Dynamic Color Flow: A Motion-Adaptive Color Model for Object Segmentation in Video

Xue Bai[1], Jue Wang[2], and Guillermo Sapiro[1]

[1] University of Minnesota, Minneapolis, MN 55455, USA
[2] Adobe Systems, Seattle, WA 98103, USA

Abstract. Accurately modeling object colors, and features in general, plays a critical role in video segmentation and analysis. Commonly used color models, such as global Gaussian mixtures, localized Gaussian mixtures, and pixel-wise adaptive ones, often fail to accurately represent the object appearance in complicated scenes, thereby leading to segmentation errors. We introduce a new color model, *Dynamic Color Flow*, which unlike previous approaches, incorporates motion estimation into color modeling in a probabilistic framework, and adaptively changes model parameters to match the local properties of the motion. The proposed model accurately and reliably describes changes in the scene's appearance caused by motion across frames. We show how to apply this color model to both foreground and background layers in a balanced way for efficient object segmentation in video. Experimental results show that when compared with previous approaches, our model provides more accurate foreground and background estimations, leading to more efficient video object cutout systems.[1]

1 Introduction

Creating accurate masks for video objects is a fundamental component in the professional video post-processing pipeline. Once being accurately segmented from the video, the target objects can be used to create seamless composites, or be manipulated to create special visual effects. Recently, interactive, or user-guided video segmentation systems have gained considerable attention, given the fact that interactive systems can generate more accurate segmentation results than fully automatic ones, on a wide range of videos.

Although significant breakthroughs have been achieved in recent years on interactive video segmentation and matting [1], this problem remains difficult for complex real world video sequences. The difficulty comes from two main aspects, namely *appearance complexity* and *motion complexity*. Appearance complexity refers to the fact that the targeted object could contain very similar, or even the same colors and features as the background, thus distinguishing the object from its background using color information becomes a hard problem. [2] In addition,

[1] Work partially supported by NSF, NGA, ONR, ARO, and NSSEFF.

[2] Not limited to colors, the object appearance can also incorporate other types of features depending on the applications.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part V, LNCS 6315, pp. 617–630, 2010.
© Springer-Verlag Berlin Heidelberg 2010

video objects or backgrounds often exhibit nonuniform motions. Thus, applying to the next frame an appearance model constructed from the current one, will be problematic without correctly adapting it to the new position of the possibly deforming object/background caused by the motion.

Although various approaches have been proposed in recent years to tackle these problems, they either do not employ color models that are powerful enough to handle the appearance complexity, or do not adequately consider the motion complexity when updating the models across frames. We will analyze these limitations in more detail in the next section. As a result, the color models used in previous systems are often too rigid to handle video sequences with complex appearance and motion. Even with the help of other priors such as shape, pose, and structure, color is still an important feature in most natural videos, thus inaccurate color modeling often directly leads to segmentation errors. While these errors are correctable in an interactive setting, the user has to provide more manual input, which could be time consuming in many cases.

We introduce a new motion-adaptive color model called *Dynamic Color Flow*, or *DCF*. In this model, we combine motion estimation and color modeling into a single probabilistic framework that simultaneously addresses the appearance and motion complexities. The basic idea is to automatically and adaptively select the suitable color model, continuously ranging from a global model to a localized one, for different parts of the object, so that it can be reliably applied to segmenting future frames. The proposed framework does not assume accurate motion estimation. In fact, it takes into account the estimation errors and only assumes the motion estimation to be probabilistic, thus any motion algorithm with reasonable performance can be embedded into our system. Furthermore, we show how to apply the proposed DCF model to both foreground and background layers, leading to an efficient video cutout system as demonstrated by numerous examples.

1.1 Related Work

Video object segmentation is a classic problem that has been extensively studied for decades. Instead of surveying the large volume of literature, which is impractical here, we focus on classes of recent works that are most related to our system, and analyze their limitations, especially on color modeling.

Global color models. Some modern interactive video cutout systems use global color models, such as the popular choice of global Gaussian mixtures (GMM), to represent the appearance of the dynamic objects, e.g., [2–4]. Global color models do not consider the spatial arrangements of color components, thus are robust to object motion. However, the discrimination power of global models is too limited to deal with objects with complicated appearance.

Pixel-wise color models. The other extreme of color modeling is to consider every pixel on the image plane independently. Such method is often used in background subtraction systems. Assuming the camera is fixed and the background is static, these systems will form statistical models at every pixel location to describe the observed background colors, e.g., [5, 6]. However, using these models

require accurate frame-to-frame alignment, which may not be possible with dynamic background scenes.

