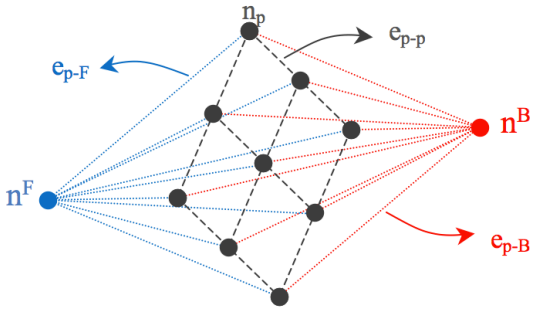
Summary of the response to reviewers' comments

Reviewer (R): Figure 2, page 2: There are two archs in the graph labelled e_{p-p}. The blue one should be e_{p-F}

Author (A): The edge notation was corrected in Figure 2. Here below the new figure that has been included:



R: Does your Equation 5 corresponds to equation 1 of reference [1]? If so, the notation is different. Add a sentence indicating the correspondance and the change in notation, since it is quite likely that anyone reading your paper reads (or have read) the reference [1], too.

A: The correspondence between the notations used in the two papers was introduced (the following text being introduced below equation 5):

This leads to two complementary, foreground and background weight images. For each pixel x, the geodesic distances $d^F(x)$ and $d^B(x)$ to the F and B scribbles are then computed conditionally to the W^F and W^B weight maps, respectively, as described in the following equation (which corresponds to the concatenations of the first two equations in **[1]**):

$$\forall\, l \in \{F,\, B\}, \qquad d^l(x) = \min_{s \in \{l\}} \min_{\gamma \in \Gamma(x,s)} \sum_{p \in \gamma} |\nabla W^l(p) \cdot \dot{\gamma}(p)|\,, \tag{1}$$

where:
– $\Gamma(x, s)$ denotes the set of all possible paths γ between pixels x and s ($C_{s,t}$ in **[1]**);
– $\dot{\gamma}(p)$ represents the path's tangent vector in point p;
– $\{l\}$ denotes the set of scribble pixels s associated to label l ($\{s \in \Omega_c : \text{label}(s) = l_i\}$ in **[1]**);
– d^F and d^B are the foreground and background geodesic distance maps (d_i in **[1]**).

R: Lines 49-52 of page 4 indicates a problem that appears due to the nature of your method and of the compressed images but the reader doesn't know yet why this problem arises (which is quite surprising). It is fully explained in a further section but from now on it is better to indicate that "Nevertheless, the use of a GMM instead of a single Gaussian provokes a problem in compressed images, due to the uniformity artifically introduced into each image block by the compression algorithms. This will be explained later, together with the solution we propose to overcome this limitation".

A: The paragraph was modified and now the source of the problem is presented as follows:

We notably show that the direct extension from a single Gaussian model to a GMM leads, in the case of compressed images, to block artifacts that can significantly degrade the segmentation results. This issue is due to the uniformity artificially introduced into each image block by the compression algorithms. This will be explained later (Section III.2), together with the solution we propose to overcome this limitation (Section III.3).

R: Lines 33-37 of page 5 needs to be ellaborated; explain a little bit if the two steps (EM and estimation of the the number of GMM) are done in a single step or in two steps. Have you used the "clust" software indicated in reference [17]? If so, say it because the software (at least according to its manual) seems to estimate both, the parameters and the number of components. Otherwise, indicate exactly how and using which software the estimation has been done.

A: The corresponding clarifications were included, *cf.* the following text:

Starting from each of the two sets of foreground and background scribbles, the corresponding GMM's parameters (*i.e.*, ω^i, Σ_i, μ_i) are estimated using the iterative Expectation-Maximization (EM) algorithm **[14]**, **[15]**. The number of GMM components (N_{GMM}) is adaptively estimated using the Rissanen criterion **[16]**. For this purpose, we have used the software described in **[17]**, which allows to estimate simultaneously the GMM model parameters and the number of components. The resulting foreground and background probability densities are respectively denoted by g^F and g^B.

R: The subfigures d and g of figure 4 show the centers of the found components. There are exactly five components for each class, but this is casual, since in other examples (page 8) there are 9 and 7 respectively. This is not surprising since they are estimated automatically but state it so that the reader can see this clearly.

A: The following explanatory text has been added:

Let us also note that in this example, an equal number of 5 GMM components has been determined for both foreground and background. However, this is casual since in the general case, different numbers of GMM components are determined with the algorithm in **[17]** (such as the examples illustrated in Figures 6 and 7).

R: Lines 4-6 of page 7 should relate the subfigures with the corresponding equations. I assume that images (b) and (d) represent the value of equation (4) for each pixel with l respectively equal to F and B and that images (c) and (e) represent the (geodesic?) distance of equation (5) for l equals F and B. Explicitly indicating this will help the reader.

A: The correspondence between the figures and the equations was included.

This leads to two complementary, foreground and background weight images. For each pixel x, the geodesic distances $d^F(x)$ and $d^B(x)$ to the F and B scribbles are then computed conditionally to the W^F and W^B weight maps, respectively, as described in the following equation (which corresponds to the concatenations of the first two equations in [1]):

$$\forall\, l \in \{F,\ B\}, \qquad d^l(x) = \min_{s \in \{l\}} \min_{\gamma \in \Gamma(x,s)} \sum_{p \in \gamma} |\nabla W^l(p) \cdot \dot{\gamma}(p)|\,, \tag{1}$$

where:
– $\Gamma(x, s)$ denotes the set of all possible paths γ between pixels x and s ($C_{s,t}$ in [1]);
– $\dot{\gamma}(p)$ represents the path's tangent vector in point p;
– $\{l\}$ denotes the set of scribble pixels s associated to label l ($\{s \in \Omega_c : \text{label}(s) = l_i\}$ in [1]);
– d^F and d^B are the foreground and background geodesic distance maps (d_i in [1]).

R: The example of the flowers in figure 5 shows a small problem: the petaloid in the lower right corner is missed (assigned to the background). Would another scribble on it have solved this? If so, you could add one figure with the original containing the new scribbles and the segmentation and comment on this problem that may occasionally arise.

A: The figure and the corresponding explanation have been introduced.

Let us also note that the small petal in the lower right corner has been missed. In order to recover such a detail, we have reiterated the segmentation process by adding a new foreground scribble (Figure 6a). The corresponding segmentation result is presented in Figure 6b.

a. **b.**

Figure 6: Refining the segmentation result. **a.** the input image and the labels: an additional foreground scribble has been introduced on the lower right petal; **b.** the obtained segmented object.

We can observe that in this case the petal has been recovered. This illustrates the possibility of iteratively refining the scribble specification in order to enhance the obtained segmentations.

R: The solution provided to the appearance of degenerate distributions (equation (11)) is too ad-hoc, and certainly the weakest point of the paper, but I couldn't find a better way to solve a problem that is not due to you but to artifacts introduced by others. May be getting more points to estimate the components or artifically introducing noise in the date before estimation could alleviate but it is not to be done now. Make a comment in "Future work".

A: We have introduced new perspectives of future work which include the ideas suggested by the reviewer. Here is the last paragraph included in the Conclusion and perspectives section:

We also intend to further enlarge the set of pixels considered for GMM estimation by including any pixel which has very high likelihood of belonging to the corresponding label. This may allow overcoming the compression artifacts. Another solution for this issue could be to artificially introduce noise in the image before the estimation process.

R: In page 14 equation (12) introduces another ad-hoc correction. It may be acceptable as long as the pixels affected by the problem are not too many. Indicate which proportion of the pixels in example 15 are below the stablished threshold \epsilon.

A: A new paragraph was introduced in order to explain what determines the amount of pixels affected by the proposed correction. The percentage of such pixels for the example in Figure 15 was also specified:

We have to mention that the amount of pixels affected by this correction depends on the complexity of the image as well as on the set of labeled pixels. For example, in the case illustrated in **Figure 16**, less than 0.5% of the pixels have probabilities below the established threshold.

R: Page 14, lines 50 to 52: I think it should be better to choose parameter N not to a fixed value (2000) but to a proportion of the total number of pixels in the image, or even better, to a fixed proportion for each number of marked scribbles (smaller for more scribbles).

A: We have introduced the following explanation:

Parameter N (cf. Section III.1) was empirically set to 2000 (i.e., each foreground/background scribble pixel set is attempted to be extended to at least 2000 pixels). Let us note that the number of considered pixels can be also computed with respect to the size of the image or to the amount of labeled pixels. However, the size of this set of pixels has to be established with respect to several aspects:

– a set which is too small can not ensure reliable GMM computation. The minimum size of this set depends on the maximum number of GMM components N_{GMM};

– the time required for the GMM estimation depends on the size of the dataset;

– only those pixels which are neighbors to the scribbles can be considered as belonging to the same label and therefore added to the initial set.

Therefore, we have preferred a fixed size approach, which corresponds to defining a target minimum size N. Setting N to a value of 2000 ensures scribble sets large enough for reliably estimating GMMs with up to 10 components.

R: Page 14, lines 54 to 59: indicate which software have you used, particularly which language (it is very important for final speed). If you have done the software yourselves, will it be available? Under which kind of license?

A: The corresponding specifications have been introduced.

Concerning the availability of the software, this issue is subject to the valorization policy of our Institute. However, at longer term, the set-up of a semi-interactive segmentation platform to be remotely accessed on the web can be envisaged.

R: Finally with respect to English usage I have not found problems but I am not an English native speaker. Please, follow the advice of the other reviewer if he/she provides any. I probably would say differently these things but don't change if you don't consider appropriate:

- *Page 11, line 59: ... GMM components can present a high probability to belong to a background GMM...*
  ... GMM components can nevertheless present a higher probability of belonging to a background GMM component...
- *Page 13, line 7: The straight contour...*
  The straight vertical contour...
- *Page 13, line 30: component should not be severely denatured. In the same, it should make ...*
  component should not be severely denatured but at the same time it should make ...
- *Page 13, line 32: we have experienced various...*
  we have performed experiments using various...
- *Page 17, line 17: , and which is insensitive to small...*
  , and is therefore insensitive to small...

A: The corresponding phrases were reformulated accordingly to the reviewer's suggestions.

