Augmented Tattoo: evaluation of an augmented reality system for tattoo visualization

Jairo Calmon
Department of Exact Sciences (DEXA)
State University of Feira de Santana
Feira de Santana, Bahia, Brazil
jairocalmon@ecomp.uefs.br

João Queiroz
Institute of Arts And Design (IAD)
Federal University of Juiz de Fora
Juiz de Fora, Minas Gerais, Brazil
queirozj@pq.cnpq.br

Claudio Goes, Angelo Loula
Department of Exact Sciences (DEXA)
State University of Feira de Santana
Feira de Santana, Bahia, Brazil
cegoes@gmail.com,
angelocl@ecomp.uefs.br

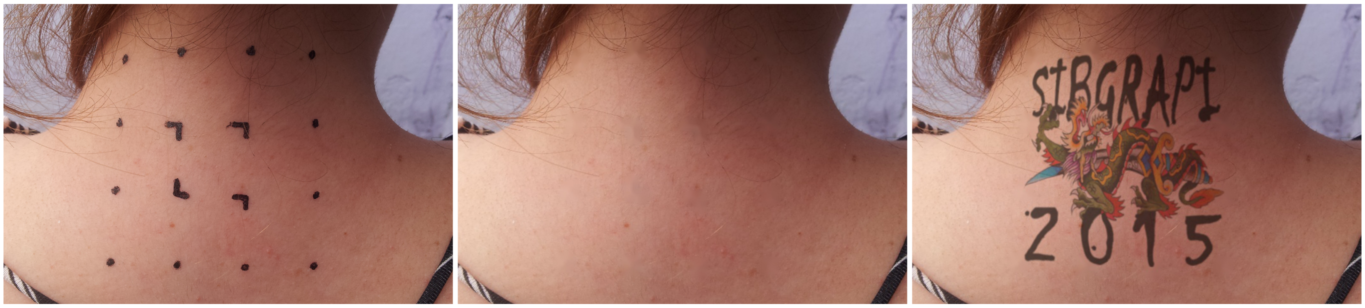


Fig. 1. Teasing result of our method: from the captured image input (left), the markers were detected, registered and inpainted (middle), then, a tattoo was superimposed producing effective result (right).

***Abstract*—Augmented Reality complements reality overlaying computer modeled virtual objects in real world images, allowing users to be immersed in synthetic environments. Tattoos are an ancient cultural practice, performed for centuries, with intervention in epithelial layers of the human body, and technological devices can further expand this practice. Here we describe and evaluate an Augmented Reality system for visualization of virtual tattoos on the skin, accompanying the surface of the human body. From a captured image, the system detects markers on the skin, and outputs an image without the markers with an AR tattoo inscribed to the skin. Markers detection provides high accuracy rates that allow adequate mesh and, along with skin segmentation and markers removal, the system generates suitable final images. Our system reveals robustness concerning the several possible operation conditions, dealing with different lighting conditions, background content, skin tones, drawing quality of the markers in the skin, and possible occlusions of skin and markers.**

***Keywords*-augmented reality; tattoo; markers**

I. Introduction

Graphics technologies designed to immerse the user in synthetic environments, are used for many different purposes, such as medicine, education, therapy, entertainment, robotics, among others [1]. Augmented Reality (AR) allows the user to visualize the real world, by overlaying computer modeled virtual objects. AR applications complement the reality, rather than trying to replace it [1], from the real-time view of a physical real world that is modified by the addition of computer-generated information [2].

The interactions between the real and the virtual world in Augmented Reality applications are carried, for the most part, in a non-intrusive approach. On the other hand, there are ancient cultural practices, performed for centuries, regarding intervention in epithelial layers of the human body [3]. There are various types of interventions done in the body, including the most popular and recent, such as tattooing and piercing, and the motivations of its users are diverse. Recent data estimate that 23% of Americans and 21% of Canadians have at least one tattoo. It is estimated that these people consume in tattoos, in the US alone, an amount of 1.65 billion dollars per year [4]. Thus, tattoos were the motivation for this project although the resulting work can be used in/for different fields and applications.

This paper describes and evaluates an Augmented Reality system for visualization of virtual tattoos on the skin, accompanying the surface of the human body. It is, therefore, an exploration of new human-computer interfaces coupled with the motivation of individuals to modify their bodies. This new form of tattoo – virtual – allows dynamic approaches, and even iterative and multimodal, through animations and videos, plus the ability for the user to change it when wanted.

The next section presents a discussion about tattoo, technologies involved and related work. Section III describes the methodology used, detailing specifications and system constraints and also the steps of the proposed system. Section IV presents the results, and finally, closing remarks.

II. Tattoo and Technology

Physical intervention on body surface constitutes practices observed in various cultures, motivated for many reasons. In the last decade, the popularity of tattoos and piercings increased dramatically, both in numbers and in relation to the spectrum of social classes of its users [3]. The motivations to perform body interaction through tattooing are diverse. A study described in [3] defined some categories that describe the impulses to modify the body: beauty, art and fashion; personal narrative; physical tolerance; group membership and commitment; resistance; spirituality and cultural tradition; addiction; and sexual motivation.

