Learning and Propagation of Dominant Colors for Fast Video Segmentation

Cédric Verleysen and Christophe De Vleeschouwer

Université catholique de Louvain, ICTEAM institute,
Place du Levant 2, B-1348 Louvain-La-Neuve, Belgique.
cedric.verleysen@uclouvain.be

Abstract. Color segmentation is an essential problem in image processing. While most of the recent works focus on the segmentation of individual images, we propose to use the temporal color redundancy to segment arbitrary videos. In an initial phase, a k-medoids clustering is applied on histogram peaks observed on few frames to learn the dominant colors composing the recorded scene. In a second phase, these dominant colors are used as reference colors to speed up a color-based segmentation process and, are updated on-the-fly when the scene changes. Our evaluation first shows that the proprieties of k-medoids clustering make it well suited to learn the dominant colors. Then, the efficiency and the effectiveness of the proposed method are demonstrated and compared to standard segmentation benchmarks. This assessment reveals that our approach is more than 250 times faster than the conventional mean-shift segmentation, while preserving the segmentation accuracy.

Keywords: Segmentation, clustering, color learning, k-medoids, box-cox transform

1 Introduction

Video segmentation has always been a major topic in computer vision and multimedia. Indeed, partitioning the frames of a video into non-overlapping areas with different semantical contents has tremendous applications in data compression, tracking, augmented reality, activity or object recognition, video annotation and video retrieval. Although all these applications are very different from each other, the developed segmentation methods can be categorized into two classes.

The first class of segmentation algorithms aims at partitioning each frame of an arbitrary video, that could represent static scenes, into a complete set of non-overlapping regions. In this case, despite the strong temporal redundancy of video frames, most of the proposed methods still segment the frames individually. For example, in [9], a frame-by-frame process ($2D$) first smoothes the images using a variant of anisotropic diffusion and then merge the neighboring color pixels according to their color similarity. In [19] and [5], a frame-to-frame framework ($2D+t$) is used to associate independent segmentations of $2D$ frames and refine the segmented regions. Because image segmentation methods are often quite costly both in memory resources and in computational power [13], such approaches are unsuitable for real-time applications. Spatio-temporal segmentation methods (segmenting directly the $3D$ volume of the video) have also been recently investigated [15]. Specifically, due to its good performances on various type of images/videos, mean-shift segmentation [7] has gained considerable attention these last years. However, because of its slow convergence [6], this algorithm is also not adapted to real-time applications.

The second class of methods aims at segmenting a specific moving object in a video sequence. In this case, it is possible to take advantage of the motion analysis to segment the object. For example, Ellis and Zografos [10] use a semi-supervised appearance learning method coupled with a motion segmentation algorithm to perform on-the-fly segmentation. However, because the segmented region is determined using motion features, their fast segmentation method is restricted to moving objects, as it is the case in [17] and [16].

In this paper, we combine the strengths of both classes by proposing a color segmentation algorithm that autonomously learns the dominant colors of the recorded scene, and both propagates forward and updates this weak prior to speed-up the segmentation. More precisely, on a few frames at the beginning of the video sequence, the dominant colors composing the scene are recursively determined using a k-medoids algorithm applied on the principal mode of the images color histograms. On the next frames, these dominant colors are used as reference colors to segment the frames into regions of approximatively uniform color. The approach is especially relevant when dealing with video sequences whose color distribution does not significantly change over time. In that case, the initial training phase just needs to be run once on few frames at the beginning of the sequence, to define the dominant colors for the whole sequence. This case typically happens both when the camera viewpoint does not significantly change over time, like in surveillance contexts capturing scenes with still cameras, and when capturing a scene with low variance color distribution, as it happens in sport scenes for example. To make our method valid for arbitrary sequences, we however propose a simple approach to update the set of dominant colors when required. By restarting the learning process when a certain percentage of the pixels are not represented by the list of learnt dominant colors, we avoid regular manual corrections of the propagated segmentation cues, as done in [2] and [20]. Moreover, as it is shown in Section 3.1, because the dominant colors are determined by k-medoids clustering, the learnt colors are actually present in the image. This enable to create artistic stylizations (e.g. cartoons and paintings) without manual rectification of the colors, as opposed to [25].