Scribble-based Object Segmentation with Modified Gaussian Mixture Models

ABSTRACT

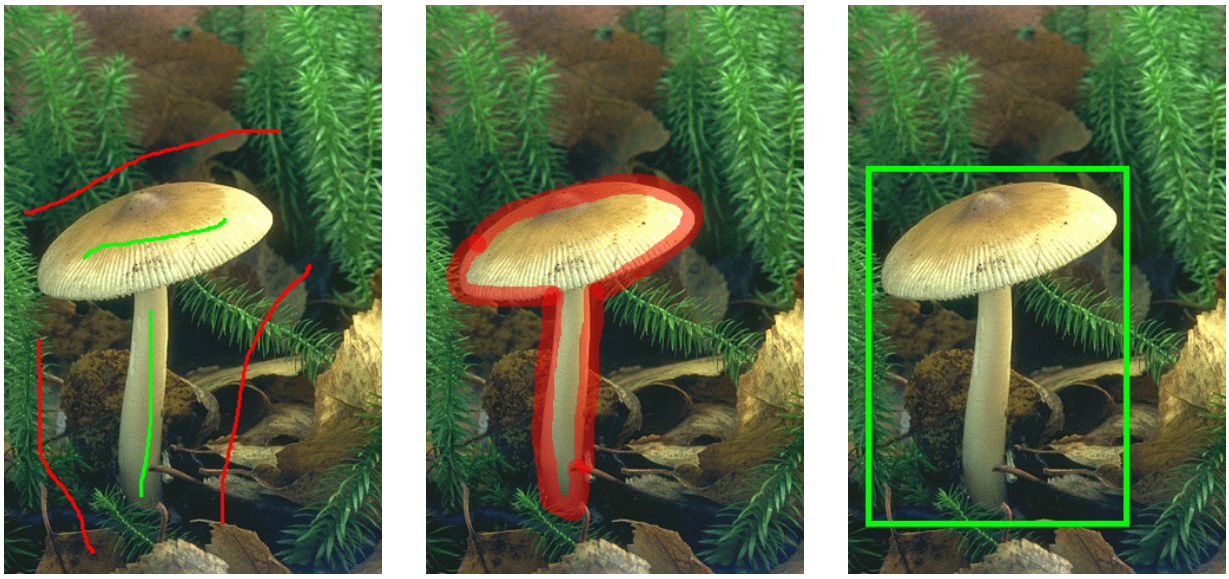
In this paper, we present an interactive segmentation method, designed to help the user to extract an object of interest from an image. The proposed approach adopts the scribble-based segmentation paradigm. The user interaction consists of specifying a set of lines, corresponding to both foreground and background scribbles. The segmentation process is based on color distributions, estimated with Gaussian Mixture Models (GMM). We show that such a technique presents some limitations when dealing with compressed images, even for relatively high quality compression factors: in this case, blocking artifacts may degrade the segmentation results. In order to overcome such a drawback, a modified GMM model, which re-shapes the Gaussian mixture based on the eigenvalues of the GMM components, is proposed. The experimental evaluation, carried out on a corpus of various images with different characteristics and textures, demonstrates the superiority of the modified GMM model which is able to appropriately take into account compression artifacts.

Key words: scribble-based interactive image segmentation; foreground extraction; Gaussian mixture model;

I. INTRODUCTION

This paper tackles the issue of interactive, semi-automatic image segmentation. The objective is to assist the user to extract meaningful objects of interest from a given image, whatever their complexity in terms of color, texture and shape characteristics, while minimizing the required human interaction. Let us underline that, in order to be able to design a general public tool and to facilitate its adoption in industrial applications, the interaction process should not involve any specific knowledge or abilities from the user.

In recent years, a scribble-based interactive segmentation paradigm has emerged [1], [2], [3] [4]. The principle consists of interactively specifying a set of scribbles, marking both the object of interest and the background. Such scribble may be arbitrary lines or free-form curves [1], [2], [3] [4] (Figure 1a), rough object boundaries [5] (Figure 1b) or bounding rectangles [6] (Figure 1c).



a. **b.** **c.**

Figure 1: Various scribbles encountered in the literature: **a.** free-form curves; **b.** rough object boundaries; **c.** bounding rectangles.

Nevertheless, when the interaction required from the user is too low, the lack of information has to be compensated by some additional *a priori* knowledge, like in [7]. Here, the user is asked to specify the center of the convex star shape (*cf.* section II) which is then used to control the segmentation process. However, such a notion is often quite difficult to deal with for non-expert users.

In this paper, we present a novel segmentation method suited for both expert and non-advised users. The human interaction is limited to sketching some line segments over the desired object as well as over some parts of the background. The main advantage of the proposed method is its ability to perform well on both compressed and uncompressed images. This is particularly useful since most of the auto-created image content today is available in a compressed form (*e.g.*, JPEG/JPEG2000 for most of the existing commercial cameras).

The remainder of this paper is organized as follows. In section II, we present an overview of the state of the art scribble-based segmentation techniques. The proposed method is then detailed in section III. The experimental evaluation, carried out on a corpus with ground truth and including various objects, is presented in Section IV. Finally, Section V concludes the paper and opens some perspectives of future work.

II. Related Work

Among one of the first and in the same time one of the most popular interactive segmentation approaches is *Graph Cut* technique introduced in [3]. Here, the user marks some more or less thick scribbles on the image in order to specify the foreground F and background B components. The labeled pixels are used to approximate the image intensity distribution of the F and B components. In order to perform the segmentation, a weighted graph is generated (Figure 2). The graph contains one node n_p for each pixel p of the image and two terminal nodes (n^F, n^B) that represent the two possible labels $l \in \{F, B\}$. Each node n_p is connected by an edge to the set of its 8 neighbors and to the two terminals nodes ($e_{p\text{-}F}$ and $e_{p\text{-}B}$ edges). The weight of an edge that connects a pixel with a terminal node ($e_{p\text{-}F,B}$) denotes the probability that the considered pixel belongs to the corresponding F or B plane. The weight $w(p,q)$ of an edge that connects two neighboring pixels p and q is given by an exponential function that takes into account both the similarity between their intensities (I_p, I_q) and the Euclidian distance $dist(p,q)$ between the considered pixels, as defined in the following equation:

$$w(p,q) = exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}. \quad (2)$$

Thus, two similar pixels will have a strong connection, expressed by a high weight of the edge. Finally, the segmentation is performed by searching for the cut of minimal cost [11] that partitions the graph into two components, one containing the n^F node and the other one the n^B node.

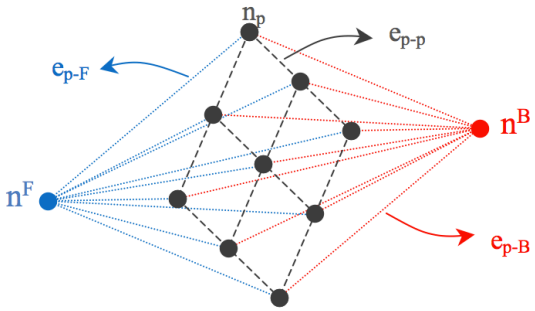


Figure 2: Graph representation exploited in the Graph Cut approach.

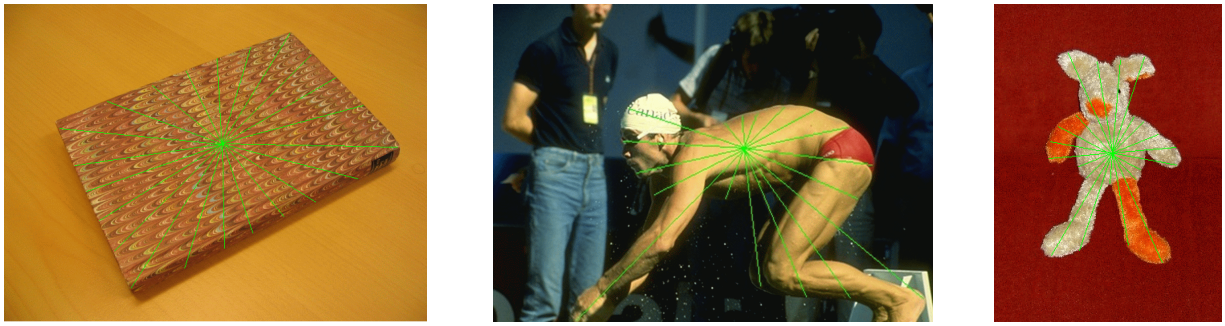
The *Graph Cut* approach has been subject to numerous extensions. Among them, let us mention the *GrabCut* method, introduced in [6]. The human interaction is here reduced to a rough specification of a bounding rectangle, placed around the object of interest. The pixels located inside the rectangle are considered as being part of the foreground, while the external pixels are labeled as background. Two Gaussian Mixture Models (GMM) [12] are generated: one for the foreground (F) and one for the background (B) component. Further, an iterative process is applied. Each F and B pixel is assigned to the most probable cluster from the corresponding GMM. Then, each component of the two GMMs is re-estimated according to the updated pixel assignment. In the same time, the weights of the graph edges are re-calculated accordingly and the minimum cut segmentation is performed. The iterative process continues using the resulting partition of the F and B pixels until convergence.

Another extension of the *Graph Cut* approach is proposed in [7]. Here, some *a priori* information is introduced in the segmentation process. Thus, the authors propose to consider solely objects that can be represented by star-convex shapes [13]. Therefore, the ambiguity is reduced by excluding all the shapes that violate this assumption.

Let us recall that a shape S is called star-convex if there exists a point c in S such that for any arbitrary point x within S the line segment from c to x is completely included within the shape. The star-convexity condition can be re-formulated in the discrete domain with respect to neighboring pixel as follows. Let us consider S a star-convex shape with the center c and x a pixel within S. If y is a neighbor of x that belongs to the (x,c) segment, than y should also belong to S. This observation makes it possible to introduce an additional, star-convex shape constraint in the

Graph Cut process, which consists of forbidding cuts along the (*x,c*) segment.

Compared to *Graph Cut* and *GrabCut*, the *Star-Convexity* approach is significantly less demanding in terms of human interaction, which represents its main advantage. Here, solely the center of the star-convex shape needs to be specified by the user. As no background pixels are marked, the border of the image is used as background label. However, the star-shape constraint is not appropriate for all the objects that can be encountered in practice, which are not star-shaped (Figure 3).



a. **b.** **c.**

Figure 3: Star-shape condition: **a.** example of star-shaped object; **b.&c.** example of shapes which do not fulfill the star-shape condition.

This drawback may be overcome using multiple stars and geodesic star-convexity, as proposed in [4]. Here, the segment that connects a given pixel x and the center c is replaced by the shortest geodesic path $\Gamma(x,c)$ between x and c. A shape is geodesic star-convex (*GSC*) if the path $\Gamma(x,c)$ lies inside the shape. Authors define the elementary distance $d(p,q)$ between two neighbor pixels p and q as a function of the Euclidian distance $d_E(p,q)$ and of the image gradient between the two pixels, as described in the following equation:

$$d(p,q) = \sqrt{(1-\gamma_g)d_E(p,q)^2 + \gamma_g \|\nabla I(p)\|^2}. \quad (3)$$

where the parameter γ_g weights the Euclidian distance versus the geodesic term.