In a simplified manner, a tattoo is an invasive inscription on the second layer of skin (dermis) through repeated micro-incisions. Thus, new approaches for tattoo development should be based on its stand, the skin. After a long period without significant changes in technological design and implementation, new techno-scientific devices have begun to emerge, allowing, for example, the use dynamic epithelial devices, i.e., devices deliberately inscribed or attached to the skin in which the appearance is potentially modified along time [5].

A. Augmented Reality and Tattoos

Augmented Reality (AR) is defined as direct or indirect real-time visualization of the physical world which has been modified by the addition of computer-generated information [2]. The practical applications arising from the technology are diverse, since the virtual objects add to the real world information that the user can not directly perceive. Thus, the information provided by the virtual objects can help the user perform very different tasks, among which we can mention: medical imaging, entertainment, advertising, maintenance and repair, annotation, route planning, among others [2].

No academic works in AR were found, involving the visualization of tattoos by detecting the skin surface, although there are some commercial projects and applications for mobile devices. In these cases, a marker is tattooed, or the user's own tattoo is used; then the application detects and superimposes a virtual animated object, such as ThinkAnApp project [6]. These projects are similar to detection made with plain cards, the difference being only on the nature of the surface in use. Thus, the virtual object is only positioned so as to float on the skin without any deformation or integration with the surface.

The use of AR projections on the skin can be seen in many research fields such as, for example, in medicine. Nicolau et al. [7], for example, have an AR system to aid medical procedures in the liver. Important similarities between this application and our work can be identified, such as the use of markers on the skin. In contrast, a considerable difference is in the fact that the medical application is based on a very controlled environment and makes use of sophisticated capture devices, expensive and of limited use, as computerized tomography.

III. Methodology

While many works use the skin only as a surface to hold a card to be detected by an AR application, we describe an approach capable of detecting deformations of the skin surface, so that the virtual tattoo is adjusted to follow such deformations and actually look like it is inscribed in the skin.

During system design, various restrictions and specifications were considered. The resulting application can be used by diverse users in different body parts in uncontrolled and minimally prepared environments (only markers). Images come from regular consumer monocular cameras, so it is necessary to analyze the image searching for information that can be used to obtain skin surface pose as well as its deformation and possible occlusions. Therefore, skin detection was obtained in this work by 2D registration provided by the detection and identification of adhoc markers painted in the skin. Although this technique demands skin modification (by marker inscription), it was considered the most adequate due to its low computational cost, preservation of skin texture, tolerance to occlusions and easy removal from the scene.

The system was developed in a desktop environment, using OpenCV computational vision library along with OpenGL graphics libraries. The system starts by capturing an image of a body part where the skin holds painted markers (background is not controlled) and outputs these images of the body part without the markers (same background) with an AR tattoo inscribed to the skin, considering skin surface deformation and occlusions.

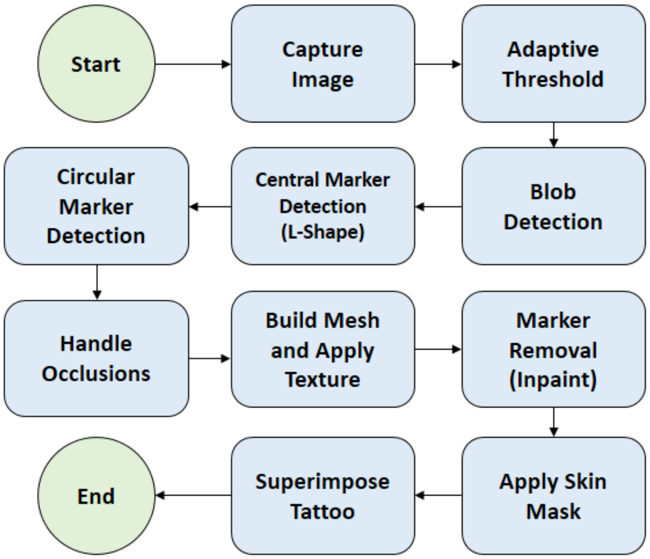


Fig. 2. Flow diagram of the processing steps executed by the system.

The processing steps of the proposed system are represented in the Fig. 2. The process starts with a single image captured by a monocular camera and the system tries to identify the markers as shown in Fig. 3. For this purpose, the image is segmented using an adaptive threshold algorithm followed by blob detection. First, a search for central markers is performed. If central markers are found, the system searches for peripheral circular blobs and handle possible occlusions of these markers. Then, the markers positions are used to build a mesh with the tattoo, rendering a separated image. A skin mask is then applied to handle skin boundaries and possible skin occlusions. Marker blobs are removed from the original captured image (inpainting). The masked tattoo image is then superimposed in the inpainted image and this frame is ready to be shown. The next sections describe in more detail each of the steps.

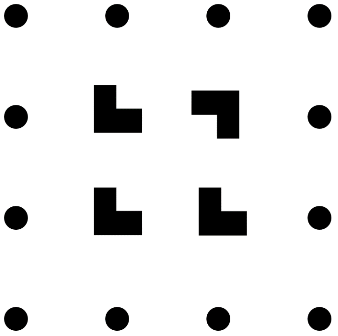


Fig. 3. Complete marker composed of four central L-shapes and twelve circular markers.