This paper is organized as follows. The proposed algorithm is first described in Section 2. Section 3.1 shows that the better robustness of a k-medoids clustering makes it more adapted for the learning of dominant colors than the usual k-means clustering. Finally, in Section 3.2, we highlight the key advantage of our approach by showing that the computation time of the proposed color segmentation is significantly smaller than the one of conventional segmentation methods working without color prior, such as the mean-shift algorithm, while preserving the segmentation effectiveness.

2 Color segmentation with autonomous learning of dominant colors

This section introduces the main contribution of our work, which consists in exploiting dominant color priors to segment the video stream in a spectacular efficient, whilst effective, manner. As illustrated in Algorithm 1, the proposed approach relies on three complementary phases: the learning of dominant colors, the fast color segmentation and the update of the learnt dominant colors. Each phase is individually detailed in the rest of this section.

Algorithm 1 Fast color segmentation algorithm

Input: $videoStream$, N (amount of frames to learn a dominant color),
T_1 (thresholds the color dissimilarity) and T_2 (triggers the state change)
Output: $\{segFrame_i\}$ (the segmented frames of $videoStream$)

Initialize: $i \leftarrow 1$; $n \leftarrow 1$; $isLearning \leftarrow true$; $isStable \leftarrow false$; $D_c \leftarrow [\,]$; $D_{\text{tmp}} \leftarrow [\,]$;

Procedure:
while NOT(end of $videoStream$) **do**
  $I \leftarrow getFrame(videoStream,\ i)$;
  $segFrame_i \leftarrow [\,]$;
  if $isLearning$ OR NOT($isStable$) **then**
    $mask \leftarrow \{\mathbf{x} \mid (d(I, \mathbf{x}, D_c) > T_1)\}$;
    $D_{\text{tmp}} \leftarrow \text{concatenate}\big(D_{\text{tmp}},\ \underset{b \in \mathbf{bins}}{\text{argmax}}\ hist3D(I(mask))\big)$;
    $n \leftarrow n + 1$;
    if $n > N$ **then** Learning phase
      $D_c \leftarrow \text{concatenate}\big(D_c,\ \text{kmedoids}(D_{\text{tmp}}, k = 1)\big)$;
      $D_{\text{tmp}} \leftarrow [\,]$; $n \leftarrow 1$; $isLearning \leftarrow false$;
    end if
  end if
  if NOT($isLearning$) **then**
    ∘ Replace each pixel $\mathbf{x}$ of I with its closest (minimum Running phase
    $d(I, \mathbf{x}, D_c)$) dominant color of D_c and store in $segFrame_i$;
    ∘ Compute percentage p of pixels $\mathbf{x}$ with $d(I, \mathbf{x}, D_c) > T_1$;
    if $p > T_2$ **then**
      if $isStable$ **then**
        ∘ Remove from D_c the dominant color with the
        largest mean dissimilarity $d(I, \mathbf{x}, D_c)$;
        $isStable \leftarrow false$;
      end if Update phase
      $isLearning \leftarrow true$;
    else
      $isStable \leftarrow true$;
    end if
  end if
  $i \leftarrow i + 1$;
end while

2.1 Learning phase: determination of the dominant colors

The learning phase recursively determines the dominant colors based on sets of N consecutive frames. Each set is used to learn a new dominant color[1], by taking into account only the pixels that are not similar with the previous learnt dominant colors. More precisely, let D_c denote the list of the first c dominant colors identified on c previous sets of N consecutive frames. The recursive learning process starts with an empty set D_0 of identified dominant colors. A pixel of coordinate $\mathbf{x} \in \mathbb{R}^2$ of a frame I (where $I \in \mathbb{R}^{m\times n\times d}$ is the image, m is the height of the image, n is its width and d is its number of channels) of the $(c+1)^{\text{th}}$ set is said to be active if its distance to all colors in D_c is larger than a threshold T_1. The distance between a color pixel $I(\mathbf{x})$ and a reference color $C \in \mathbb{R}^d$ is computed using a variant of the robust contrast adaptive color dissimilarity proposed in [4]:

$$d(I, \mathbf{x}, C) = 1 - \exp\left(\frac{-\|I(\mathbf{x}) - C\|^2}{2\langle\|I - C\|^2\rangle}\right)$$