Moreover, in order to cover a larger range of shapes, the star-convexity is extended to multiple stars. Here, the center c is replaced by a set of centers $\{c_i\}$. For each pixel x, only the closest center is considered $c_{i(x)}$. Thus, a shape S is geodesic multiple-star-convex if for any x within S, the path $\Gamma(x,c_{i(x)})$ lies entirely inside the shape. As the choice of the centers $\{c_i\}$ is a difficult task, authors propose to use the foreground labeled pixels as multiple-star centers. The experimental evaluation presented in [4] shows that *Geodesic Star-Convexity* algorithm outperforms *Star-Convexity*, *GraphCuts* and *DistanceCut* methods.

The powerful geodesic distance principle is also adopted, in a different manner in [1]. Here, authors present a semi-automatic segmentation approach which uses the concept of *adaptive weighted distances*. The algorithm requires two sets of labeled pixels, one for the foreground *(F)* and the other for the background *(B)*, which are roughly scribbled (*i.e.*, free-form curves) by the user in an interactive manner. Each image is described by a combination of $N_C = 19$ channels consisting of the luminance/chrominance components in the Y, C_b, C_r color space, as well as the output of 16 Gabor filters [8] (with 4 scales and 4 orientations) applied on the Y channel and aiming at describing the texture information. Based on the values of the labeled pixels, the probability density functions (PDF_i^l) are approximated by a Gaussian distribution for each channel i and for each foreground/background label $l \in \{F, B\}$. As the channels that better discriminate between planes are not the same in all cases, a weighting coefficient ω^i is computed for each channel. The value of this coefficient depends on the probability to assign a wrong label to a given pixel, according to the PDF of the considered channel. Finally, the global likelihood $P^l(x)$ of each pixel x to belong to the foreground ($l = F$) and to the background ($l = B$) is computed as described by the following equation:

$$\forall\, l \in \{F,\ B\}, \quad P^l(x) = \sum_{i=1}^{N_c} \omega^i \frac{\mathrm{PDF}_i^l(X_x^i)}{\mathrm{PDF}_i^F(X_x^i) + \mathrm{PDF}_i^B(X_x^i)}. \tag{4}$$

where:
– $\{X_x^i\}$ are the channel components of the pixel x;
– N_C denotes the number of channels (19, in this case).

A weight $W^l(x)$ is associated to each pixel x and for each label l:

$$\forall\, l \in \{F,\ B\}, \quad W^l(x) = 1 - P^l(x). \tag{5}$$

This leads to two complementary, foreground and background weight images. For each pixel x, the geodesic distances $d^F(x)$ and $d^B(x)$ to the F and B scribbles are then computed conditionally to the W^F and W^B weight maps, respectively, as described in the following equation (which corresponds to the concatenations of the first two equations in [1]):

$$\forall\, l \in \{F,\ B\}, \quad d^l(x) = \min_{s\,\in\,\{l\}} \min_{\gamma\in\,\Gamma(x,s)} \sum_{p\in\gamma} |\nabla W^l(p) \cdot \dot{\gamma}(p)|\,, \tag{6}$$

where:
– $\Gamma(x,s)$ denotes the set of all possible paths γ between pixels x and s ($C_{s,t}$ in [1]);
– $\dot{\gamma}(p)$ represents the path's tangent vector in point p;
– $\{l\}$ denotes the set of scribble pixels s associated to label l ($\{s \in \Omega_c : \mathrm{label}(s) = l_i\}$ in [1]);
– d^F and d^B are the foreground and background geodesic distance maps (d_i in [1]).

The segmentation is finally achieved by assigning each pixel to the closest scribble pixel, in the sense of the geodesic distance in equation (5). Thus the label l_x associated to pixel x is obtained as:

$$l_x = \arg\min_{l\in\{F,\,B\}} d^l(x). \tag{7}$$

A similar method, called *DistanceCut*, is proposed in [2]. Here, the representation of the image is reduced to the 3 channels corresponding to the Luv color space [9]. The color distribution of the labeled pixels, previously assimilated as a Gaussian, is now approximated in a finer manner, with the help of the kernel density estimation technique described in [10]. Finally, the segmentation is performed in a similar manner to the method in [1], by assigning each pixel to the closest label via the foreground and background geodesic distances.

In this paper, we propose a different extension of the scribble-based segmentation method proposed in [1]. Instead of using kernel density estimation as in [2], we consider Gaussian mixture models for color distribution estimation, which makes it possible to obtain an adaptive representation, well-suited for characterizing non-uniform regions. Let us note that this has already been suggested as a perspective by the authors in [1]. However, to our very best knowledge, such an extension has not yet been presented and validated by any of the methods reported in the literature. We notably show that the direct extension from a single Gaussian model to a GMM leads, in the case of compressed images, to block artifacts that can significantly degrade the segmentation results. This issue is due to the uniformity artificially introduced into each image block by the compression algorithms. This will be explained later (Section III.2), together with the solution we propose to overcome this limitation (Section III.3).

Let us first present the GMM-based segmentation approach.

III. GMM-BASED SEGMENTATION

As in [1], the proposed method starts from two arbitrary sets of labels, marked by the user, indicating the foreground and the background parts. For the user's convenience, a straight line drawing tool is used in order to mark the labels (Figure **4**a). However, the algorithm performs as well when the labels are free-form strokes. The labeled pixels are used to estimate the color distribution of the foreground and background planes.

The proposed method extends the reference method introduced in [1]. Instead of modeling the foreground/background PDFs by a single Gaussian function, which can solely approximate roughly the color distributions, we have considered a GMM model, described in the following section.

III.1. From Gaussian PDFs to GMMs

In order to accurately model the foreground and background PDFs, two dedicated GMMs [12] are employed. In our work, we have considered the Luv color space, because of its recognized perceptual uniformity features [9].

A GMM model, denoted by $g(X)$, is by definition a weighted sum of N_{GMM} multivariate Gaussian functions:

$$g(X) = \sum_{i=1}^{N_{GMM}} \omega^i g_i(X), \tag{8}$$

where:
– X is a D-dimensional vector (in our case $D = 3$ and X stores the Luv values of a given pixel x);
– ω^i is a positive weight associated to the i^{th} component of the GMM;
– $g_i(X)$ is the i^{th} multivariate Gaussian distribution, defined as:

$$g_i(X) = \frac{1}{\sqrt[2]{(2\pi)^D}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(X-\mu_i)^T \Sigma_i^{-1}(X-\mu_i)\right\} \tag{9}$$

where Σ_i is its covariance matrix and μ_i the mean vector.

Starting from each of the two sets of foreground and background scribbles, the corresponding GMM's parameters (*i.e.*, ω^i, Σ_i, μ_i) are estimated using the iterative Expectation-Maximization (EM) algorithm [14], [15]. The number of GMM components (N_{GMM}) is adaptively estimated using the Rissanen criterion [16]. For this purpose, we have used the software described in [17], which allows to estimate simultaneously the GMM model parameters and the number of components. The resulting foreground and background probability densities are respectively denoted by g^F and g^B.

Let us mention that straight lines scribbles are highly convenient in terms of simplicity of the required user interaction but may lead to a relatively low number of pixels, insufficient to obtain a reliable GMM estimation. In order to overcome such limitation, the user-specified sets of scribbles are enlarged as follows. A parameter α^l is computed for each label $l \in \{F, B\}$ based on the total length L^l of their corresponding scribbles (*i.e.*, number of pixels included in the considered F and B sets of scribbles). Further, each set of F and B pixels is enlarged by including, for each considered pixel, a number α^l of its neighbors, which are randomly selected. The parameter α^l is computed as follows:

$$\alpha^l = \min\left(\left\lceil\frac{N}{L^l}\right\rceil, 8\right), \tag{10}$$

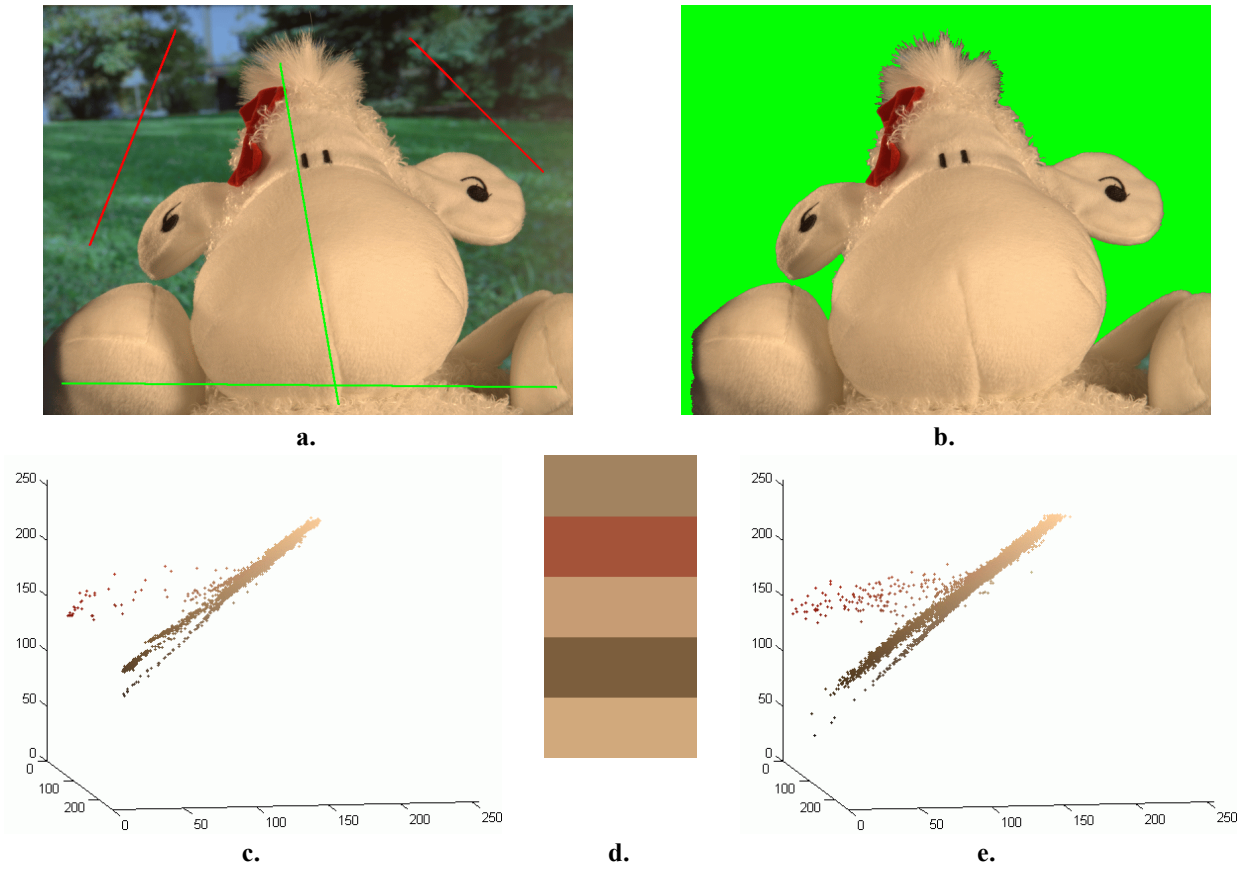
where N is a parameter that specifies the target number of pixels in each set. The maximal value of the parameter α^l is limited to 8, since a 8-connected neighborhood is considered.