A. Dataset

In order to define parameters and evaluate the system, an image dataset was assembled. This dataset was divided in subsets – set of photos under similar conditions (same person and same marker imprinting). Images in each subset followed a protocol for image capture requirements:

1) Frontal shot – close.
2) Frontal shot – mid distance.
3) Frontal shot – far distance.
4) Object occlusion (entire line or column).
5) Object occlusion (one marker).
6) Finger occlusion.
7) Free distortion.
8) Perspective shot (side or bottom).
9) Extreme perspective shot.

The complete photo dataset has a total of 9 subsets and 120 images, from 7 different individuals of different skin tones. All images are scaled to a predefined size (720p). At least one photo for each requirement was included in every subset. Some variations and mixing between requirements were allowed to be added. One person submitted two subsets – white and black markers, another submitted two subsets with different backgrounds and the same markings, while the others submitted one subset each. Markers were painted by free hand trying to follow the given pattern, using blue ballpoint pen, black marker pen and white paint. It is important to note that more important than the color of the markers is the contrast between the marker and the skin.

Each of the 120 images was labelled manually. The collected labels for each photo include the size of the image, the position of each individual marker and whether it is occluded (in which case a rough estimate of the position was given). This was important to automate the process of adjusting the parameters of the system.

A video dataset was also assembled to evaluate the developed system, allowing tests more similar to real system use system and including other aspects that can be avoided in still images (occurrence of blur, for instance). The defined protocol demands at least four videos per subset, each having a few seconds (3–10 seconds) showing the following situations:

- Frontal to lateral shot.
- Freely moving the body.
- Rotating the camera (forward axis).
- Introducing occlusion.

In order to automate the process of testing, frame samples were collected from the videos (two samples per second), resulting in 372 samples for the entire dataset.

B. Markers

The first step in the system development was to define the markers to be used. To design the markers some requirements were taken in consideration.

> *Marker size* – The markers should be small, since the skin had to be preserved as much as possible and marker removal process could be done effectively.
>
> *Easy of application* – The marker should be easy to insert on the skin surface and should not require specific materials – a pen should be enough.
>
> *Robustness* – As the user is responsible for drawing the marker on the skin surface, color, size and thickness variations, smudges, among others can occur, and the application should still return reasonable output.
>
> *Occlusion* – Even though the marker is partially occluded it should still be possible to be detected with satisfactory results.

A grid of markers was designed to be detected in the captured image. Each marker corresponds to a vertex in a regular grid allowing mesh construction in a following step. The size of each marker can vary but it should be roughly as shown in Fig. 3. To handle orientation and occlusion we decided to divide the marker in two parts: (a) a central part composed of "L-shaped" markers that provide important placement and orientation cues, and thus, must be visible and (b) dots around the central marker that refine surface information (deformations) and some of them can be occluded (at most four of them).

The shape and arrangement of the central markers was chosen to simplify marker detection and reduce the occurrence of false positives, besides the requirements listed before. The search for this central marker follows the rules: (a) there must be four L-shapes; (b) the arrangement should form a rough quad; (c) one of the markers should be rotated by 90 degrees.

Fig. 3 shows the complete grid of markers. Fig. 4 presents close-ups of the markers in the photo dataset.

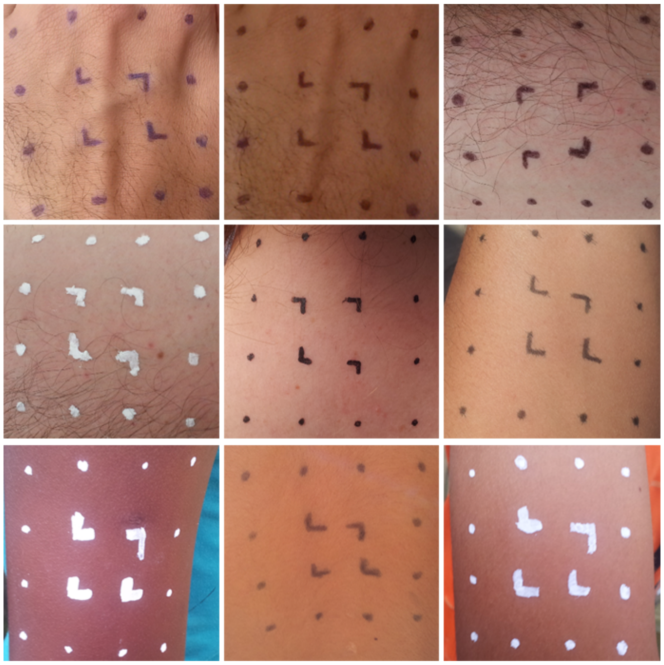


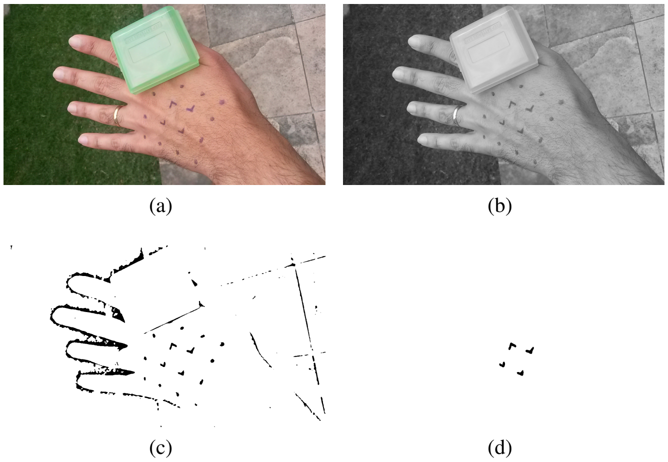
Fig. 4. Close-ups of markers in the dataset showing different painting materials and robustness to variations.