where $\|I(\mathbf{x}) - C\|$ is the $\mathcal{L}_2$ norm of the RGB color difference and $\langle . \rangle$ is the expectation operator. To learn the j^{th} dominant color, the color that appears the most frequently among the active pixels of each frame of the j^{th} set is computed. Because this value corresponds to the highest peak in the color d-dimensional histogram of the active pixels, we name it the first histogram mode. By assuming that the color distribution of the recorded scene does not significantly change over the N consecutive frames of the j^{th} set, the vectorial center of these N accumulated first histogram modes gives a reliable representation of the j^{th} dominant color. In Section 3.1, two different definitions of vectorial centers, namely centroid and medoid, and their associated computation methods (k-means clustering and k-medoids clustering), will be compared. As it will be seen, because of its robustness to noisy data, medoid is chosen to define the dominant color from a set of N first histogram modes. Also, because the authors of [23] have shown that k-medoids clustering process runs faster than k-means (complexity of $\mathcal{O}(ikl)$, where i is the total number of iterations, k is the total number of clusters, and l is the total number of data points) under normal distribution of the data points, we first pre-process the N first histogram modes in order to make them more normal distribution-like. Practically, this is done by stabilizing their variance with a box-cox transform [3]. For a vector $[x_1 \dots x_P]$ of strictly positive entries ($x_p > 0\ ,\ \forall\ p \in [1; P]$), the box-cox transform determines a parameter $\lambda \in \mathbb{R}$ such as to maximize the correlation of the transformed data distribution with a normal distribution plot. The following transformation is then applied:

$$x_p^{(\lambda)} = \begin{cases} \frac{x_p^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x_p) & \text{if } \lambda = 0 \end{cases}$$

Although the box-cox transform preserves the similarity of the orderings of the data, the imperfect symmetry of the transformed distribution could produce a small bias in the medoid estimation. However, in practice, as observed in [11],

[1] Instead of learning multiple dominant colors per set, we propose to reduce N.

this bias has only a negligible impact on the final solution, while enabling an important speed-up. Finally, the learning process stops if the percentage of pixels that are correctly approximated by one of the learnt dominant colors reaches a threshold T_2, *i.e.* when the percentage of active pixels drops below T_2.

2.2 Running phase: fast color segmentation

Once the dominant colors have been determined, they are used as color priors in a segmentation process. First, based on the robust color distance presented previously, all the pixels of a new frame are compared with all the learnt dominant colors. For a given pixel, only the smallest distance and the index of the associate dominant color are stored. After, despite the use of a robust metric for labelization, it could happen that some pixels are not labelled consistently, *i.e.* their label does not correspond to the label of the surrounding pixels that belong to the same region. It mainly appears when the sensor used to record the scene is noise sensitive or when the video is highly compressed, leading to wrong local colors or noisy frames. For this reason, we propose to filter the result of the labelization process by a median filter (size 3×3). In this way, the frames are segmented into regions of uniform color and connected pixels.

2.3 Update phase: renewing of the list of dominant colors

In the two last sections, a color prior has been learnt and propagated temporally. However, in both static and dynamic camera setups, the color distribution generally changes over time. For this reason, we propose to extend the list of learnt dominant colors by learning several new dominant colors from batch of N consecutive frames when the percentage of active pixels overshoot T_2. In order to avoid a discontinuous segmentation of the frames, these N frames are also segmented using D_c. Also, to limit the expansion of the list of learnt dominant colors, we delete the least representative dominant color from D_c, *i.e.* the one with the highest mean color dissimilarity $d(I, \mathbf{x}, C)$, before adding a newly learnt dominant color. Finally, the switch from the update phase to the running phase is done when the percentage of active pixels drops below T_2.

3 Experimental validation

In this section, we first show the necessity of determining each dominant color in a robust way. This is done by comparing the results of two different validations. The first validation learns a dominant color by representing the set of accumulated first histogram modes by its centroid (computed via k-means clustering), while the second one represents the set by its medoid[2] (computed via k-medoids clustering). This comparison support the well-known robustness of k-medoids over k-means clustering [23], which makes it more adapted to learn the dominant colors. After, the efficiency and the effectiveness of the proposed segmentation method are evaluated on three different datasets.

[2] A medoid can be seen as a generalization of a median value when the dimension of the data space is higher than 1.