Localized color models. The recently proposed SnapCut system [7] employs localized color models (see also [8]). It consists of a group of spatially constrained color components that are distributed along the object's boundary in an overlapping fashion. Each color component includes a GMM with a fixed spatial domain. When propagated across frames, these local models are first pushed by optical flow vectors to arrive at new destinations, before being applied for local segmentation. It has been shown that by localizing the color models, the foreground object can be modeled more accurately, leading to efficient segmentations. Although in this approach motion estimation is used to move local color models across frames, it is treated independently from color modeling and classification. The scale (spatial domain) of all local color models are fixed without considering the underlying motion. This can cause two problems: when the local motion is strong (like a waving hand), optical flow may lose track, and the fixed window size will be too small to allow the localized color models to capture the object. On the other hand, for parts of the object where local motion is small, the window size may become too large to accurately model the foreground to background transition. We will demonstrate these problems with real examples in later sections.

Bilayer segmentation. Recently, significant success has been achieved for live speaker-background segmentation for video conferencing. Assuming a stationary background, the background cut system [9] uses a background contrast attenuation method to adaptively suppress the contrasts that belong to the background, making extracting the foreground easier. The *i2i* system, [10], avoids explicit motion estimation using a second order HMM model as a temporal (learned) prior on segmentation. These systems can efficiently segment a video in a constrained environment, but are hard to generalize for other types of videos, such as the examples shown in this paper.

2 Dynamic Color Flow

To explain the proposed *Dynamic Color Flow model* (*DCF*), we first put aside the whole interactive video object segmentation workflow, and focus on the fundamental problem of *segmentation propagation*, that is, given a known correct foreground/background segmentation on frame t, how to use it to build accurate color models for segmenting the foreground/background on frame $t + 1$. Note that for now we do not distinguish between foreground and background, we will show in Section 3 how to apply the model to both regions.

Segmentation is trivial if an accurate motion vector field between frames is available: for every pixel on frame $t + 1$, we just trace it back to the previous frame and see whether it comes from the target region or not. However, a perfect motion vector field is almost impossible to compute in real world, and directly using it for segmentation will be erroneous. The DCF model proposed in our

system explicitly models the motion inaccuracy, and provides a probabilistic framework unifying the local colors of the object and their dynamic motion.

Let Ω be the region of interest on frame t (Ω can be foreground F, background B, or other object in case of multiple objects). Ω contains $|\Omega|$ pixels X_i ($i = 1, 2, 3, \ldots, |\Omega|$). Denote the position of pixel X_i as x_i. For each pixel X_i inside Ω, we use the locally-averaged optical flow $\boldsymbol{v}$ as the motion vector to predict its position in frame $t+1$, $x_i' = x_i + \boldsymbol{v}$.[3] Assuming the motion vector is not accurate enough, instead of using x_i' deterministically, we treat it as the center of a Gaussian distribution,

$$f_i(y) = \frac{1}{\sqrt{2\pi}\sigma_i} exp(-\frac{\|y - x_i'\|^2}{2\sigma_i^2}), \tag{1}$$

where y is a location in frame $t+1$. The variance σ_i measures the fuzziness of the prediction. Its value is dynamically set for each pixel, as we will explain in the next section.

Let c_{X_i} be the color vector of pixel X_i. The probabilistic prediction propagates the colors in Ω to the next frame and generates a distribution $p(c, y|\Omega)$, the probability of observing the color c at location y on frame $t+1$ given that all colors come from Ω on frame t. The conditional color distribution at y is

$$p(c|y, \Omega) = \frac{p(c, y|\Omega)}{p(y|\Omega)}, \tag{2}$$

where $p(y|\Omega) = \sum_{i=1}^{|\Omega|} p(X_i)p(y|x_i)$ is a spatial term independent of color, so we treat it as a normalization constant. Since $p(c, y|\Omega)$ is contributed by all pixels in Ω, it can be written as

$$p(c, y|\Omega) = \sum_{i=1}^{|\Omega|} p(X_i)p(c, y|X_i). \tag{3}$$

Since the predicted position of X_i is independent of its color,

$$p(c, y|X_i) = p(c|c_{X_i})p(y|x_i). \tag{4}$$

Then we have

$$p(c|y, \Omega) = \frac{\sum_{i=1}^{|\Omega|} p(X_i)p(c|c_{X_i})p(y|x_i)}{p(y|\Omega)}, \tag{5}$$

where $p(c|c_{X_i})$ is the probability of observing color c on frame $t+1$ given the existence of c_{X_i} on frame t. Given the fact that colors of the same object may vary across frames due to illumination changes, compression, and noise, we model this as a 3-D Gaussian distribution with mean vector c_{X_i} and covariance matrix Σ, i.e., $p(c|c_{X_i}) = \mathcal{N}(c|c_{X_i}, \Sigma)$. We will describe the explicit computation later.