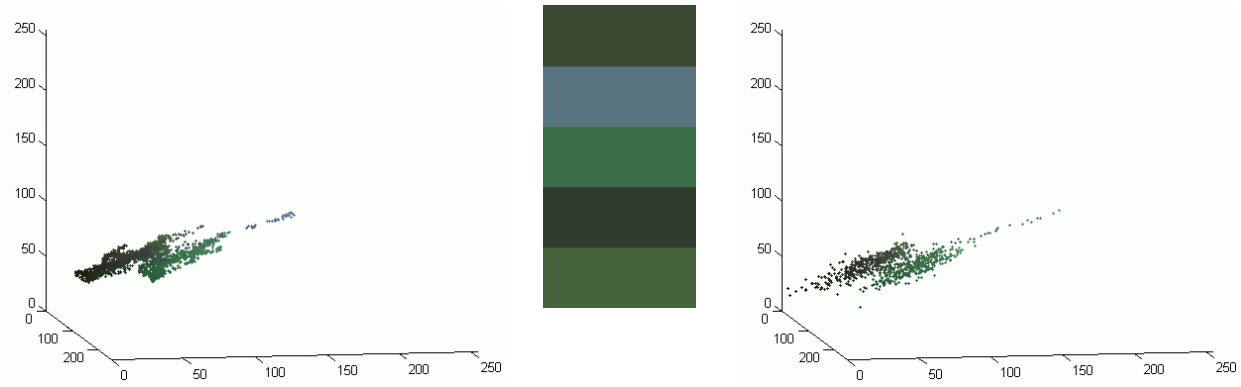
For each label $l \in \{F, B\}$, the likelihood maps $P^l(x)$ defined in equation (3) become:

$$\forall\, l \in \{F,\ B\}\,, \quad P^l(x) = \frac{g^l(X_x)}{g^F(X_x) + g^B(X_x)}. \tag{11}$$

The weight maps $W^l(x)$ are further obtained by applying the equation (4). The segmentation is then achieved with the help of geodesic distances computed conditionally to the background and foreground weights, as described in Section II (*cf.* equations 5 and 6). Finally, if several connected components are obtained, only those adjacent to a foreground scribble are retained.

The sets of N^l pixels used for foreground and background GMM estimation are illustrated in Figure **4** c and f, respectively. In order to allow the comparison between the input and the output of the GMM estimator, N^l random variables (Figure **4** e. and h.) were generated using the parameters $\{\omega i,\ \Sigma_i,\ \mu_i\}^l$ associated to each label *l*. Here, only for visualization purposes, Figures c., e, f, and h are illustrated in the RGB color space. Let us also note that in this example, an equal number of 5 GMM components has been determined for both foreground and background. However, this is casual since in the general case, different numbers of GMM components are determined with the algorithm in [17] (such as the examples illustrated in Figures 6 and 7).



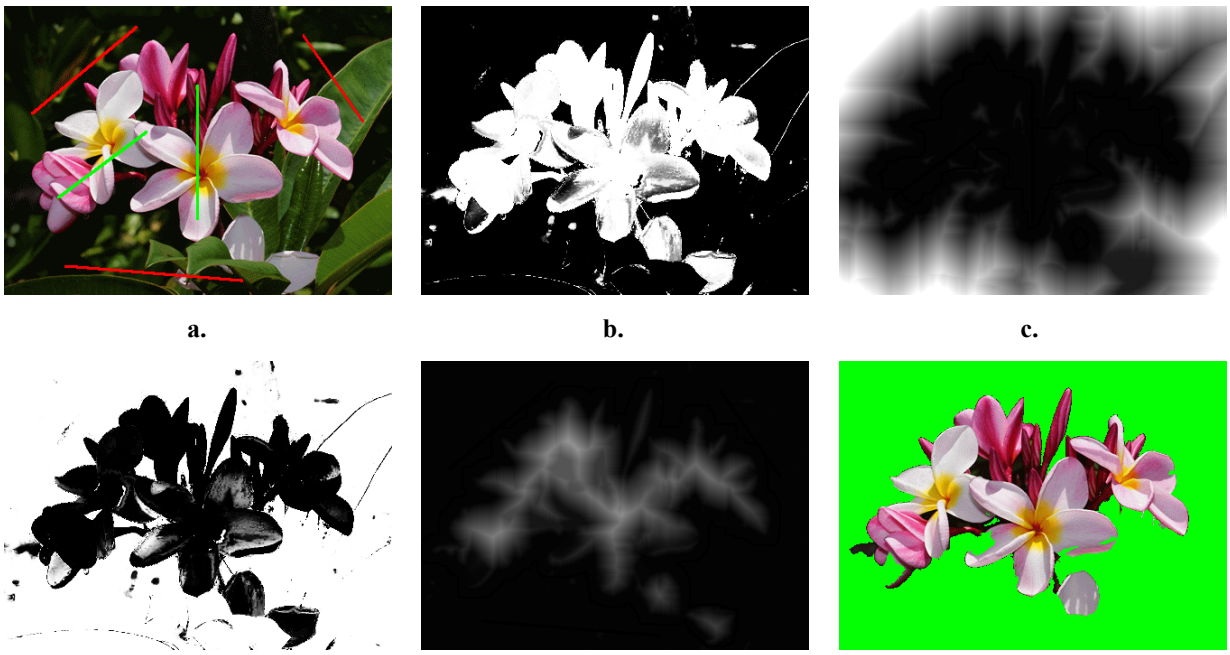


f. **g.** **h.**

Figure 4: Example of GMM-based segmentation. **a.** the user's scribbles (foreground in green and background in red). **b.** the segmentation result. **c.&f.** The set of N^l pixels used for foreground, respectively background GMM estimation; **b.&g.** the mean values of the GMM components; **e.&h.** the estimated GMM. N^l random variables were generated using the parameters $\{\omega i, \Sigma_i, \mu_i\}^l$ of the estimated GMM.

Figure 5 presents another GMM-based segmentation result. Figure 5c and e illustrate the distance (*cf.* equation 5) between each pixel and the closest foreground, respectively background scribble, with respect to the corresponding likelihood maps (Figure 5 b and d) computed with equation 4. In Figure 5 c and e low values (dark pixels) denote short distances to labeled pixels.

The result of the segmentation process is presented in Figure 5f. We can observe that, in this case, despite of the richness of the texture information present in the image, the foreground flower has been correctly segmented with the help of solely 5 scribbles (2 for the foreground and 3 for the background).



a. **b.** **c.**

d. **e.** **f.**

Figure 5: The segmentation process. **a.** The input image and the labels; **b.** the foreground likelihood map; **c.** the foreground distance map; **d.** the background likelihood map; **e.** the background distance map; **f.** the obtained segmented object.

Let us also note that the small petal in the lower right corner has been missed. In order to recover such a detail, we have reiterated the segmentation process by adding a new foreground scribble (**Figure 6**a). The corresponding segmentation result is presented in **Figure 6**b.

a. **b.**

Figure 6: Refining the segmentation result. **a.** the input image and the labels: an additional foreground scribble has been introduced on the lower right petal; **b.** the obtained segmented object.

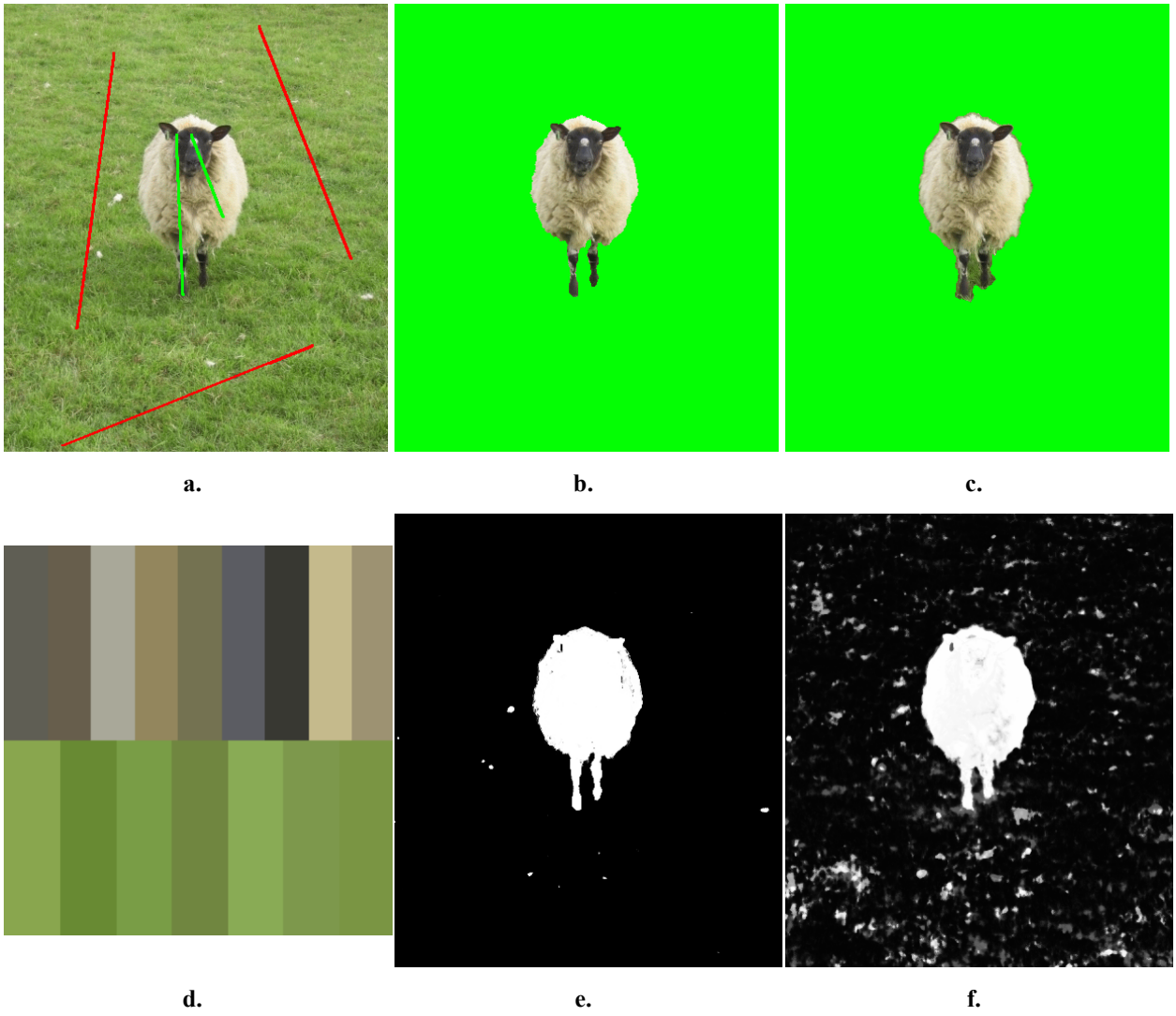
We can observe that in this case the petal has been recovered. This illustrates the possibility of iteratively refining the scribble specification in order to enhance the obtained segmentations.