3.1 Comparison between k-means and k-medoids learning

K-means [21] and k-medoids [23] are both partitions-based clustering methods. They aim at dividing a database into groups, such that the samples that belong to the same group are similar and those belonging to different groups are dissimilar. More precisely, a partitioning method generates k clusters from a given set of n data objects. Because image/video segmentation is defined as partitioning n pixels into separated regions, such methods are thus perfectly adapted to the segmentation problem. However, while k-means clustering has been deeply investigated in segmentation [24, 18], k-medoids tends to be rarely used.

K-medoids and k-means differ in the way of representing a cluster. While k-means represents a cluster by the average value (called centroid) of its associated data, k-medoids takes the most centrally located data (called medoid) of the cluster to represent it. The fact that the k-medoids method defines a cluster by its most representative point has two major consequences:

- K-medoids clustering is robust to outliers and noisy data, as opposed to k-means clustering [23]. Indeed, while a mean is highly sensitive to extreme values, a medoid is perfectly suited to derive a representative tendency from its central sample, even in skewed distributions.
- By definition, a medoid belongs to the data space. In contrast, if the space is not convex, a centroid (average) may lie outside the space. This might end up in the definition of a reference “dominant“ color that is close to several colors of the scene, without actually matching any of those colors.

These two fundamental differences between k-means and k-medoids are illustrated in Figure 1.

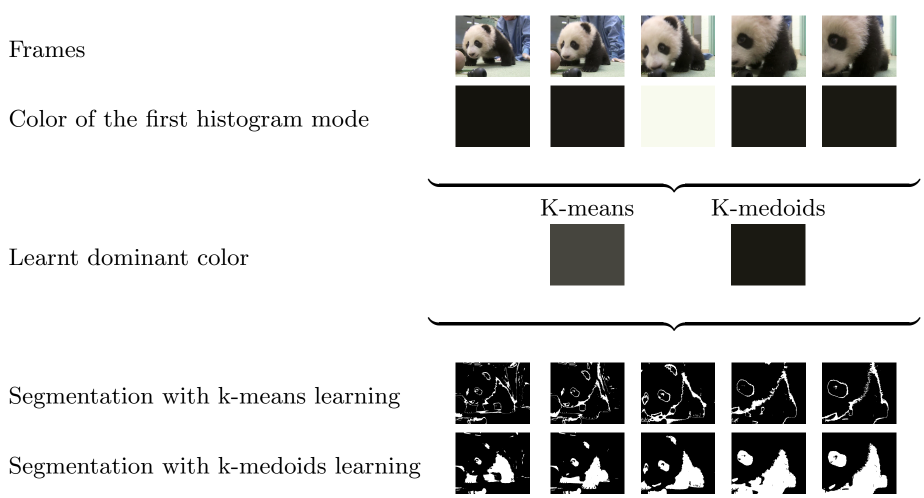

Fig. 1: The dominant color (3^{rd} row) determined on the first histogram modes (2^{nd} row) of a video (1^{st} row) shows that a learning based on k-medoids clustering (5^{th} row) is more robust to noisy data than one based on k-means (4^{th} row).

The first row represents different frames taken from a video. The second row represents the first histogram mode for each frame, as defined in Section 2.1. The third row illustrates the dominant color computed by the 1-means clustering (left) and 1-medoids clustering (right), based on the set of first histogram modes extracted from the frames. Because this video focuses on a panda, we expect the first dominant color to be either black or white.

As a first observation, the first histogram mode detected on the third frame corresponds to the bright color of the wall. It can thus be considered as an outlier in the set of first histogram modes, where the black color of the panda is preponderant. The grey dominant color learnt by k-means clustering (third row of Fig. 1) shows that this outlier highly attracts the centroid in the k-means clustering, while it does not influence the medoid of the k-medoids clustering. The learning of dominant colors based on k-means clustering is thus strongly biased by noisy data, while a learning based on k-medoids clustering is robust. As a second observation, the dominant color learnt by k-means clustering (grey) does not represent a color of the panda. Any segmentation algorithm that tries to segment the panda based on this learnt dominant color will thus fail, as illustrated in the fourth row of Fig. 1. At the opposite, the last row of Fig. 1 shows that k-medoids clustering is well adapted to recursively learn the reference colors used in a color-based segmentation. Those two observations led us to prefer k-medoids over k-means clustering. In the proposed method, k-medoids clustering is thus applied to determinate a dominant color from N first histogram modes. To still decrease the sensitivity to noisy data in the determination of a dominant color, the medoid is defined by minimizing the sum of the L_1 dissimilarities to the data, instead of the common euclidian distance.