[3] Similarly to [7], we average optical flow vectors locally to remove noise.

As previously defined, $p(y|x_i) = f_i(y)$. Assuming equal priors for every pixel, $p(x_i) = 1/|\Omega|$, then

$$p(c|y, \Omega) \propto \sum_{i=1}^{|\Omega|} f_i(y)\mathcal{N}(c|c_{X_i}, \Sigma). \tag{6}$$

From Eqn. (6) it is clear that $p(c|y, \Omega)$ can be interpreted as a non-parametric density estimation of the color sample set $\{c_{X_i}|i = 1, 2, \ldots, |\Omega|\}$. Each sample c_{X_i} is weighted by $f_i(y)$, which is the probability of c_{X_i} arriving at y. We observe that the color sample set encodes the motion estimation of the color samples across video frames, thus the model inherently fuses motion and appearance into a unified framework.

It is worth mentioning that there has been some formal studies on modeling the statistics of optical flow ([11],[12],[13],[14],[15]). Particularly, [12] studied its spatial properties and the brightness constancy error, resulting in a probabilistic model of optical flow. Our model is quite different in the following aspects. First, those works aim at improving optical flow estimation of natural images by considering learned prior distributions from ground truth training data, while our framework employs probabilistic methods on existing optical flow results for the purpose of generating more accurate color models for segmentation. Second, the learned statistics in [12] are global priors, while ours allows defining the distribution for individual pixels depending on the local motion. In fact, our model works with any optical flow algorithm that has reasonable performance and can certainly benefit from replacing the simple Gaussians by more accurate distributions as in [12].

Directly estimating $p(c|y, \Omega)$ for each pixel location on frame $t + 1$ is computationally expensive, therefore we employ the following approximations to efficiently speed up the computation. First, we use the *Luv* color space and assume class-conditional independence of the three channels, [4] thus $p(c|y, \Omega)$ can be estimated as the product of three 1-D *PDFs* rather than a 3-D *PDF*, and the covariance matrix Σ in Eqn. (6) can be computed in each channel. Second, the 1-D *PDFs* at every y location are incrementally built using a quantized histogram containing 32 bins. Denoting the L-channel histogram at y as H_y^L, when propagating X_i, the L component of c_{X_i} with weight $f_i(y)$, is added to H_y^L for every y that is within a neighborhood centered at x_i' with a radius of $R = 4\sigma_i$ (we then use a truncated Gaussian to replace the Gaussian function in Eqn. (1)).

After propagating all pixels within Ω, we apply 1-D kernel density estimation, [16], on every histogram. Let now $\bar{H}_y^L$ be the estimated density for the L channel at y, u and v channels are similarly computed. Also, let us denote the color at y in frame $t + 1$ as $c_y = \{l, u, v\}$. Finally, the probability of c_y coming from Ω is

$$p(c_y|y, \Omega) = \bar{H}_y^L(l) \cdot \bar{H}_y^u(u) \cdot \bar{H}_y^v(v). \tag{7}$$

[4] Note that class-conditional independence is a weaker assumption than feature independence.

This procedure computes the probability for every pixel in the next frame $t+1$ once the parameters σ_i are given. Next we show that the model adaptively changes scales by using different σ_i-s.

Global Color Model. When all the $\sigma_i \to \infty$, all color samples are equally weighted, generating identical color distribution at each location, which is equivalent to a global color model. As mentioned earlier, this model only works well when the object has distinct colors from the rest of the scene, and is not affected by large motions which are hard to track.

Localized Classifier. Setting all σ_i-s to the same value r, we get a set of moving localized classifiers similar to the recently proposed SnapCut system [7]. This model assumes the object can be tracked reasonably well, i.e., the tracking error is less than $4r$.

Stationary Pixel-wise Model. When $\sigma_i \approx 0$, we have the pixel-wise color models commonly used in previous background subtraction systems ([5],[6],[9]). This model can be used if the video background is still or an accurate alignment can be achieved.

Dynamic Model. In all the above cases the motion scales of different parts of the object are assumed to be the same. However, we argue that most real world examples are likely to contain of motions of mixed scales. For instance, for a walking person, his/her hand or foot motion obviously has a larger scale than his/her body. Thus, by dynamically determining σ_i for every pixel, our model offers the flexibility to adapt to different motion scales, even on the same object. This is a key advantage of the proposed DCF model. We will describe how to compute σ_i in the next section when we demonstrate how to apply this model to video segmentation.