Figure 7 and Figure 8 illustrate the segmentation result obtained with simple Gaussian PDFs, as in [1], and with GMM for two different examples. In the case of Gaussian PDFs, each color component (*i.e.*, Y, C_b, C_r) of each label $l \in \{F, B\}$ is modeled by a single Gaussian PDF. The likelihood $P^l(x)$ of the pixel x to belong to the label l is computed as a weighted sum of Gaussian probabilities (equation **(4)**).

Both methods provide good results for the first example (Figure 7), representing a sheep on a relatively uniform grass texture. Here, the background is almost monochromatic (green) and the corresponding GMM^B components (Figure 7d, bottom row) are very close. Thus, the background distribution can be correctly modeled with the help of simple Gaussian PDFs. In addition, the background and the foreground do not contain similar colors and can be therefore easily separated (as indicated by the likelihood maps presented in Figure 7e and Figure **7**f). However, the GMM approach performs better at the level of boundaries (where color blending effects occur), which are more accurately delineated. This behavior is also true for the region corresponding to the level of the sheep's legs, which are more accurately segmented by the GMM model, able to better take into account the shadowing effects.

The superiority of the GMM approach is even more evident in the second example (Figure 8), which represents a boy skating. The image is here acquired by night and the background presents relatively important variations in terms of level of luminance. In addition, both the foreground and the background include similar colors and present more complex distributions (Figure 8d), which cannot be correctly modeled by simple Gaussians. It can be observed on the foreground likelihood map (Figure 8f) that the object of interest has similar likelihood to some regions of the background.

However, in the case of GMMs (Figure 8e), the differences between the likelihood of foreground and background regions are more accentuated, which makes it possible to obtain a more accurate segmentation result (Figure 8b).



a. **b.** **c.**

d. **e.** **f.**

Figure 7: a. Object of interest representing a sheep on relatively homogeneous grass texture: Input image and scribbles; **b.** object segmented with GMM; **c.** object segmented with simple Gaussian PDFs; **d.** GMM components: foreground (top) and background (bottom); **e.** foreground likelihood map obtained with GMM; **f.** foreground likelihood map obtained with single Gaussian PDFs.

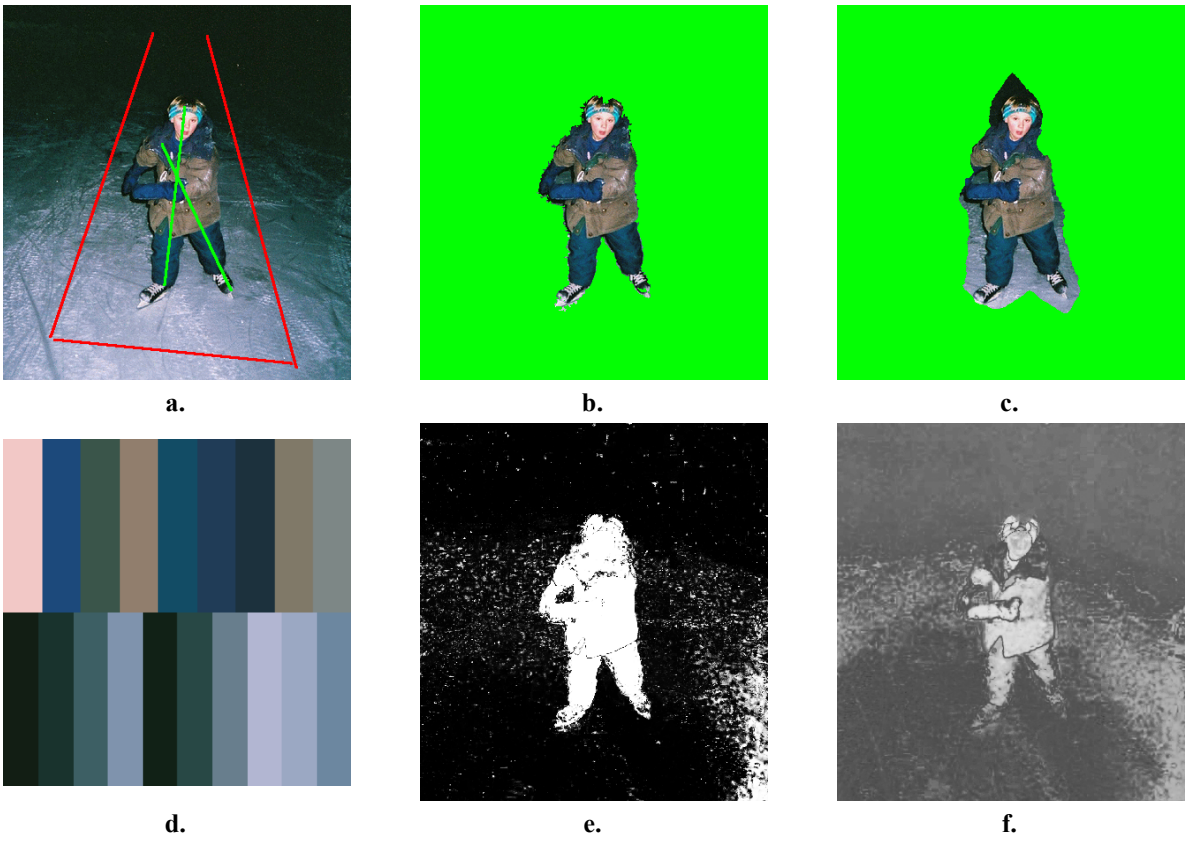


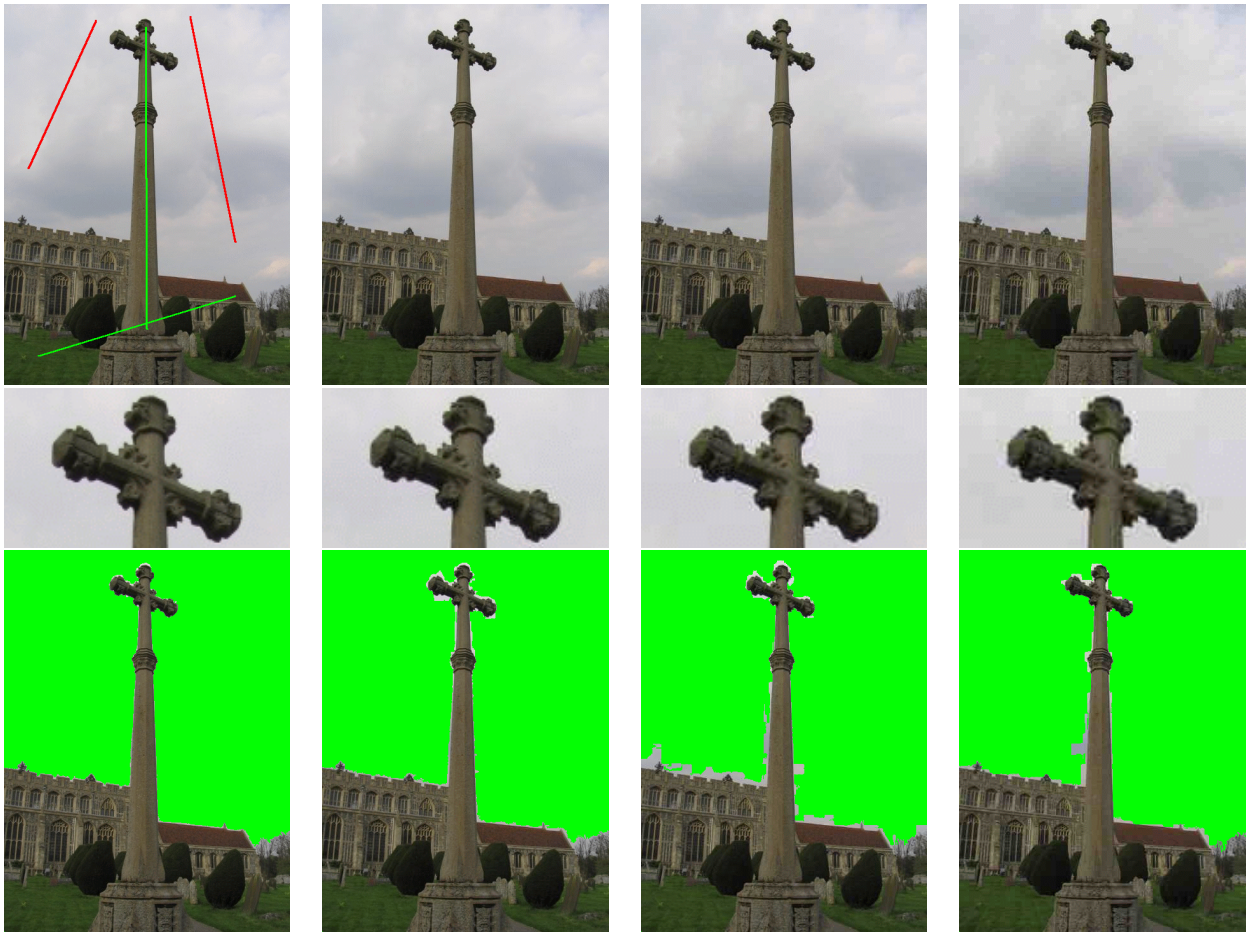
Figure 8: a. Object of interest representing a boy skating in a night scene: **a.** input image and scribbles; **b.** object segmented with GMM; **c.** object segmented with simple Gaussian PDFs; **d.** GMM components: foreground (top) and background (bottom); **e.** foreground likelihood map obtained with GMM; **f.** foreground likelihood map obtained with single Gaussian PDFs.

III.2. On the influence of the compression process

When dealing with compressed images, several artifacts, as block or contour effects, may appear. In the case of high quality compression, such artifacts are not visible on the image. However, they can significantly influence the segmentation performances. This problem is illustrated in Figure 9, which shows the segmentation results obtained by the GMM method described in the previous section for original, uncompressed images, as well as for their JPEG compressed versions (with quality factors of 75%, 50% and 25%). In our work, we have considered JPEG compression because of its high popularity and availability on existing, commercial cameras.

We can observe that the segmentation results are strongly affected by the compression process, even when the quality factors are relatively high (*e.g.*, 75%).

In order to analyze this phenomenon, Figure 10a shows the foreground likelihood map obtained for the 50%-compressed image in Figure 9. We can observe that the block artifacts are strongly affecting the likelihood map, even if they are not visible on the compressed images. This result is due to the block-based compression mechanism adopted by JPEG but also to the structure of the GMM model employed. Let us underline that such a phenomenon does not appear in the case of the original method in [1], where a unique Gaussian function is used to model the corresponding PDFs (Figure 10b and Figure 11).