3.2 Performances

The efficiency and the effectiveness of the proposed segmentation method are evaluated on three different datasets. On the one hand, as the ground-truths are not available for the first dataset, we visually compare our results with others. On the other hand, the ground-truth of the second dataset is used to objectively measure the efficiency of the proposed algorithm. Finally, the last dataset validates the effectiveness of the proposed update phase.

The first dataset, proposed by [12], is composed of several video clips representing natural scenes and is used to give a qualitative comparison between our video segmentation method and the conventional mean-shift algorithm [8]. Figure 2 (Figure 3) illustrates the results of both algorithms when the color range parameters (h_r in the mean-shift algorithm and T_1 in our method) are high (low), in such a way to segment the video into a very small (high) amount of regions. First of all, as shown on these figures, the proposed method preserves the segmentation effectiveness. After, the computational complexity (in terms of running time of a Matlab implementation on a 3GHz Intel I7 CPU, 8Gb RAM machine) of both methods has been evaluated on all the video sequences of this dataset. This assessment shows that while the mean-shift algorithm segments at approximatively 0.026 fps, our algorithm is more than 250 times faster (7.24 fps).

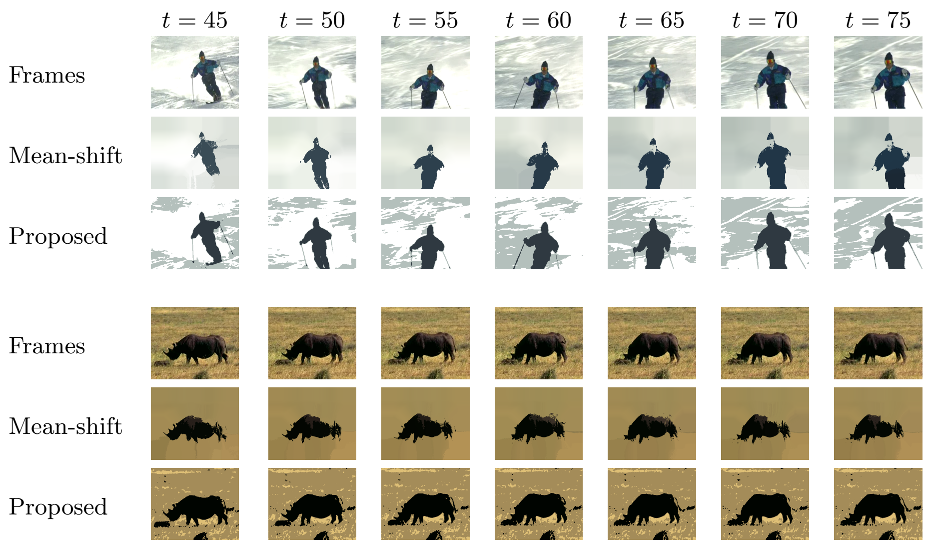

Fig. 2: For a dense segmentation ($T_1 = 0.35$ and $T_2 = 1$ in our method; $h_s = 50$ and $h_r = 30$ for the mean-shift segmentation), a learning of the dominant colors on $N = 15$ frames gives similar segmentation results, while segmenting more than 250 times faster.

Fig. 3: For an over-segmentation ($T_1 = 0.2$ and $T_2 = 1$ in our method; $h_s = 50$ and $h_r = 10$ for the mean-shift segmentation), a learning of the dominant colors on $N = 5$ frames gives similar segmentation results, while segmenting more than 250 times faster.

In Figure 4, we propose a similar qualitative comparison between our method and the one proposed in [12] (from which this first dataset has been provided). By extracting the regions corresponding to the dominant color of the fox, we can see that our method achieves similar effectiveness, while using only a weak prior in the segmentation process.