3 DCF for Video Object Segmentation

In this section we apply the proposed DCF model to user-guided video object segmentation. We assume the video contains two independent foreground (F) and background (B) layers, although there is no fundamental limit on extending our model to multiple layers. The DCF model is applied to both F and B layers for a balanced modeling. The segmentation is then solved within a MRF framework.

3.1 The Foreground Layer

The foreground object usually presents various local motion scales. σ_i, by its definition (see Eqn. (1)), is related to the prediction error of the foreground optical flow. For erratic movement where the optical flow is likely to contain large errors, we set σ_i to large values. For slow or stable motion, the optical flow is generally more reliable, thus the value of σ_i is reduced, yielding more localized color models which have greater classification power. In this way σ_i changes adaptively with the prediction error for the different parts of the object.

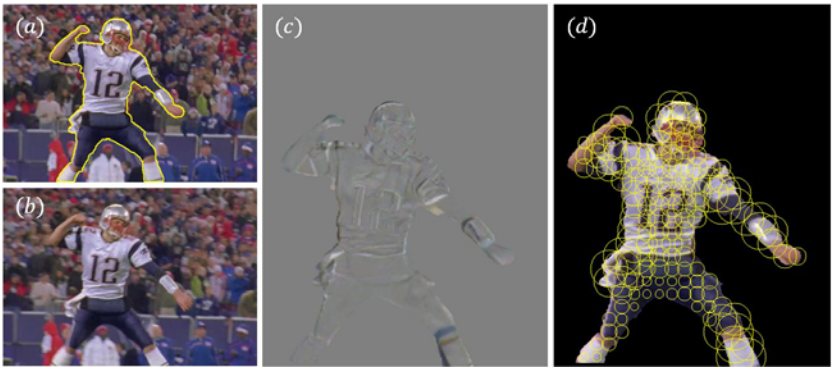

Fig. 1. (a) Frame t with known segmentation (*the yellow contour*). (b) Frame $t+1$. (c) The difference image between the warped frame t and frame $t+1$. (d) Values of σ_i (*yellow circles*), adapting to the local average intensity of (c) across the object.

To compute the prediction error, the key frame is warped by the (locally averaged) optical flow to align with the next frame. In our system we define the alignment error $e(x)$ as the local average of frames difference, $e(x) = \sqrt{\frac{1}{m}\sum_{x\in N_x\cap\Omega'_F}\|I'_t(x) - I_{t+1}(x)\|^2}$, where N_x is a square neighborhood centered at x, I'_t and Ω'_F are the warped color image and the binary foreground map from frame t to $t+1$ respectively, and m is the number of foreground pixels in N_x. Accurate alignment generally indicates reliable optical flow in the local regions, thus σ_i can be defined linearly proportional to $e(x)$. For flat, textureless regions where the local alignment error is always small, a lower bound term σ_{min} is added to increase robustness. Defining the local smoothness as $s(x) = \frac{1}{1+\beta\cdot\bar{g}(x)}$, where $\bar{g}(x) = \sqrt{\frac{1}{m}\sum_{x\in N_x\cap\Omega'_F}|\nabla I_\sigma(x)|^2}$ is the local average of image gradient, and $I_\sigma = I'_t * G_\sigma$, we compute

$$\sigma_i = \begin{cases} \alpha\cdot e(x'_i) + s(x'_i)\cdot\sigma_{min}, & e(x_i) \le e_{max}, \\ \alpha\cdot e_{max}, & e(x'_i) > e_{max}, \end{cases} \tag{8}$$

where $\alpha\cdot e_{max}$ is the upper bound of σ_i. Typically $\alpha = 0.2$, $\beta = 10$, $e_{max} = 50$, and $\sigma_{min} = 4$. We will later show that this definition leads to improved results over traditional fixed σ_i color models (see Fig. 6), while our system is general to adopt more sophisticated estimation of σ_i. Compared to [7], where the colors are sampled within windows of a constant size, our algorithm uses a flexible sampling range that generates more accurate local color distributions. An example is shown in Fig. 1, where we can clearly see how σ_i changes based on local motion estimation errors.