C. L-Shape Markers

The marker detection process was separated in two steps: finding the central markers (the four L-shapes) and finding the peripheral blobs. In this section we describe the central marker detection process.

First, the captured image is converted to grayscale and a mean blur with a given kernel parameter is applied (Fig. 5-b). Converting to grayscale allows the prompt use of adaptive thresholding [8], which is applied to segment the markers from the background (skin) as shown in Fig. 5-c. While the skin is relatively uniform (which would make the use of global thresholding techniques suitable), due to the nature of the surface and uncontrolled lighting conditions, effects like gradients frequently occurs, which is better processed by the adaptive version. The algorithm calculates thresholds in regions of a given block size surrounding each pixel (i.e. local neighborhoods). Each threshold value is the weighted mean of the local neighborhood minus a given offset value [8]. After the adaptive threshold, morphological operations [9] (erosion and dilatation) may take place and helps to eliminate noise using a default $3x3$ rectangular structuring element.

A contour detection algorithm [10] is applied to extract from the image a set of marker candidate contours. Contours retrieval is a useful tool for shape analysis and object detection and recognition, and in this context represent each of the possible markers. Some properties like moments, area, perimeter, center of mass and others can be extracted from the contours, which can then be used to filter contours as needed. All these properties describes what we named "blob".



(a)

(b)

(c)

(d)

Fig. 5. (a) Captured Image. (b) Gray scale image. (c) Adaptive Threshold. (d) Selected central marker.

Next, two simple filters were applied to reduce the search space for the central markers: (a) blob size filter to discard blobs either too small or too big , and (b) a noise filter that discard further processing if too many blobs are still present.

The previous steps for central markers detection have some parameters that needed adjustments to improve the detection rate of markers: block size and offset parameters of the adaptive threshold, and also parameters for blurring and morphological operations. To find out which configuration yielded best results we fed the system with the photo dataset and a generated set of 240 different configurations, varying each parameter. For each configuration, we checked if the generated blob set contained four blobs that corresponded position-wise to each of the four manually labeled central markers, in the respective photo. After applying each configuration to all images in the dataset, a detection rate was determined for each configuration.

The configuration with the best overall detection rate (75%) was selected to be used in the system. Seven additional configurations were also selected, based on the number of cases each could solve that the previous configurations could not, complementing overall detection (97.5%). These set of configurations support different image conditions and they are used one a time, in a alternating sequence ordered by detection rates, when the noise filter interrupts marker detection or if the system cannot find valid central markers candidates. The list of configurations is shown below in the form $\{$*blur kernel, threshold block size, threshold offset, morphological iterations*$\}$.

1) $5, 63, 30, 0.$
2) $7, 33, 25, 1.$
3) $3, 53, 20, 0.$
4) $7, 53, 25, 1.$
5) $5, 53, 15, 1.$
6) $1, 33, 15, 0.$
7) $1, 13, 20, 1.$
8) $5, 13, 25, 0.$

D. L-Shape Matching

The previous step provides a list of marker candidates as blob contours. To determine which blobs have an L-shape, a contour matching algorithm compares them against a set of L-shaped template contours (representing possible variations). The algorithm returns a similarity value for each blob compared to each template. The similarity value $I(A, B)$ between shapes A and B is determined using Equation 1, where m_i corresponds to the Hu Moments [11] of the contour.

$$I(A, B) = \underset{i=1..7}{max} \frac{m_i^A - m_i^B}{m_i^A} \tag{1}$$

To determine a maximum similarity value that distinguishes L-shaped blobs from ordinary ones, an auxiliary dataset was composed. This dataset was composed from 1418 blob samples from the photo dataset, which were manually labeled as L-shaped or not. After determining similarity values for all blob samples in the dataset, a maximum similarity value of 0.5 was defined.

After filtering the L-shaped blobs, an algorithm was applied to find sets of four blobs that roughly form a quad as, even under perspective or distortions, approximate spatial relations between each central blobs is still preserved. This algorithm iterates through each blob and filter out blobs that are not approximately the same size. Then, it checks whether the three closest blobs to the current one roughly form a quad. This step also registers each candidate marker based on its orientation, using the rotated marker as a starting point. The resulting set can be represented by a 4-tuple of markers $C = L_1, L_2, L_3, L_4$, where L_1 is the rotated marker and the others are assigned clockwise.