100% 75% 50% 25%

Figure 9: Compression influence on the segmentation results: Top row**:** Input images with different compression levels (quality factor from 100% to 25%); Middle row: details from the compressed images: block effects are not or merely visible (for 25% compressed image); Bottom row**:** Corresponding GMM segmentation results.

a. **b.**

Figure 10: Background likelihood map for the 50% compressed image illustrated in Figure 9. **a.** obtained with GMM; **b.** obtained with Gaussian PDFs.

100% 75% 50% 25%

Figure 11: Segmentation results for uncompressed and compressed images with single Gaussian PDFs.

A finer analysis of such results shows that the block artifacts are caused by the degenerate shape of some components of the GMM model. More precisely, the presence of block effects in the likelihood map (Figure 10) is caused by the parameters of the GMM. Figure 12 illustrates the 3D Gaussian distribution of one of the background GMM components associated to the *cross* image in Figure 11. The eigenvalues of the corresponding covariance matrix are: $\lambda = [17.26 \quad 0.00125 \quad 0.00125]$.

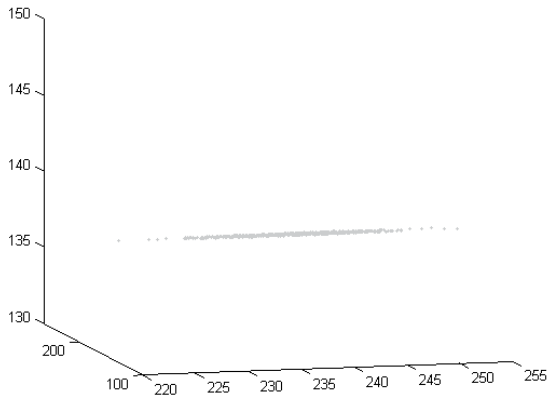


Figure 12: Example of elongated Gaussian distribution.

The last two eigenvalues indicate that the considered Gaussian component has very low variabilities on the corresponding eigen-directions, with respect to the first direction. This explains the elongated shape of the Gaussian function in Figure 12. As a consequence, the probability field is very high near the center of the Gaussian distribution and presents a strong decay when moving away in the direction of the last two eigenvectors. Thus, two colors which are close in terms of distance within the Luv color space can take very different probabilities.

This phenomenon is highlighted in Figure 13. For simplicity of representation we have considered two 1D Gaussian functions, denoted by g^B (background) and g^F (foreground) and respectively characterized by small and large variances. For the argument a considered in Figure 13, the probability $g^F(a)$ is higher than the probability $g^B(a)$. This means that the variable a is more likely to belong to the foreground than to the background. However, in terms of distances to the two Gaussian mean values, a is much closer to the foreground than to the background: $d(a, \mu^F) \ll d(a, \mu^B)$. When extrapolating this analysis to the 3D case, this means that a color which is visually very close to a foreground GMM component can nevertheless present a higher probability of belonging to a background GMM component.

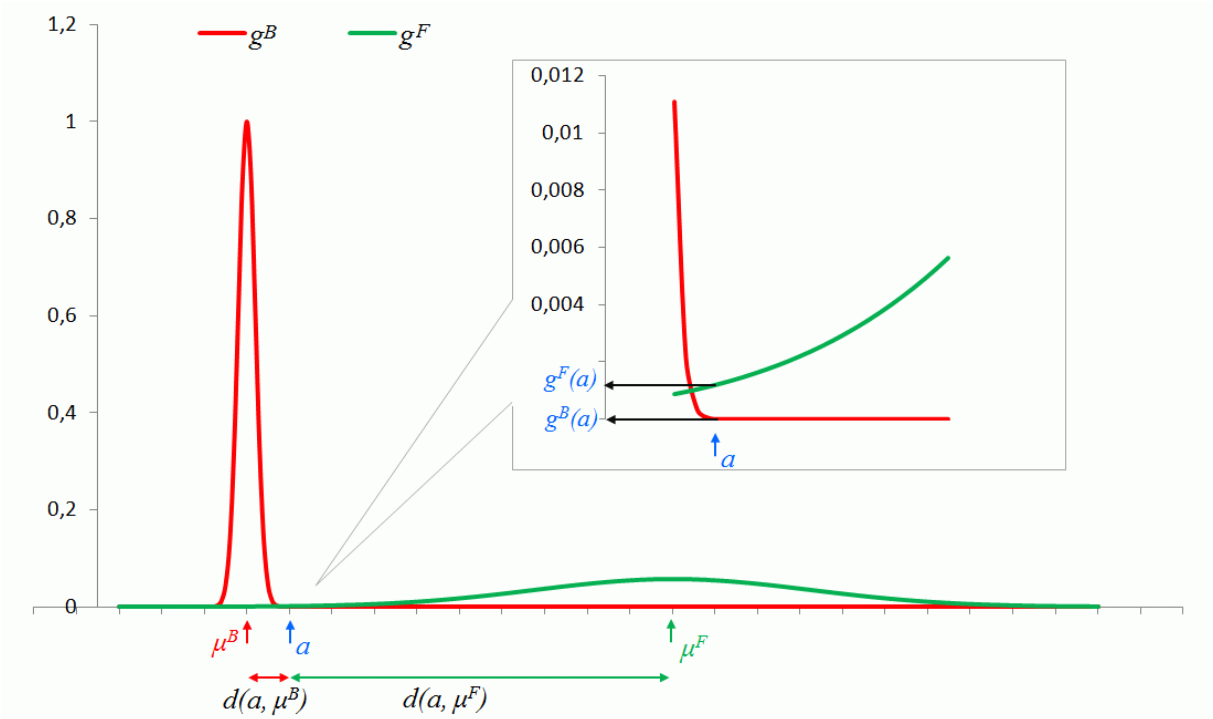
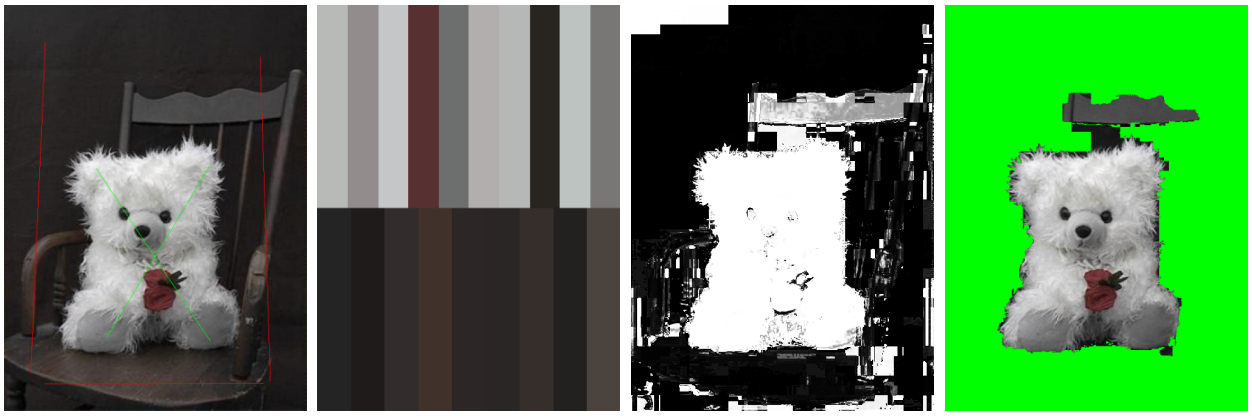


Figure 13: Example of 1D Gaussian distributions with strongly different standard deviations ($\sigma^F >> \sigma^B$). The variable a is much closer, in terms of distance, to the mean value μ^B of the Gaussian g^B than to the mean value μ^F. However, the probability $g^B(a)$ of belonging to the B distribution is smaller than the probability $g^F(a)$.

This phenomenon explains the block effect obtained in the case of block-based coding methods such as JPEG. Such compression methods transform relatively uniform regions in blocks of slightly different colors (phenomenon which is often invisible for the human eye, at least for reasonable compression quality factors). However, such small differences may lead to strong variations of the likelihood maps, for the entire blocks. This phenomenon is far from being isolated and appears frequently in practice, notably in the case of images available over Internet, which are in most cases JPEG compressed. Figure 14 illustrates a second example, where the object of interest is representing a soft toy placed on a chair.



a. **b.** **c.** **d.**

Figure 14: Compression block artifacts for a soft toy image: **a.** the input image; **b.** Gaussian components determined for foreground and background components; **c.** the foreground likelihood map; **d.** segmented object with GMM.

In this example, some regions of the wall (*e.g.*, the upper left corner, the back of the chair above the soft toy) have high likelihood to belong to the foreground, even if they are visually highly similar to the rest of the background. The segmentation result is illustrated in Figure 16e. It can be observed that several parts of the background were wrongly extracted as foreground (*e.g.*, the part of the wall above the stuffed toy). The straight vertical contour on the right of the extracted object is also a consequence of the block-based coding. The upper left corner is not detected as foreground (Figure 14d), although it has a high foreground likelihood, because only the components which are adjacent to an *F* scribble are kept.

In order to overcome this drawback we propose to slightly modify the GMM color distribution model, as described in the following section.

III.3. Modified GMMs

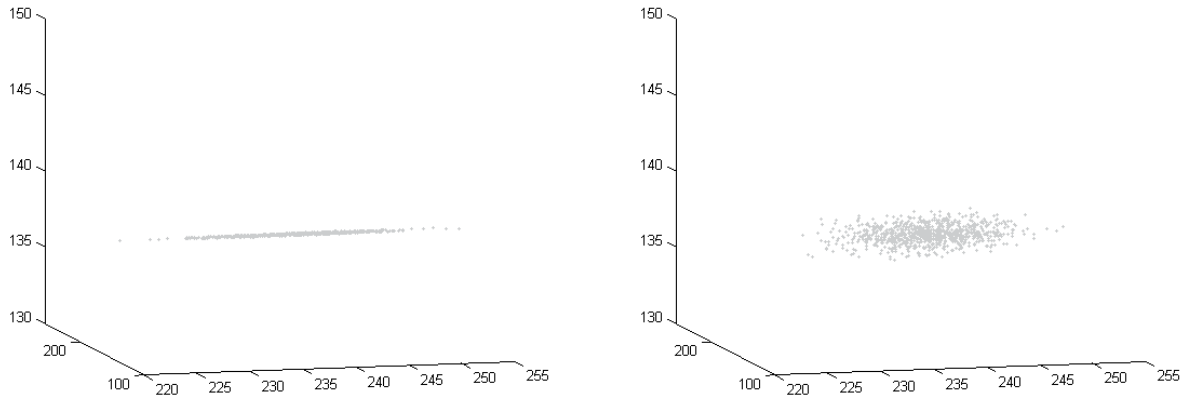
In order to attenuate the block artifacts, we propose to alter the GMM components that present low eigenvalues. Thus, the Gaussian distributions are expanded (Figure 15) by modifying the low eigenvalues λ_i of each GMM component as follows:

$$\widehat{\lambda_i} = \begin{cases} \lambda_i , & \text{if } \lambda_i > \frac{1}{\beta} \cdot \max_j \lambda_j \\ \frac{1}{\beta} \cdot \max_j \lambda_j , & \text{otherwise} \end{cases} . \quad (12)$$

where β is a parameter used to limit the maximum elongation of each GMM component.