3.2 The Background Layer

The background layer can be essentially treated in the same fashion as the foreground one. However, the occluded background behind the object is missing

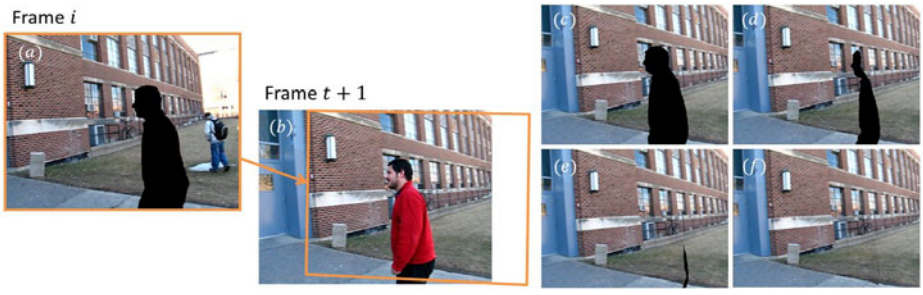

Fig. 2. Consider that all the frames i prior to frame $t+1$ have been segmented. (a),(b) A frame with known background is warped to frame $t+1$ (working frame) using homography. (c)-(f) As additional prior frames are projected, the background is gradually completed, and (f) is used as the background layer for frame $t+1$.

in frame t. In this section we explain two simple scenarios and methods to reconstruct the missing background. Note that our system is not limited to these two methods, and more complicated video mosaicking such as [17],[18], or hole filling algorithms such as [19], can be employed for more accurate background reconstruction.

A Clean Plate. For videos which present a shot of the scene without the objects present, we can directly use the clean plate to build the background model. To deal with moving cameras, we estimate a homography by SIFT matching and RANSAC filtering, and then project the clean plate onto the current frame to be segmented. Similar to the foreground modeling, the DCF model is applied to the reconstructed clean plate, except that σ_i is fixed for every background pixel. Typically for static background σ_i is set to $[2, 4]$ to compensate for small alignment errors.

Progressive Background Completion. In case a clean plate is not available, we use a progressive background completion method similar to the one proposed in [20]. Suppose the first t frames have been segmented, the segmented backgrounds are projected onto frame $t+1$ in a reverse order, from frame t to frame 1, recovering as much occluded background as possible. In general if the foreground object has a large relative motion against the background, a dynamic background plate can be quickly recovered as the segmentation process evolves. Such an example is shown in Fig. 2.

Once the DCF model is constructed for both the foreground and background layers, the foreground probability of a pixel y is computed as

$$p^C(y) = \frac{p(c_y|y, F)}{p(c_y|y, F) + p(c_y|y, B)}, \quad y \in I_{t+1}. \tag{9}$$

As demonstrated in [7], constructing an accurate probability map is the key to achieve accurate object segmentation. Compared with color models used in previous video segmentation systems, the DCF model produces more accurate results,

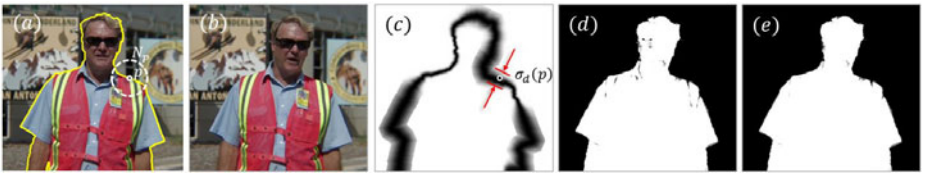

Fig. 3. The shape prior is a variable bandwidth border around the warped object contour (*yellow curve*). (a) For every point p on the contour, compute the average of histogram distance $D(p)$ in the neighborhood N_p. (b) The next frame. (c) Shape prior function, p^S shown in gray scale from darkest (0) to brightest (1). Similar F/B color distributions result in narrow local bandwidth and tight shape constraint, and vice versa. (d) Foreground color probability $p^C(y)$. (e) Integrated shape and color foreground probability $p(y)$.

thanks to the motion adaptive local scale and improved background modeling. In Fig. 6 we will compare the color probability maps generated by DCF and those generated by simple background subtraction, global GMM color models, and the SnapCut system. We tested on difficult examples where foreground and background color distributions are highly overlapping, and the backgrounds are highly cluttered.

3.3 Segmentation with Shape Priors

Directly feeding the color probability map generated by Eqn. (9) (see also Fig. 6) to a graph cut optimization may still result in some small segmentation errors, since the color probability map tends to be noisy. To further improve the segmentation, we borrow the general idea from [7] of incorporating dynamic and local shape priors. The basic idea is to create a variable bandwidth contour adaptive to the local statistics of the DCF model. This is in spirit similar to the variable bandwidth trimap proposed in [4] for the purpose of image matting.