The previous step outputs a set of 4-tuple candidates representing possible solutions for the central marker. A single 4-tuple is selected as the one with the lowest (a) variance of the quad sides and (b) variance of the orientations of markers L_2, L_3, L_4 (blobs with same orientation). Fig. 5 shows the steps to detected the central marker.

E. Circular Markers

At this point, the central markers were properly registered and the system proceeds to circular markers detection. In order to remove distortions and facilitate the search for markers and the registration of circular markers, a rectification (warping) is performed in the thresholded image (before L-Shape matching filtering). This warping process (Fig. 6) consists of an affine transformation – a matrix multiplication (linear transformation) followed by a vector addition (translation) – using the central markers as control points to scale, rotate and translate to match a reference template grid.

In this template grid all markers are registered, so the system must match each of them to blobs in the rectified image (Fig. 7). If there was no distortion in the original image the circular blob positions should match exactly a template grid with the same spatial relation as the rectified image. These distortions are limited though and this premade grid of points can be used to find which blob in the rectified image corresponds to each position in the grid. This is achieved by a keypoint descriptor matcher, with each blob corresponding to a keypoint with a descriptor composed solely by its location in the image. For each descriptor in the first set (template grid), the matcher finds the closest descriptor in the second set (rectified image). A brute force approach was chosen due to its simplicity and the small number of blobs.

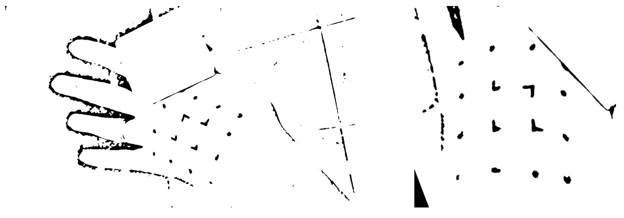

(a) (b)

Fig. 6. Image Rectification (a) Binary Image. (b) Warped Binary image.

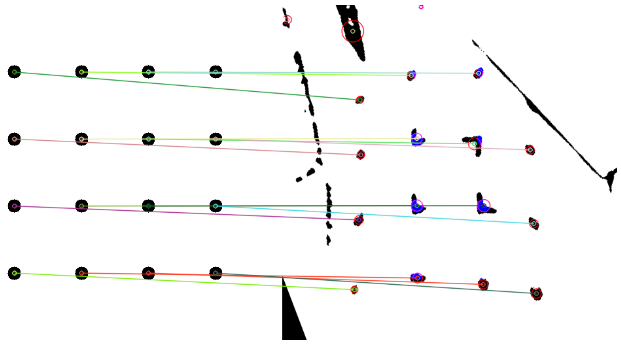


Fig. 7. Matching process. In the left the template grid and in the right the rectified image. Lines represent the matches.

F. Missing Markers / Occlusion

The previous step returns a set of registered blobs. This means that each blob represents a specific marker. Some markers might be occluded or not be correctly detected by the previous steps, though. The system should ensure some degree of robustness in these situations, so the occlusion is handled by interpolating/extrapolating from the visible detected markers. For each occluded marker the process analyzes the current line and column searching for non-occluded blobs and uses its positions to predict the final position, prioritizing lines and columns that have 3 visible markers. This procedure gives a fair estimate of occluded blobs locations.

At this point, we already have a complete set of blobs and its respective positions that can be used to draw the tattoo. In the next sections we describe the rendering process, achieved by erasing markers, handling object occlusions, superimposing the tattoo. Fig. 8 shows the tattoo image used.

G. Mesh Building

In order to get a 2D surface model for the skin surface, a mesh of triangles is built using OpenGL using the position of

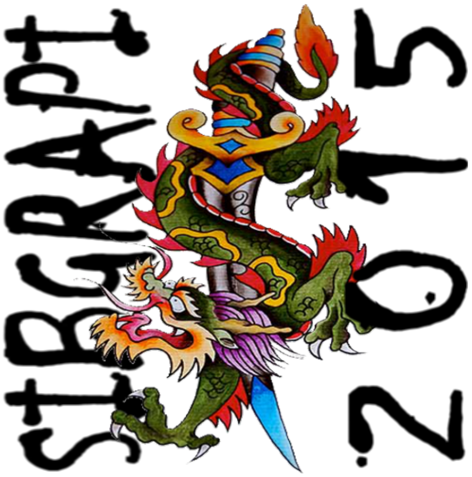


Fig. 8. Tattoo image.

the detected markers and interpolated markers. The position of each vertex is related with the center of mass coordinates of its respective marker and, therefore, the tattoo is positioned and distorted accordingly and has the same size of the mesh.

In the mesh construction process, the UV mapping (2D representation of the 3D surface mapping) is performed and the texture of the tattoo, chosen by the user, is assigned to the mesh. Thus, this rendered OpenGL mesh follows the same deformation of the skin surface.

The rendering of this process is done separately (not overlaying yet), since it must handle the case of possible occlusions, and perform the blending process, which may involve more than simple overlay.

H. Occlusion Handling

The previous step provides a rendered image of a mesh of vertices at positions corresponding to the respective markers. However, parts of the skin region bounded by the grid may not be visible in the original image due to occlusions and thus the rendering process should consider this restriction, not displaying the tattoo in areas other than the skin.

