		Body Parts				
		Nose	Right eye	Left eye	Right Ear	Left Ear
Views	Front	1	1	1	1	1
		1	1	1	0	0
		1	1	1	0	1
		1	1	1	1	0
	Back	0	0	0	0	0
		0	0	0	1	1
		0	0	0	1	0
		0	0	0	0	1
	Right Side	1	1	0	1	0
	Left Side	1	0	1	0	1

Fig. 7. Created views conditions represented horizontally. A detected part is represented by 1 and not detected by 0.

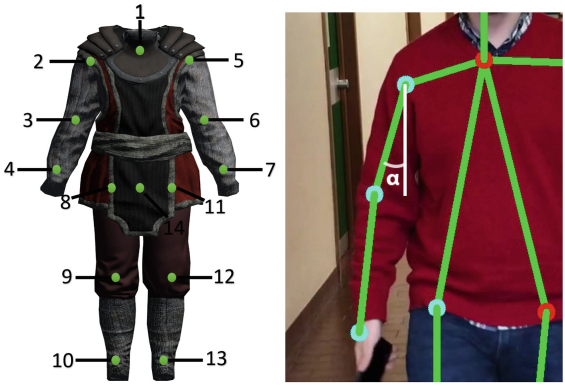

Fig. 8. Left, volume keypoints. Right, example of a limb's angle.

accordingly to four pose views, as presented in Fig. 6 where frontal, back, side right, and side left views of the volume can be seen. (a.4) The views were then associated to the OpenPose detected and non detected body parts (namely: nose, right eye, left eye, right ear and left ear) according with the conditions presented in Fig. 7, where 1 represents a detected body part, and 0 a non detected body part. Additionally, (a.5) to strengthen the assurance of front or back view, the x coordinates distance between right and left hips and shoulders coordinates should be more than 5% of the frame width (this value was empirically chosen). (a.6) A previous view is used if none of the above conditions are met. Finally, (a.7) the volume is resized using the distance between ankles and neck which is an approximation to the person's height.

The resized volume is now project over the detected person (b). To achieve the referred projection, the volume body parts keypoints (see Fig. 8 left) are (b.1) overlapped over the estimated OpenPose pose body part keypoints, and (b.2) rotated accordingly to the angle (α_i) between a vertical alignment and each OpenPose's i-limb, see Fig. 8 right.

Fig. 9. Examples of human shape superimposition using "volumes".

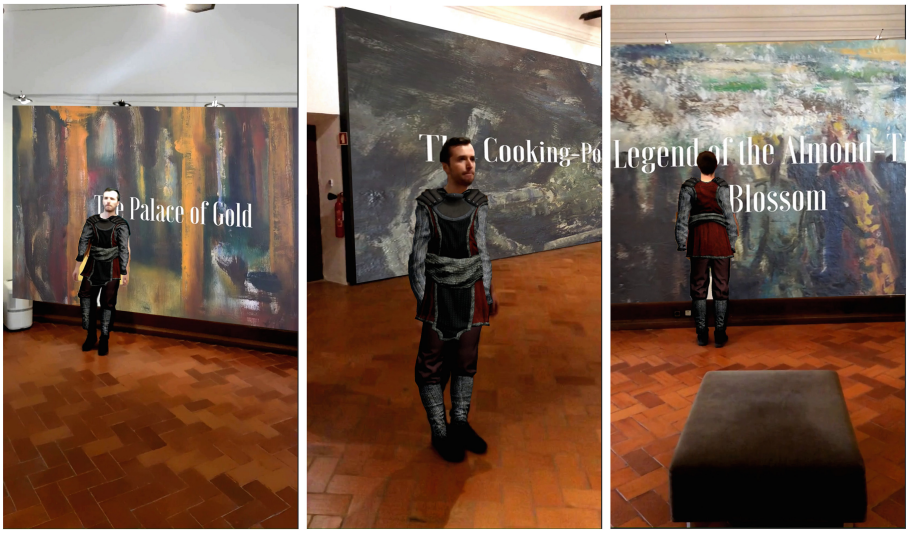

Fig. 10. Examples of both modules working together.