Let p be a point on the object contour (warped from the previous frame), and N_p a neighborhood centered at p. Define the distance between two histograms as $d_H(\bar{H}_1, \bar{H}_2) \triangleq 1 - \sum_i min\left\{\bar{H}_1(i), \bar{H}_2(i)\right\}$. Let $\bar{H}^L_{F,y}$, $\bar{H}^u_{F,y}$, $\bar{H}^v_{F,y}$ be the three foreground color histograms at a pixel y, and $\bar{H}^L_{B,y}$, $\bar{H}^u_{B,y}$, $\bar{H}^v_{B,y}$ the corresponding background color histograms at y. Then, define

$$D(y) \triangleq min\left\{d_H(\bar{H}^L_{F,y}, \bar{H}^L_{B,y}), d_H(\bar{H}^u_{F,y}, \bar{H}^u_{B,y}), d_H(\bar{H}^v_{F,y}, \bar{H}^v_{B,y})\right\}, \tag{10}$$

and for added robustness, consider $\bar{D}(p) \triangleq \frac{1}{K}\sum_{y\in N_p} D(y)$, where K is the number of pixels in N_p. The local shape profile $p^S(y) = 1 - \mathcal{N}(d_y|\sigma_d)$. $\mathcal{N}(d_y|\sigma_d)$ is then a Gaussian distribution with variance σ_d, which is linearly proportional to $\bar{D}(p)$, and with d_y the Euclidean distance from y to the contour point p. Larger $\bar{D}(p)$ indicates that the local foreground and background colors are more separable, thus a wider shape profile is used to give less spatial constraint, and vice versa. Finally, the integrated probability at y, combining both local shape and color models, is defined as

$$p(y) = p^C(y)(1 - p^S(y)) + M'_{t+1}(y)p^S(y), \tag{11}$$

where M'_{t+1} is the warped object mask with 1 inside and 0 outside. Essentially $p^S(y)$ is used as a weight to linearly combine the color probability $p^C(y)$ with the warped object mask M'_{t+1}. Please refer to [7] for more details of this equation. An example is shown in Fig. 3.

Using $p(y)$ as the data term, and the image gradient statistics for the neighborhood term as proposed in [21], the current video frame $t + 1$ is then segmented with a standard graph cuts image segmentation algorithm [22]. Examples are shown in figures 4 and 5. The user can optionally add scribbles to correct segmentation errors towards a more accurate segmentation, which then becomes the key frame for the next frame. This process is repeated until the whole sequence is segmented. Additionally, if necessary, the binary segmentation can be processed with a temporally-coherent matting algorithm, [7], producing soft alpha mattes for the foreground object for high-quality compositing tasks.

4 Experiments and Comparisons

We have tested our system on a variety of challenging video examples containing complex color distributions (figures 4 and 5(b)), highly cluttered background (figures 4 and 7), rapid topology changes (Fig. 5(c)), motion blur (Fig. 4), and camera motion (Fig. 2).

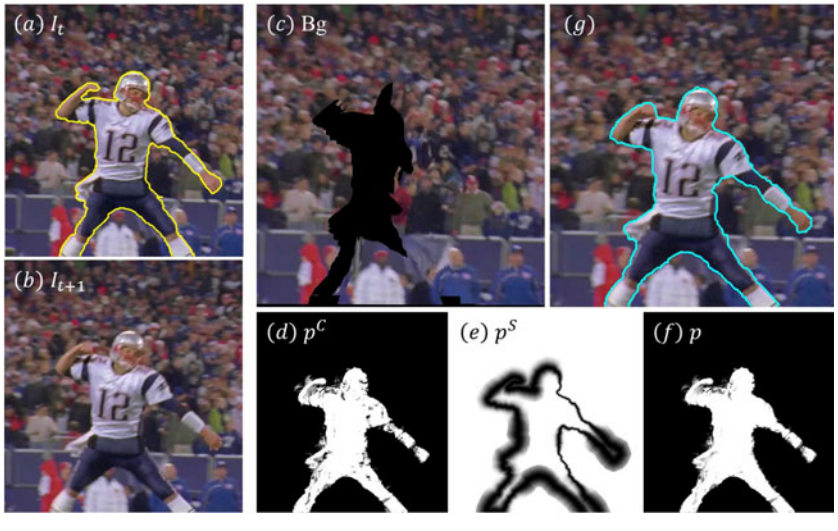

Fig. 4. Propagating the segmentation from frame t to $t + 1$. (a) Frame t with segmentation (*yellow curve*). (b) Frame $t + 1$. (c) The partially recovered background of frame $t + 1$. (d) Color probability $p^C(y)$ shown in gray scale. (e) Shape prior derived from frame t. (f) Incorporating the shape prior further improves the quality of the probability map $p(y)$. (g) Final segmentation (*cyan curve*) of frame $t + 1$ without any user interactions.