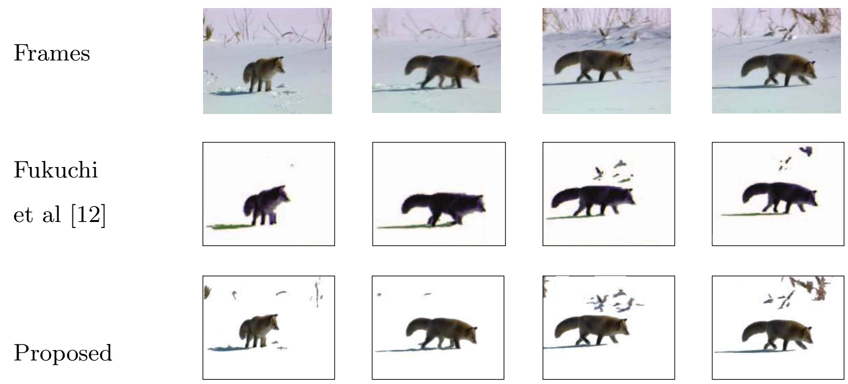

Fig. 4: Our method achieves similar effectiveness than [12], while using only a weak prior in the segmentation process ($N = 3$, $T_1 = 0.65$ and $T_2 = 1$).

The second evaluation is used to go one step beyond the general validations proposed in video segmentation, by giving a complementary quantitative assessment. To the best of the authors' knowledge, no segmentation ground-truth exists for the entire frames of a video. However, elaborated benchmarks and ground-truths are available for individual image segmentation. For this reason, we evaluate the accuracy of the proposed video segmentation method on videos constituted of a repetition of these test images. This validation is thus complementary to the first one: after having shown that the learning and segmentation of our method are effective, we objectively assess their efficiency. We follow the segmentation evaluation methodology proposed by [1], by computing the F-measure (harmonic mean of precision and recall [1]) on segmentation results obtained on various complex images and multiple corresponding ground-truths. In Table 1, the F-measure of our method is compared with several standard methods and shows that our method outperforms them.

Algorithms	F-measure
Proposed method	$\mathbf{0.94 \pm 0.107}$
Alpert et al. (CVPR'07) [1]	0.86 ± 0.012
Galun et al. (ICCV'03) [14]	0.83 ± 0.016
Shi and Malik (PAMI'00) [22]	0.72 ± 0.018
Comaniciu and Meer (PAMI'02) [8]	0.57 ± 0.023

Table 1: Our method outperforms conventional algorithms

As a last validation, we proof the effectiveness of the update of the dominant colors. Figure 5 illustrates a video of another dataset, proposed by [15]. As a first observation, only few frames are dedicated to learn and update the list of dominant colors. More precisely, in this arbitrary video, less than 8% of the frames are used either for the learning or for the update phase, explaining the high computing speed of our algorithm. As a last observation, the effectiveness of the update phase is demonstrated at $t = 14$ and $t = 85$. Indeed, because of the apparition of the whale at $t = 14$, the list of dominant colors is automatically updated by detecting and adding its dark color. Also, at $t = 85$, when the whale falls back in the sea, the list of dominant colors is anew updated, because of the apparition of the light color representing the scum.

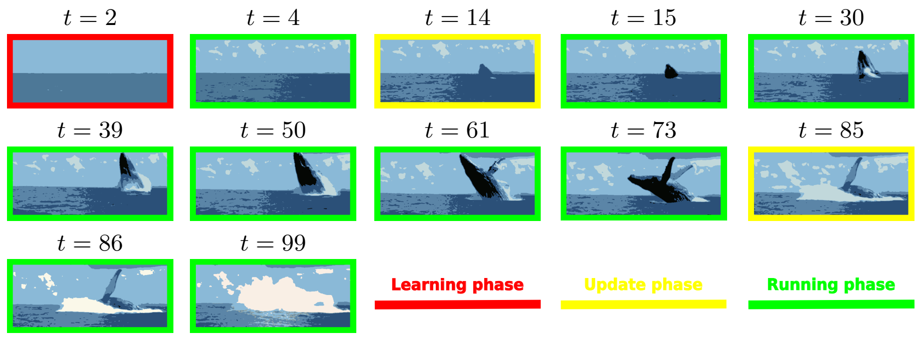

Fig. 5: The list of dominant colors is updated on-the-fly when the colors of the scene change (segmentation obtained with $N = 1$, $T_1 = 0.2$ and $T_2 = 1$).