In order to apply the tattoo only on skin-visible areas of the surface, a segmentation operation is performed in the original image. This operation consists of a simple per pixel color range selection operation, and the threshold parameters were configured to wide range values in order to work with most skin tones. The selection of this parameter it is a trade-off that affects how well the skin is segmented at skin boundaries as was defined empirically based on the photo dataset. The result of this operation is a skin mask which is then used to cut the desired part of the image of the tattoo as shown in Fig. 9.

I. Marker Removal (Inpainting)

One of the stages preceding the process of tattoo overlay, however, is the removal of the original markers in the image. This is necessary to improve the final result, since the tattoo to be inserted may have hollow areas, and such labels may impair visualization. To accomplish the removal of such markers it was used an inpainting technique available in OpenCV, which has the purpose of restoring small parts of images [12].

The inpainting technique uses neighborhood information of the areas to be restored to determine its appearance. In our system, the restored areas are the areas corresponding to the markers, which are excluded from the original image at this stage and are replaced by skin texture (the neighborhood of markers), providing a new marker-free image to finally overlap the tattoo (Fig. 10).

J. Tattoo Overlaying

In this step, the skin-masked and already deformed tattoo image (section III-H) is superimposed on the inpainted image (section III-I), as shown in Fig. 9. The overlaying process is done through a blending operation.

In most cases, the tattoo is an image that is composed of a drawing that has some fully transparent parts. However, it is important to note that the opaque part of the image (the drawing itself) is also treated by the blending operation, in order to preserve some texture of the skin in these areas. This blending was performed using a weighted multiplication between images.

IV. Results and Discussion

In order to evaluate our system, we performed a series of tests using the datasets described in section III-A. Note that the photo dataset was used to adjust parameters, but the videos samples dataset was not used for system development. Images from both datasets were used for evaluation and both show several different conditions (uncontrolled lighting, different skin tones, background variation, etc.) and thus we will also describe how our system behaves in each situation.

System evaluation was performed to test quality of (a) the central marker detection step and (b) the peripheral markers detection step . These two marker types were tested separately because the circular markers detection depends upon the central marker detection phase (so when the central marker detection outputs wrong results, there is little sense in analyzing further). In the central marker detection step, the system correctly identified the 4 central markers in 105 of the 120 captured images, obtaining 87.5% of accuracy. In 6 images no set of 4 markers was found and in 9 images a wrong set of central markers was obtained. A brief qualitative analysis shows that of these 15 images, 13 of them are included in difficult situations introduced in the dataset to test the limits of the system (far and perspective shots).

For the next phase, the detection of peripheral circular blobs, we will consider only the 105 images that passed the previous phase (as the detection of the circular markers depends upon the right detection of the central markers). A total of 1260 circular markers were present in the dataset and 1148 of them were visible. Among the visible markers, our system could correctly identify 1059 markers (92% of accuracy). It is worth mentioning that it is possible that non-detected markers may still be approximately positioned by the interpolation process.