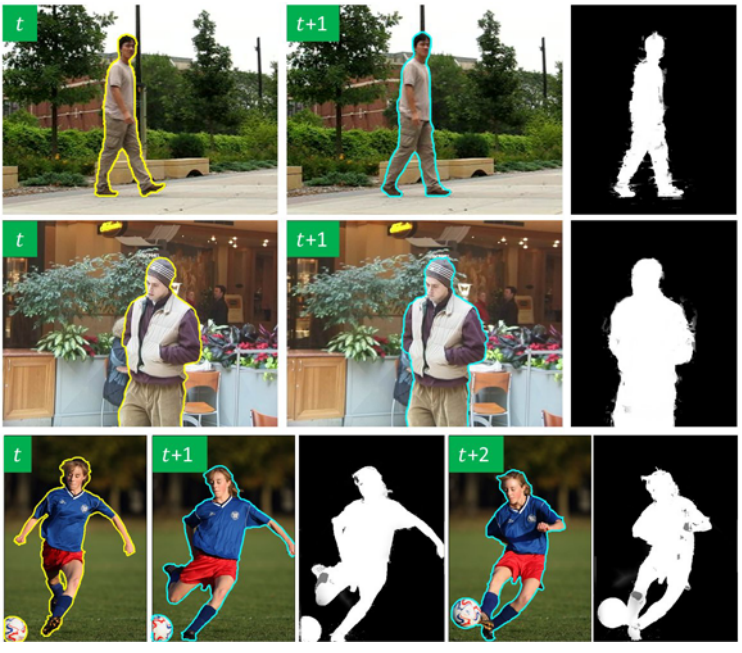

Fig. 5. Additional examples of segmentation propagation on objects with diverse motion. In each example we show the key frame t with segmentation (*yellow curves*), the computed segmentation on frame $t+1$ (*cyan curves*), and the probability maps in gray scale. In the last example, the segmentation propagates two consecutive frames.

First, Fig. 4 shows the intermediate results of segmenting one frame. Note the background reconstruction in (c) is only partially complete. For those pixels without background reconstruction colors, we simply sample nearby background colors for them in our current implementation, which already leads to satisfactory foreground estimation and segmentation, as shown in (d) and (g).

Then, Fig. 5 contains three additional examples that demonstrate different motion scales. In the first example, the walking person moves with dynamic (nonuniform) motion. The foreground in the second example is more stable but contains very complex colors (see supplementary material for the full video). The third example exhibits erratic motion and rapid topology changes that are very hard to track. Our system automatically adapts to these very different examples and produced accurate foreground probabilities that lead to high quality segmentation results.

We compared our proposed color model with background subtraction, global GMM, and the SnapCut system on two examples, as shown in Fig. 6. We used a basic background subtraction algorithm and manually selected the optimal threshold for each example. Due to the rigidity assumption for the static background and the lack of accurate foreground model, the algorithm is generally incapable of high quality segmentation tasks. The global GMM is without any

Fig. 6. Comparing color probability maps and segmentation results generated by simple background subtraction, the global GMM color model, the local color model from [7], and the proposed DCF on two examples. The gray scale images are the color probabilities generated by each method followed by their corresponding segmentation results. (For better visualization, images are cropped from original videos. See figures 3 and 7 for the full frames.)

doubt the least preferred in these examples, as both the foreground and background contain very similar colors. The SnapCut system improves the color probability results by localizing the color sampling. However, errors can occur if colors are confusing even in local regions, e.g., the black color in the glasses and in the background, first example. The DCF model generated more accurate color probabilities and segmentations for these examples.

To evaluate the complete interactive system, we compared our system with SnapCut on a video sequence, Fig. 7 (see supplementary material for additional sequences with comparisons in terms of segmentation accuracy and the amount of user interaction). Our system requires less user input to achieve comparable results. As the propagation progresses, the amount of interactions is further reduced thanks to the improved foreground and background models.

Of course our system cannot deal with all possible situations one may face in video segmentation. The DCF model assumes that all foreground colors on frame $t+1$ have been seen on frame t, thus cannot model newly appeared foreground colors due to occlusion and disocclusion, such as a self-rotating colored