The parameter β involved should be selected in a manner such that the overall shape of the corresponding Gaussian component should not be severely denatured but in the same time, it should make it possible to overcome the stability issues related to such degenerated Gaussians. In our work, we have performed experiments using various values of the β parameter, ranging from 10 to 100, with quite equivalent results in terms of obtained segmentations.

In this way, the variability on the corresponding eigen-directions is increased and thus the variation of the probability is decreased. Figure 15 illustrates a Gaussian distribution and its modified version in the Luv color space.



a. **b.**

Figure 15: Example of Gaussian distribution in Luv color space.
a. original Gaussian distribution; **b.** modified Gaussian distribution.

A final refinement has been introduced: we impose likelihood P^l equal to 0.5, for any color which has both the

foreground and the background probabilities g^F and g^B inferior to a given threshold ε. This makes it possible to eliminate the influence of colors with very small foreground and background probability values, which are often unreliable. In this way, pixels which are highly unlikely to belong to either foreground or background are neutralized, in terms of corresponding likelihood values.

$$P^l(X) = \begin{cases} P^l(X); & \forall l \in \{F,B\}\ g^l(X) > \varepsilon \\ 0.5; & otherwise \end{cases}. \tag{13}$$

In our work, the probability threshold ε has been set to 10^{-6}. We have to mention that the amount of pixels affected by this correction depends on the complexity of the image as well as on the set of labeled pixels. For example, in the case illustrated in Figure 16, less than 0.5% of the pixels have probabilities below the established threshold.

The modified GMM representation makes it possible to significantly reduce the block artifacts due to compression, as illustrated in Figure 16. For the same example previously presented in Figure 14, we obtain in this case an accurate segmentation result. We can observe that this time the effect of the block artifacts is significantly reduced (Figure 16b).

a. **b.** **c.**

Figure 16: Compression artifacts reduction with modified GMM. **a.** the input image; **b.** the foreground likelihood map obtained with the modified GMM; **c.** segmentation result.

IV. Experimental Evaluation

Experiments have been carried out on the database proposed in [4], which includes the GrabCut dataset [18], images from the PASCAL VOC'09 segmentation challenge [19], and 3 images from the alpha-matting dataset [20], for a total number of 151 test images, with resolutions from 284×398 to 800×618 pixels and stored in PNG, BMP and JPEG format. The retained corpus presents a large variety of objects (humans, airplanes, cars, trains, animals, plants, furnish, household items …) acquired indoor, in urban areas or in the nature. The ground truth segmentations are also provided.

Concerning the scribble specification, we have considered at most 5 line segments per image. This allows us to keep a relatively low amount of human interaction, which is an important aspect for commercial applications.

Parameter N (*cf.* Section III.1) was empirically set to 2000 (*i.e.*, each foreground/background scribble pixel set is attempted to be extended to at least 2000 pixels). Let us note that the number of considered pixels can be also computed with respect to the size of the image or to the amount of labeled pixels. However, the size of this set of pixels has to be established with respect to several aspects:

– a set which is too small can not ensure reliable GMM computation. The minimum size of this set depends on the maximum number of GMM components N_{GMM};

– the time required for the GMM estimation depends on the size of the dataset;
– only those pixels which are neighbors to the scribbles can be considered as belonging to the same label and therefore added to the initial set.

Therefore, we have preferred a fixed size approach, which corresponds to defining a target minimum size *N*. Setting *N* to a value of 2000 ensures scribble sets large enough for reliably estimating GMMs with up to 10 components.

The elongation parameter β used to modify the GMM (equation (12)) was set to 50.

The algorithm has been implemented in C++ and run on an Intel Xeon machine with 2.8GHz and 12GB RAM, under a Windows 7 platform. The segmentation process takes 2 seconds for a 320*480 pixels image and 5 seconds for a 600*450 pixels image, which ensures interactive segmentation rates. Let us note that numerous optimizations are possible, for example using LUT for computing the Gaussian probabilities involved, or exploiting GPU implementations for parallelization.

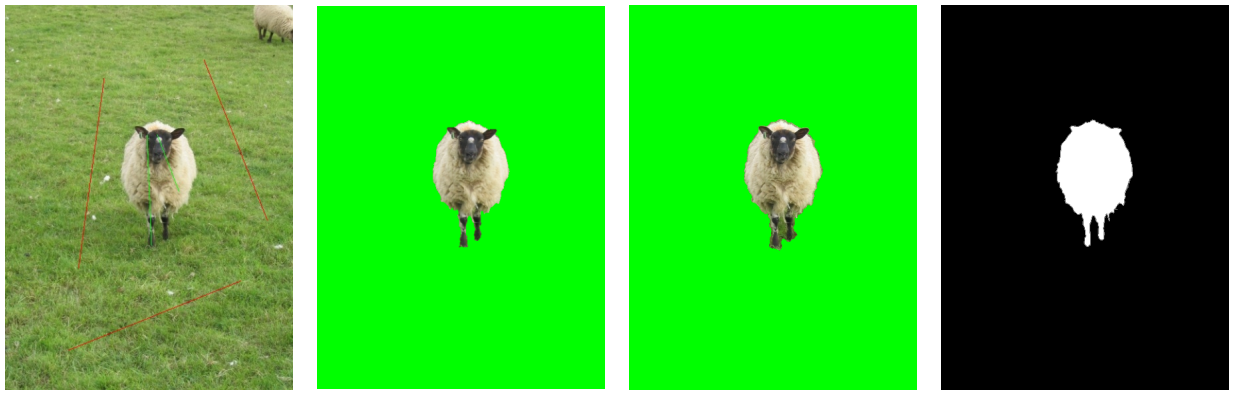
The performance measure adopted is the *overlap score* (*OS*) already used for evaluation purposes in [4], [19]. The *OS* rate measures the number of correctly and incorrectly assigned pixels and is defined as:

$$OS = \frac{N_{F|F}}{N_{F|F} + N_{F|B} + N_{B|F}} \quad . \tag{14}$$

where:
– $N_{F|F}$ is the number of foreground pixels detected as foreground,
– $N_{B|F}$ is the number of background pixels detected as foreground,
– $N_{F|B}$ is the number of foreground pixels detected as background.

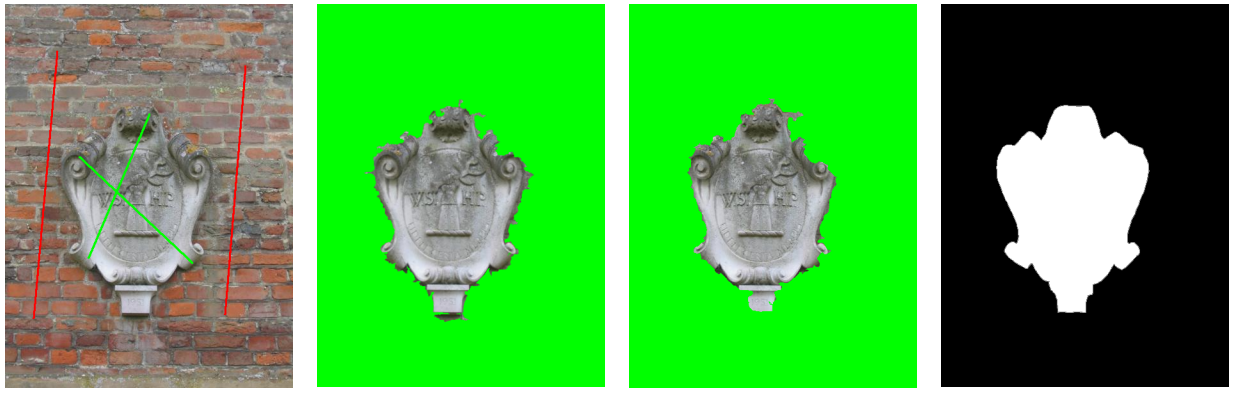
We have first analyzed the impact of using a more sophisticated color model with respect to the reference method in [1], where single Gaussians, computed marginally on each color plane, are exploited. Here, the genuine, non-modified GMM representation has been considered. The average overlap score obtained for the whole test data set with GMM is of 85.2%, while with single Gaussian PDF modeling is of only 75.6%. This clearly shows the superiority of a more realistic color distribution model as the GMM. Several examples of GMM-based segmentation results are illustrated in Figure 17 to Figure 20.

Figure 17 presents an example of a highly accurate segmentation result, where the overlap score is of 95.2%. For the same input, the reference method leads to an *OS* of 92.4%. Figure 18 illustrates a segmentation result with a score of 93.2%. Here, the few segmentation errors may be explained by the presence of similar materials (*i.e.*, cement) in foreground and background components. In the example illustrated in Figure 19, the segmentation is less precise (*OS* = 80%) because of the similarity between the black jacket of the character of interest and the dark background. However, the *OS* measure obtained for the reference method in [1] is here of only 57%. Another difficult example is presented in Figure 20. The result obtained here can be explained by the fact that the foreground GMM does not contain a white component, suited to model the horse's white spots. The absence of such component is due to the low resolution of the image (the size of the horse's bounding box is of 70×220 pixels) which lead to an insufficient number of white pixels used for GMM estimation. Also, the fact that the foreground and background components include similar colors (*e.g.*, grey) has a negative impact on the segmentation process. However, the obtained *OS* measure is of 77% for the GMM approach, which outperforms the reference method (*OS* = 62%).



a. **b.** **c.** **d.**

Figure 17: Example of segmented object. **a.** the input image and the used labels; **b.** the object segmented with the GMM-based method ($OS = 95.2\%$); **c.** the object segmented with Gaussian PDFs approach ($OS = 92.4\%$); **d.** the ground truth.



a. **b.** **c.** **d.**

Figure 18: Example of segmented object. **a.** the input image and the used labels; **b.** the object segmented with the GMM-based method ($OS = 93.2\%$); **c.** the object segmented with Gaussian PDFs approach ($OS = 90.6\%$); **d.** the ground truth.

a. **b.** **c.** **d.**

Figure 19: Example of segmented object. **a.** the input image and the used labels; **b.** the object segmented with the GMM-based method ($OS = 80\%$); **c.** the object segmented with Gaussian PDFs approach ($OS = 57\%$); **d.** the ground truth.

a. **b.** **c.** **d.**

Figure 20: Example of segmented object. **a.** the input image and the used labels; **b.** the object segmented with the GMM-based method (*OS* = 77.7%); **c.** the object segmented with Gaussian PDFs approach (*OS* = 62.7%); **d.** the ground truth.