4 Conclusion

In this article, a computationally efficient video segmentation method has been presented. The proposed approach learns, on few frames at the beginning of the video sequences, the dominant colors representing the recorded scene. We have shown that k-medoids clustering is more adapted than k-means clustering to recursively and robustly determine these dominant colors. This color prior is then used to speed-up a fast on-the-fly color-based segmentation of the next frames of the video sequence. To deal with videos in which the color distribution of the scene changes over time, we have proposed and validated an approach to update the list of learnt dominant colors. Finally, the efficiency and the effectiveness of the proposed method have been demonstrated on three standard segmentation benchmarks. This assessment shows that our approach is more than 250 times faster than the conventional mean-shift segmentation, while overcoming the segmentation performances of some state-of-the-art methods.

References

1. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: Proc. of IEEE CVPR'07. pp. 1–8 (June 2007)
2. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. ACM SIGGRAPH'09 28(3), 70 (2009)
3. Box, G., Cox, D.: An analysis of transformations. Journal of the Royal Statistical Society. Series B (Methodological) pp. 211–252 (1964)

4. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. IJCV'06 70(2), 109–131 (2006)
5. Brendel, W., Todorovic, S.: Video object segmentation by tracking regions. In: Proc. of IEEE ICCV'09. pp. 833–840 (2009)
6. Carreira, M.: Gaussian mean-shift is an EM algorithm. IEEE PAMI'07 29(5), 767–776 (2007)
7. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE PAMI'95 17(8), 790–799 (1995)
8. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE PAMI'02 24(5), 603–619 (2002)
9. Dorea, C., de Queiroz, R.: Depth map reconstruction using color-based region merging. In: Proc. of IEEE ICIP'11. pp. 1977–1980 (2011)
10. Ellis, L., Zografos, V.: Online learning for fast segmentation of moving objects. In: Proc. of ACCV'12 (2012)
11. Fitzmaurice, G.M., Lipsitz, S.R., Parzen, M.: Approximate median regression via the box-cox transformation. The American Statistician 61(3), 233–238 (2007)
12. Fukuchi, K., Miyazato, K., Kimura, A., Takagi, S., Yamato, J.: Saliency-based video segmentation with graph cuts and sequentially updated priors. In: Proc. of IEEE ICME'09. pp. 638–641 (2009)
13. Fulkerson, B., Soatto, S.: Really quick shift: image segmentation on a GPU. In: Proc. of the ECCV'10 Workshop CVGPU (2010)
14. Galun, M., Sharon, E., Basri, R., Brandt, A.: Texture segmentation by multiscale aggregation of filter responses and shape elements. In: Proc. of IEEE CVPR'03. pp. 716–723 (2003)
15. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph based video segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010)
16. Huang, Y., Liu, Q., Metaxas, D.: Video object segmentation by hypergraph cut. In: Proc. of IEEE CVPR'09. pp. 1738–1745 (2009)
17. Li, Y., Sun, J., Shum, H.: Video object cut and paste. ACM SIGGRAPH'05 24(3), 595–600 (2005)
18. Mignotte, M.: A de-texturing and spatially constrained k-means approach for image segmentation. Pattern Recognition Letters 32(2), 359–367 (2011)
19. Moscheni, F., Bhattacharjee, S., Kunt, M.: Spatio-temporal segmentation based on region merging. IEEE PAMI'98 20(9), 897–915 (1998)
20. Price, B., Morse, B., Cohen, S.: Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In: Proc. of IEEE ICCV'09. pp. 779–786 (2009)
21. Seber, G.: Multivariate observations. Wiley (1984)
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE PAMI'00 22(8), 888–905 (2000)
23. Velmurugan, T., Santhanam, T.: Computational complexity between k-means and k-medoids clustering algorithms for normal and uniform distributions of data points. Journal of Computer Science 6(3), 363–368 (2010)
24. Verleysen, C., De Vleeschouwer, C.: Recognition of sport players' numbers using fast color segmentation. Proc. of the SPIE-IS&T Electronic Imaging (SPIE'12) 8305 (2012)
25. Wang, T., Guillemaut, J., Collomosse, J.: Multi-label propagation for coherent video segmentation and artistic stylization. In: Proc. of IEEE ICIP'10. pp. 3005–08 (2010)