Figure 9 shown results of the overlapped volume in a museum environment. The overlapping volumes over a person takes an average processing time of 6.1 ms/31.4 ms for the desktop and mobile respectively. In general, the overall process takes a mean time of 76.1 ms (70 ms + 6.1 ms) and 590.4 ms (559 ms + 31.4 ms) for the desktop and mobile.

5 Conclusions

This paper presents two modules to be integrated in MIRAR framework [45], namely: the wall segmentation with overlapping of information and the human shapes segmentation with clothes overlapping. Furthermore, the modules were integrated as it can be seen in the examples in Fig. 10.

Regarding the walls' detection and information overlapping, the current results present a functional and fluid experience of content superimposition even with visitors' movements or with acute angles between the camera's position and the superimposed walls. Nevertheless, further tests in different conditions and new environments' implementations are required to improve and evolve the present algorithm into a more broad and stable performance.

For human clothes overlapping in real involvements (museum in this case), the proposed method combines OpenPose body parts detection with volumes overlapping. For better pose estimation accuracy in mobile devices, a stabilization method and the pose views were created. For real-time performances on mobile devices an OpenPose model with a MobileNet architecture was used and two input image sizes were tested (namely, 368×368 and 192×192px). The smallest size is the best option for mobile devices in term of execution time, but it is worse in term of accuracy, nevertheless is a good trade-off for the application.

For future work, a faster and more accurate performance with OpenPose could be achieved by testing new network architectures, new training strategies and other datasets. Another way to get better pose estimation results could be achieved by testing models like PersonLab [44] or others. For this specific module, other way to do pose view estimation is to train a model to do body/foot keypoints estimation and use the foot keypoints position to know the pose view. Additionally, to predict 3D poses by using the estimated 2D poses, the "lifting" system implementation could be done. In the case of the indoor localization through only computer vision is still not resolved, with the necessity of creating a new compatible method to our present tracking system. There is also a need to develop a mixed 3D layout of the regular museums' rooms in order to be able to totally replace the environment if needed. This would also allow, especially with the seamless tracking, the possibility of superimposing advanced 3D models contents that could offer better information, orientation or navigation through the user's visit, fully immersing the visitor in this new era museums' experience.

Acknowledgements. This work was supported by the Portuguese Foundation for Science and Technology (FCT), project LARSyS (UID/EEA/50009/2013), CIAC, and project M5SAR I&DT nr. 3322 financed by CRESC ALGARVE2020, PORTUGAL2020 and FEDER. We also thank Faro Municipal Museum and the M5SAR project leader, SPIC - Creative Solutions [www.spic.pt].

References

1. Araki, N., Muraoka, Y.: Follow-the-trial-fitter: real-time dressing without undressing. In: Proceedings of IEEE Conference on Digital Information Management, London, UK, pp. 33–38 (2008)
2. Artoolkit: ARtoolKit, the world's most widely used tracking library for augmented reality (2017). http://artoolkit.org/. Accessed 16 Nov 2017
3. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. IEEE Comput. Graph. Appl. **21**(6), 34–47 (2001)
4. Babahajiani, P., Fan, L., Gabbouj, M.: Object recognition in 3D point cloud of urban street scene. In: Jawahar, C.V., Shan, S. (eds.) ACCV 2014. LNCS, vol. 9008, pp. 177–190. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16628-5_13
5. Bajireanu, R., et al.: Mobile human shape superimposition: an initial approach using OpenPose. In: Proceedings 18th International Conference on Applied Computer Science (2018)
6. Bartoli, A., Sturm, P.: Structure-from-motion using lines: representation, triangulation, and bundle adjustment. Comput. Vis. Image Underst. **100**(3), 416–441 (2005)
7. Bouguet, J.-Y.: Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm. Intel Corporation **5**(1–10), 4 (2001)
8. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
9. Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., Sheikh, Y.: Openpose: realtime multi-person 2D pose estimation using part affinity fields. arXiv preprint arXiv:1812.08008 (2018)
10. Cao, Z., Simon, T., Wei, S.-E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: CVPR, vol. 1, no. 2, p. 7 (2017)
11. Catchoom: Catchoom (2017). http://catchoom.com/. Accessed 16 Nov 2017
12. Cheng, K.-H., Tsai, C.-C.: Affordances of augmented reality in science learning: suggestions for future research. J. Sci. Educ. Technol. **22**(4), 449–462 (2013)
13. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Rob. Autom. Mag. **13**(2), 99–110 (2006)
14. Elqursh, A., Elgammal, A.: Line-based relative pose estimation. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp. 3049–3056. IEEE (2011)
15. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2018)
16. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54

17. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: Proceedings IEEE International Conference on Computer Vision, pp. 1449–1456 (2013)
18. Erra, U., Scanniello, G., Colonnese, V.: Exploring the effectiveness of an augmented reality dressing room. Multimedia Tools Appl., 1–31 (2018)
19. Facecake: Facecake (2016). http://www.facecake.com/. Accessed 17 September 2018
20. Fang, H., Xie, S., Tai, Y.-W., Lu, C.: RMPE: regional multi-person pose estimation. In: Proceedings IEEE International Conference on Computer Vision, vol. 2 (2017)
21. Gimeno, J., Portales, C., Coma, I., Fernandez, M., Martinez, B.: Combining traditional and indirect augmented reality for indoor crowded environments. A case study on the casa batlló museum. Comput. Graph. **69**, 92–103 (2017)
22. Girshick, R.: Fast R-CNN. In: Proceedings IEEE Conference on Computer Vision, pp. 1440–1448 (2015)
23. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
24. Google: TensorFlow - an open-source machine learning framework for everyone (2018). https://www.tensorflow.org/. Accessed 14 Jan 2018
25. Gupta, S., Arbeláez, P., Girshick, R., Malik, J.: Indoor scene understanding with RGB-D images: bottom-up segmentation, object detection and semantic segmentation. Int. J. Comput. Vis. **112**(2), 133–149 (2015)
26. Haines, O., Calway, A.: Detecting planes and estimating their orientation from a single image. In: BMVC, pp. 1–11 (2012)
27. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
28. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
29. Huang, J., et al.: Speed/Accuracy trade-offs for modern convolutional object detectors. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, vol. 4, Honolulu, HI, USA, pp. 3296–3297 (2017)
30. Hulik, R., Spanel, M., Smrz, P., Materna, Z.: Continuous plane detection in point-cloud data based on 3D Hough transform. J. Vis. Commun. Image Represent. **25**(1), 86–97 (2014)
31. InformationWeek: Informationweek: 10 fantastic iPhone, Android Apps for museum visits (2017). https://goo.gl/XF3rj4. Accessed 04 April 2017
32. Isıkdogan, F., Kara, G.: A real time virtual dressing room application using Kinect. Computer Vision Course Project (2012)
33. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DOF camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2938–2946 (2015)
34. Fitnect Interactive Kft. Fitnect (2016). http://www.fitnect.hu/. Accessed 17 Sept 2018
35. Ildoo Kim: tf-pose-estimation (2018). https://bit.ly/2HJxxcq. Accessed 10 April 2018
36. Kiryati, N., Eldar, Y., Bruckstein, A.M.: A probabilistic Hough transform. Pattern Recogn. **24**(4), 303–316 (1991)
37. Layar: Layar (2017). https://www.layar.com/. Accessed 16 Nov 2017
38. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: binary robust invariant scalable keypoints. In: Proceedings IEEE International Conference on Computer Vision, pp. 2548–2555. IEEE (2011)