Such results demonstrate the superiority of the GMM representation. However, the main limitation is related to the block artifacts which appear in the case of compressed images (*cf.* Section III.2, Figure 14).

In order to objectively evaluate the improvement introduced by the modification of GMM, we have compressed the images used in the first experiment (JPEG compression with quality factors of 75%, 50% and 25%). Further, we have computed the overlap score when Gaussian PDFs, GMM and modified GMM representations are employed. Table 1 summarizes the average overlap scores obtained with the three approaches at different compression levels.

Table 1: Overlap score on original and compressed images.

Quality factor: / *Method:*	**100%**	**75%**	**50%**	**25%**
Gaussian PDF	75.6	75.6	75.3	75.4
GMM	85.2	83.3	82.1	79.8
modified GMM	85.8	84.4	83.9	82.2

Let us observe that the performance of the reference method in [1] (Gaussian PDFs) is not affected by the compression. This can be explained by the fact that marginal Gaussian distributions provide a global representation, where the various colors are blended together, and is therefore insensitive to small variations of the color content as those introduced in the JPEG compression process. However, in all cases, the overlap score is inferior to the one obtained with GMM (almost 10% lower for the original images).

The overlap score of the GMM method decreases from 85.2% for original images down to 79.8% on 25%-compressed images. This loss in performance is significantly attenuated in the case of the modified GMM approach, where the OS measure goes down from 85.8% (original images) to 82.2% (25%-compressed images). Over the whole range of compression factors considered, the modified GMM representation offers a slight gain in performances: 1.1% on 75%-compressed images, 1.8% on 50%-compressed images and of 2.4% for a quality factor of 25%.

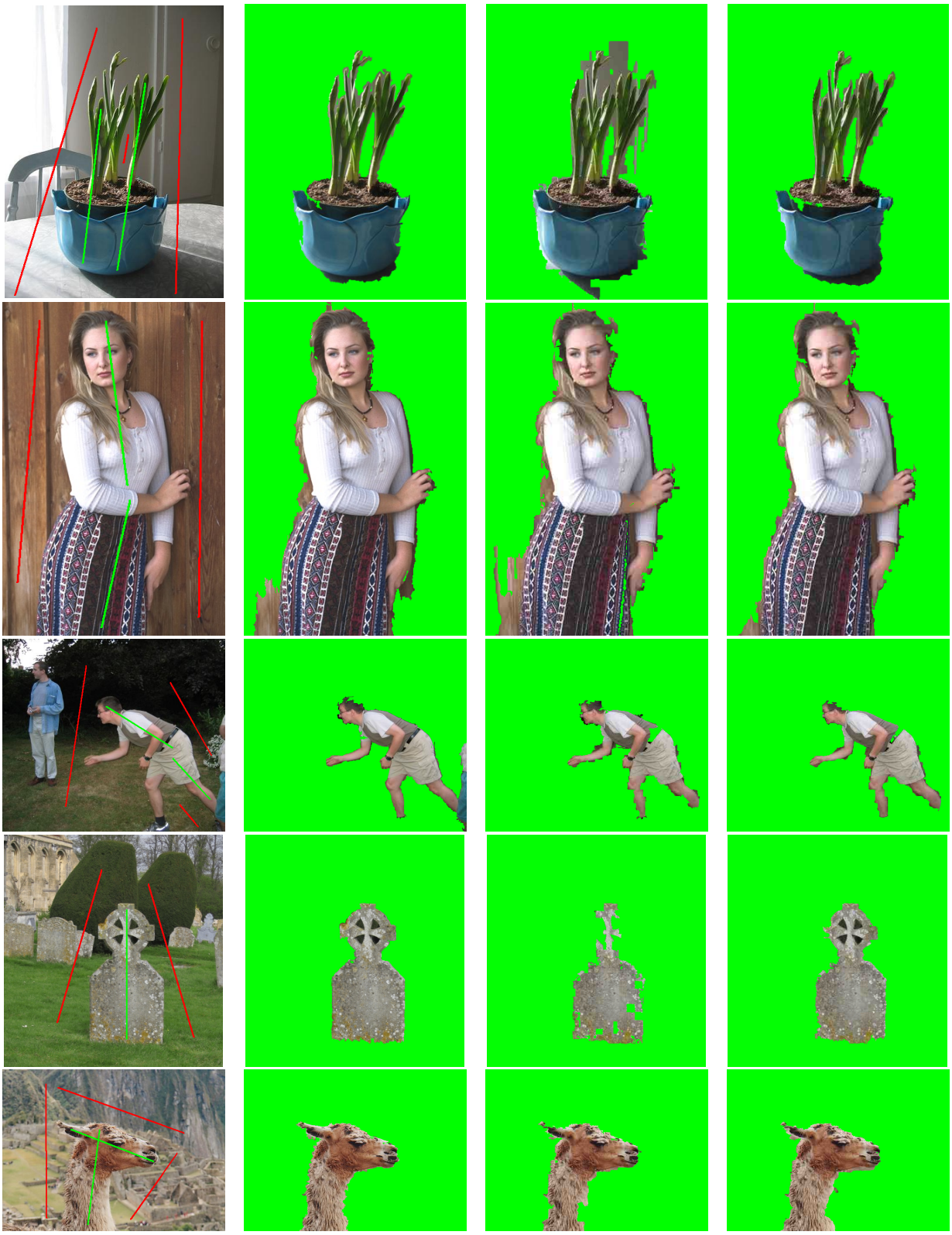
The superiority of the modified GMM model is even more manifest in terms of visual quality of the obtained segmentations. Figure 21 illustrates the segmentation results obtained with simple GMM on the original images (Figure 21b) and on the 25%-compressed images (Figure 21c) as well as those obtained with modified GMM on the 25%-compressed images (Figure 21d). We can observe that the compression artifacts mainly affect the contour regions between foreground and background, which become ragged in the case of the genuine GMM approach. In terms of overlap scores such differences are relatively low (1-2%), as the OS measure is computed globally over the object's region of support. However, the visual quality is significantly altered by such a phenomenon. The modified

GMM representation makes it possible to overcome this limitation and leads to neat borders, with results approaching those obtained on uncompressed images.

V. Conclusion and Perspectives

In this paper, we have addressed the issue of interactive object segmentation. The proposed method employs a GMM representation for both foreground and background color distribution estimation. We have observed that when dealing with compressed images, several artifacts may appear. In order to attenuate the negative impact of the compression artifacts on the segmentation process, a modified GMM representation has been proposed. We have shown that the obtained model is well-suited for dealing with compressed images. Thus, the objective experimental evaluation proposed, demonstrated that the block-based coding effect is significantly attenuated and the segmentation quality is improved.

In our future work we intend to adjust the GMM for shadowy regions. The objective here is to extrapolate the color distribution from the lighted regions to the shaded areas and inversely. A second perspective concerns the integration of a scribble refinement stage, based on an adaptive exploitation of the available contour information. We also intend to further enlarge the set of pixels considered for GMM estimation by including any pixel which has very high likelihood of belonging to the corresponding label. This may allow overcoming the compression artifacts. Another solution for this issue could be to artificially introduce noise in the image before the estimation process.



a. **b.** **c.** **d.**

Figure 21: The influence of JPEG compression: a. the input images; b. the object segmented from the initial image; c. the object segmented from the 25%-compressed image with simple GMM; d. the object segmented from the 25%-compressed image with modified GMM.

REFERENCES

[1] A. Protiere and G. Sapiro, *"Interactive image segmentation via adaptive weighted distances"*. IEEE Transactions on Image Processing, Volume 16, No. 4, pp. 1046–1057, April 2007.

[2] X. Bai and G. Sapiro, *"A geodesic framework for fast interactive image and video segmentation and matting"*, IEEE 11th International Conference on Computer Vision, pp. 1-8, 2007.

[3] Y. Boykov and M.-P. Jolly. *"Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images"*. International Conference on Computer Vision (ICCV), Volume 1, pp. 105–112, 2001.

[4] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. *"Geodesic star convexity for interactive image segmentation"*. In Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3129-3136, 2010.

[5] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, *"Interactive image segmentation using an adaptive GMMRF model"*. Proceedings of Computer Vision – ECCV 2004, Volume 3021, pp. 428–441, 2004.

[6] C. Rother, V. Kolmogorov, and A. Blake. *"Grabcut: Interactive foreground extraction using iterated graph cuts"*. ACM Transactions on Graphics, Volume 23, No. 3, pp. 309-314, 2004.

[7] O. Veksler. *"Star shape prior for graph-cut image segmentation"*. In Proceedings of the 10th European Conference on Computer Vision, Part III, pp. 454–467, 2008.

[8] S. V. Kyrki and J.K. Kamarainen. *"Simple Gabor feature space for invariant object recognition"*. Pattern Recognition Letters, Volume 25, No. 3, pp. 311-318, 2004.

[9] M. Tkalcic and J. F. Tasic, *"Color spaces: Perceptual, historical and application background"*. In Proceedings of IEEE EUROCON, Volume 1, pp. 304–308, Sep. 2003.

[10] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis. *"Improved fast Gauss transform and efficient kernel density estimation"*. Ninth IEEE International Conference on Computer Vision, pp. 664–671, 2003.

[11] Y. Boykov and V. Kolmogorov. *"An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26, No. 9, pp. 1124-1137, 2004.

[12] D. Reynolds. *"Gaussian Mixture Models"*, Encyclopedia of Biometric Recognition, 2007.

[13] C. Smith. *"A characterization of star-shaped sets"*. American Mathematical Monthly, Volume 75, No. 4, pp. 386, 1968.

[14] A.P. Dempster, N.M. Laird, and D.B. Rubin, *"Maximum likelihood from incomplete data via the EM algorithm"*, Journal of the Royal Statiscal Sociecty, Series B, Volume 39, No.1, pp.1-38, 1977.

[15] Todd K. Moon, *"The expectation-maximation algorithm"*, IEEE Signal Processing Magazine, Volume 13, No. 6, pp. 47-70, 1996.

[16] J. Rissanen, *"A Universal Prior for Integers and Estimation by Minimum Description Length"*, The Annals of Statistics, Colume 11, No. 2, pp. 416-431, 1983.

[17] C.A. Bouman, M. Shapiro, G.W. Cook, C.B. Atkins and H. Cheng. "*Cluster: An Unsupervised Algorithm for Modelling Gaussian Mixtures"*. School of Electrical Engineering, Purdue University, http://dynamo.ecn.purdue.edu/˜bouman/, 1997.

[18] GrabCut image dataset http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm

[19] M. Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman. *"The PASCAL Visual Object Classes Challenge",* 2009 (VOC2009) Results.

[20] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. *"A perceptually motivated online benchmark for image matting",* In Proc. CVPR, pages 1826–1833, 2009.