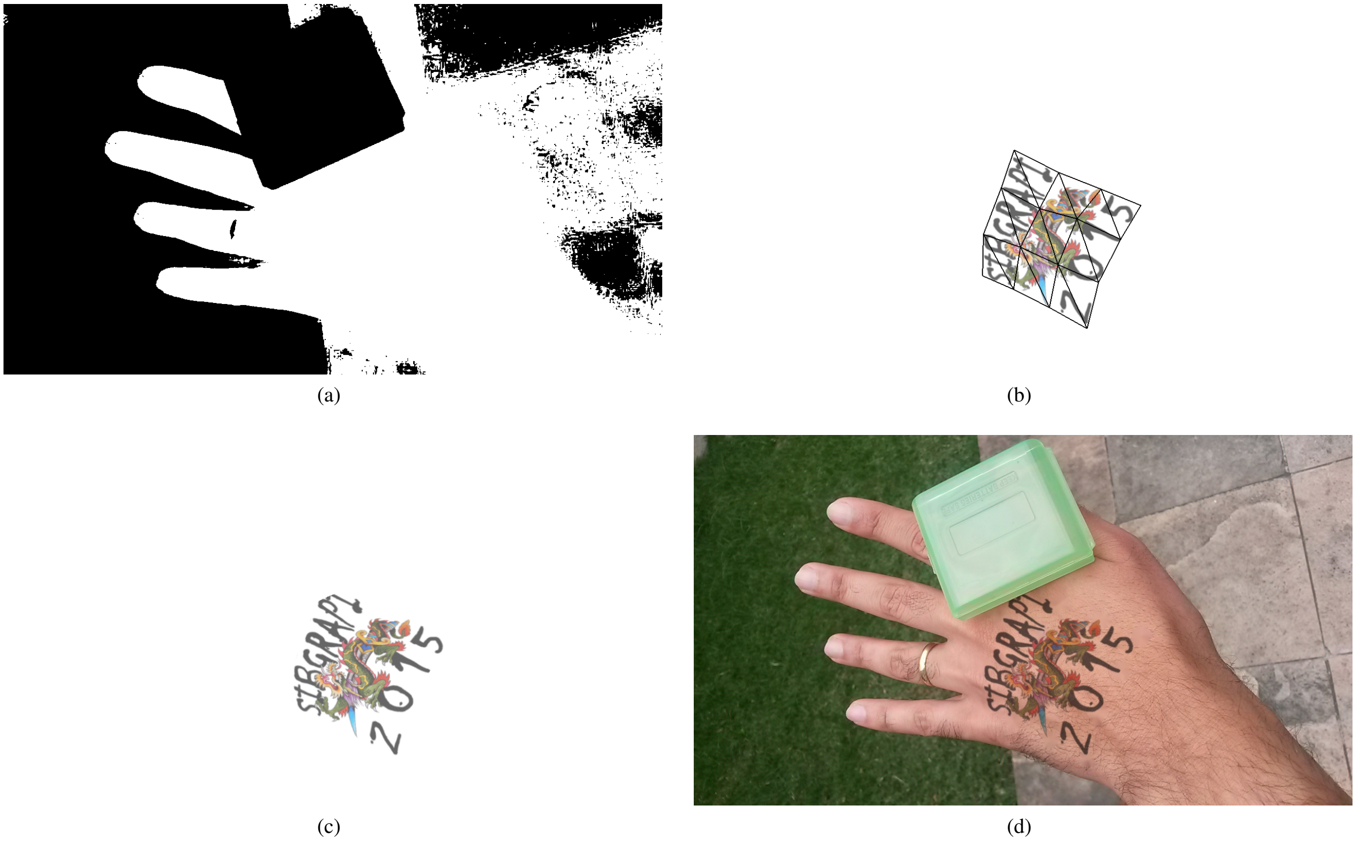


Fig. 9. Rendering of markers. (a) Skin mask. (b) Wireframe of the mesh. Each vertice is located at its respective marker position. (c) Tattoo cutted with the skin mask (ready to be overlaid). (d) Overlaying the tattoo image onto the inpainted image.

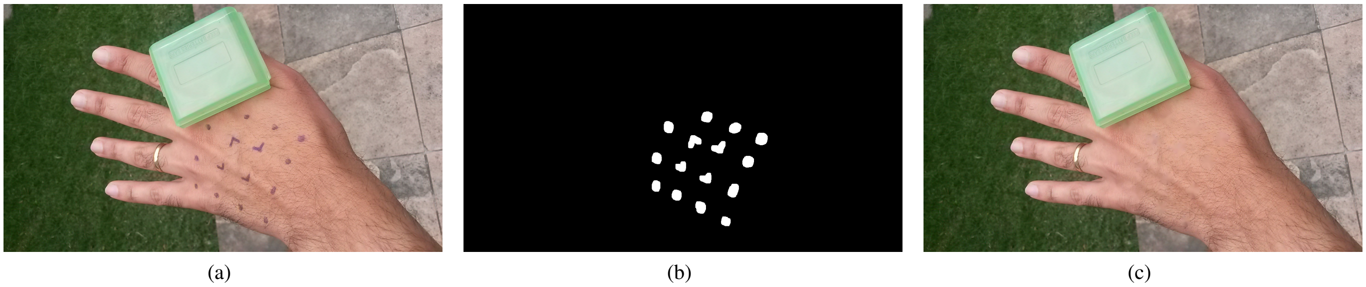


Fig. 10. Inpainting process. (a) The captured image; (b) The markers are the mask of the inpaint process. (c) The markers are erased by replacing it by its neighbor area (skin).

Besides marker detection evaluation, we also qualitatively evaluated the system final rendering step. An automatic evaluation is not trivial as it involves the user perception of how good the final image was presented. In addition, some errors from previous phases influence the quality of the result even though it is not necessarily a rendering error (a poor occlusion handling can produce distorted results, for instance), thus a more qualitative analysis is needed.

The most severe error that affects the system at this moment is the skin-masking process applied to render the tattoo over the original image. For example, in some images, skin segmentation provides a mask that cuts out skin segments. Other artifacts also appear in some specific situations, like when the blob to be inpainted is too close to an object or the skin boundary (as it will check the neighborhood and sample wrong colors). Another issue occurs when a marker is not detected. Even though the algorithm can provide a rough estimate, the rendering process is not able to erase it. Fig. 11 illustrates some of the possible errors that can occur during the rendering phase. However, for most of the tested images, the system outputs appropriate results. Fig. 12 shows some of the results in the photo dataset.

The video samples dataset had 372 images extracted from different videos and was also used to evaluate the system. The

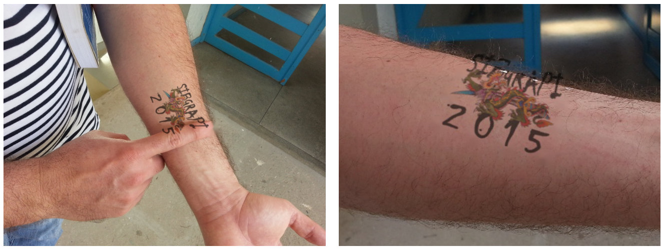

(a) (b)

Fig. 11. Some of the possible rendering problems. (a) The system could not cut out the finger as it is also skin. (b) The system could not detect a precise boundary in the top of the image (hairy texture).