39. Mallya, A., Lazebnik, S.: Learning informative edge maps for indoor scene layout prediction. In: Proceedings IEEE International Conference on Computer Vision, pp. 936–944 (2015)
40. 3DS MAX: 3DS MAX (2018). https://www.autodesk.com/products/3ds-max/overview. Accessed 3 Dezember 2018
41. Muja, M., Lowe, D.G.: Fast matching of binary features. In: Proceedings 9th Conference Computer and Robot Vision, pp. 404–410. IEEE (2012)
42. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Rob. **31**(5), 1147–1163 (2015)
43. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
44. Papandreou, G., Zhu, T., Chen, L.-C., Gidaris, S., Tompson, J., Murphy, K.: PersonLab: person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. arXiv preprint arXiv:1803.08225 (2018)
45. Pereira, J.A.R., Veiga, R.J.M., de Freitas, M.A.G., Sardo, J.D.P., Cardoso, P.J.S., Rodrigues, J.M.F.: MIRAR: mobile image recognition based augmented reality framework. In: Mortal, A., et al. (eds.) INCREaSE 2017, pp. 321–337. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-70272-8_27
46. Portales, C., Vinals, M.J., Alonso-Monasterio, P.: AR-immersive cinema at the aula natura visitors center. IEEE MultiMedia **17**(4), 8–15 (2010)
47. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
48. Ren, Z., Sudderth, E.B.: Three-dimensional object detection and layout prediction using clouds of oriented gradients. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp. 1525–1533 (2016)
49. Rodrigues, J.M.F., et al.: Adaptive card design UI implementation for an augmented reality museum application. In: Proceedings 11th International Conference on Universal Access in Human-Computer Interaction (2017)
50. Rodrigues, J.M.F., et al.: Mobile augmented reality framework - MIRAR. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2018. LNCS, vol. 10908, pp. 102–121. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92052-8_9
51. Serrão, M., et al.: Computer vision and GIS for the navigation of blind persons in buildings. Univ. Access Inf. Soc. **14**(1), 67–80 (2015)
52. Shi, J., Tomasi, C.: Good features to track. Technical report, Cornell University (1993)
53. Tareen, S.A.K., Saleem, Z.: A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. In: Proceedings International Conference on Computing, Mathematics and Engineering Technologies, pp. 1–10. IEEE (2018)
54. Tateno, K., Tombari, F., Laina, I., Navab, N.: CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, vol. 2 (2017)
55. Tome, D., Russell, C., Agapito, L.: Lifting from the deep: convolutional 3D pose estimation from a single image. In: Proceedings IEEE Conference Computer Vision and Pattern Recognition, pp. 2500–2509 (2017)
56. TWSJ: The wall street journal: best apps for visiting museums (2017). https://goo.gl/cPTyP9. Accessed 4 April 2017
57. Unity: Unity3D (2018). https://unity3d.com/pt. Accessed 10 Jan 2018
58. Vainstein, N., Kuflik, T., Lanir, J.: Towards using mobile, head-worn displays in cultural heritage: user requirements and a research agenda. In: Proceedings 21st International Conference on Intelligent User Interfaces, pp. 327–331. ACM (2016)

59. Veiga, R.J.M., Bajireanu, R., Pereira, J.A.R., Sardo, J.D.P., Cardoso, P.J.S., Rodrigues, J.M.F.: Indoor environment and human shape detection for augmented reality: an initial study. In: Proceedings 23rd Portuguese Conference Pattern Recognition, p. 21 (2017)
60. Veiga, R.J.M., Pereira, J.A.R., Sardo, J.D.P., Bajireanu, R., Cardoso, P.J.S., Rodrigues, J.M.F.: Augmented reality indoor environment detection: proof-of-concept. In: Proceedings Applied Mathematics And Computer Science (2018)