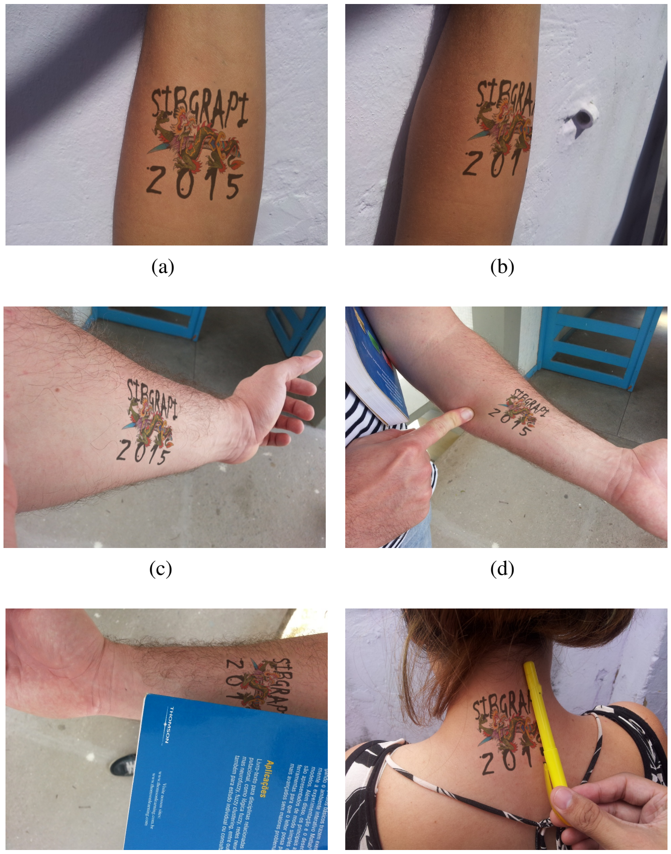

(a) (b) (c) (d) (e) (f)

Fig. 12. Some of the results of the photo dataset tests. (a) Frontal image. (b) Side image and a column of markers occluded. (c) Perspective shot. (d) Some degree of distortion is present in this image. (e) Hairy skin with a book occluding a column of markers. (f) Also shows the occlusion of a entire column of markers.

4 central markers were identified in 333 images (89.5% of accuracy). For peripheral markers detection, there were 3996 circular blobs to check against, of which 3808 were visible. Our system could identify correctly 3753 (98.5% of accuracy).

V. Conclusion

The augmented reality system for tattoo visualization described is able to overlay a virtual tattoo over a body part of an individual, considering shape, perspective, deformations and occlusions of the skin surface. To do so, a grid of markers is drawn over the skin and such markers are detected to obtain a mesh that enable appropriate tattoo rendering and overlaying. Markers detection procedures provide high accuracy rates that allow adequate mesh construction in the great majority of the images used for system evaluation. Skin segmentation and markers removal are able to generate a suitable final image with a virtual tattoo inscripted in the skin.

An important characteristic of our system is its robustness concerning the several possible operation conditions. The system can deal with different lighting conditions, background content, skin tones, drawing quality of the markers in the skin, and possible occlusions of skin and markers. This flexibility in operation conditions imposes difficulties in system design and achievable results, as seen throughout this paper. Nevertheless, the results show that, even under uncontrolled and unprepared environment conditions, the system demonstrated good accuracy and visual quality in most images.

From another perspective, this project anticipates what can be considered a radical modification of cognitive and semiotic role of skin as information processing and language niche. Its development and proliferation can quickly and easily change what we know as "embodied communication". The skin, and its "new" layout and interface, allow the exploration of new communication patterns. More speculatively, new ways of interacting relations should be established by creating new patterns of communication, associated with the use of epithelial AR devices (augmented tattoos).

Acknowledgment

The authors would like to thank FAPESB for the financial support of this project.

References

[1] R. T. Azuma *et al.*, "A survey of augmented reality," *Presence*, vol. 6, no. 4, pp. 355–385, 1997.

[2] B. Furht, *Handbook of augmented reality*. Springer Science & Business Media, 2011.

[3] S. Wohlrab, J. Stahl, and P. M. Kappeler, "Modifying the body: Motivations for getting tattooed and pierced," *Body image*, vol. 4, no. 1, pp. 87–95, 2007.

[4] M. Faille and J. Edmiston. (2013) Graphic: The tattoo industry. [Online]. Available: http://news.nationalpost.com/news/graphics/graphic-the-tattoo-industry

[5] B. Bitarello, H. Fuks, and J. Queiroz, "New technologies for dynamic tattoo art," in *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. ACM, 2011, pp. 313–316.

[6] B. Dybwad. (2010) Augmented reality tattoo makes your skin come alive. [Online]. Available: http://mashable.com/2010/02/17/augmented-reality-tattoo/

[7] S. Nicolau, X. Pennec, L. Soler, and N. Ayache, "A complete augmented reality guidance system for liver punctures: First clinical evaluation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*. Springer, 2005, pp. 539–547.

[8] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[9] R. C. Gonzalez and R. E. Woods, "Digital image processing," 2002.

[10] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[11] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.