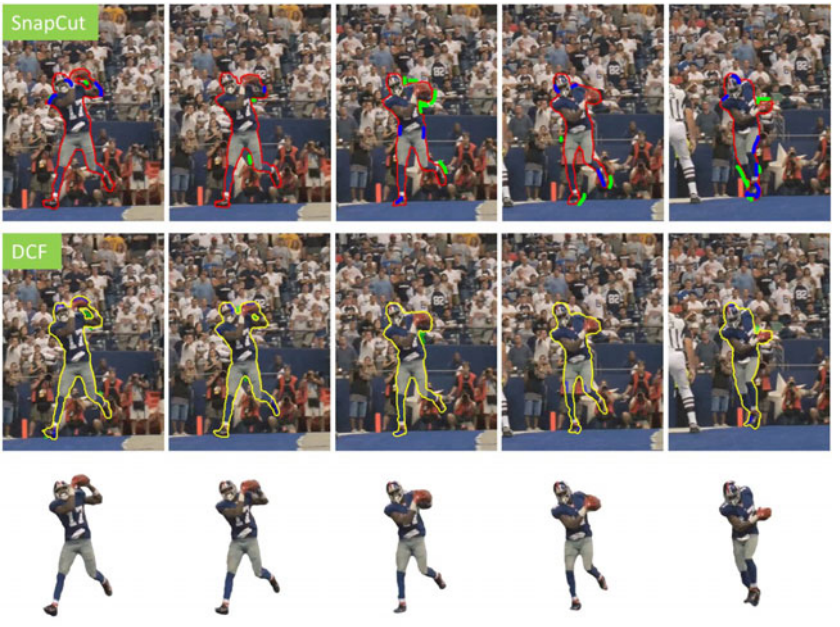

Fig. 7. The football sequence, from left to right: frames 2, 5, 10, 13, 20. Frame 1 is pre-segmented. First row: segmentation (*red curves*) and user scribbles (*blue as foreground and green as background*) by the SnapCut system. Second row: segmentation (*yellow curves*) and user scribbles by our system. Third row: new composites on white.

ball where new colors constantly appear from one side of the object. The shape prior can only be used when the foreground shape is consistent and cannot be applied for things like fire and water. Also, if the background is highly dynamic, like a foreground person passing by a group of walking people, then the simple background construction methods described in Section 3.2 will fail. In these cases, more user input, or more advanced motion estimation and background reconstruction methods, will be needed to improve the performance of the system.

5 Concluding Remarks

A new color model that, unlike previous methods, incorporates motion estimation in a probabilistic fashion, was introduced in this paper. By automatically and adaptively changing model parameters based on the inferred local motion uncertainty, the proposed method accurately and reliably models the object appearance, and significantly improves the foreground color probability estimation. We applied the new model to both foreground and background layers for video object segmentation, obtaining significantly improved results when compared to previous state-of-the-art systems.

References

1. Wang, J., Cohen, M.: Image and video matting: A survey. Foundations and Trends in Computer Graphics and Vision 3, 97–175 (2007)
2. Wang, J., Bhat, P., Colburn, A., Agrawala, M., Cohen, M.: Interactive video cutout. In: Proc. of ACM SIGGRAPH (2005)
3. Li, Y., Sun, J., Shum, H.: Video object cut and paste. In: Proc. ACM SIGGRAPH, pp. 595–600 (2005)
4. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: Proc. of IEEE ICCV (2007)
5. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In: Proc. of ICPR (2004)
6. Elgammal, A., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 751–767. Springer, Heidelberg (2000)
7. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. ACM Trans. Graph. 28, 1–11 (2009)
8. Price, B., Morse, B., Cohen, S.: Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In: Proc. of ICCV (2009)
9. Sun, J., Zhang, W., Tang, X., yeung Shum, H.: Background cut. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 628–641. Springer, Heidelberg (2006)
10. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: Proc. of CVPR (2006)
11. Roth, S., Black, M.J.: On the spatial statistics of optical flow. IJCV 74, 33–50 (2007)
12. Sun, D., Roth, S., Lewis, J.P., Black, M.J.: Learning optical flow. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 83–97. Springer, Heidelberg (2008)
13. Black, M.J., Yacoob, Y., Jepson, A.D., Fleet, D.J.: Learning parameterized models of image motion. In: Proc. of CVPR, pp. 561–567 (1997)
14. Simoncelli, E., Adelson, E.H., Heeger, D.J.: Probability distributions of optical flow. In: Proc of CVPR, pp. 310–315 (1991)
15. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: combining local and global optic flow methods. IJCV 61, 211–231 (2005)
16. Silverman, B.: Density estimation for statistic and data analysis. Monographs on Statistics and Applied Probability (1986)
17. Irani, M., Anandan, P., Bergen, J.: Efficient representations of video sequences and their applications. Signal Processing: Image Communication 8, 327–351 (1996)
18. Rav-Acha, A., Pritch, Y., Lischinski, D., Peleg, S.: Dynamosaicing: Mosaicing of dynamic scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence. 29, 1789–1801 (2007)
19. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proc. of ACM SIGGRAPH, pp. 417–424 (2000)
20. Chuang, Y.Y., Agarwala, A., Curless, B., Salesin, D., Szeliski, R.: Video matting of complex scenes. In: Proc. of ACM SIGGRAPH, pp. 243–248 (2002)
21. Rother, C., Kolmogorov, V., Blake, A.: Grabcut - interactive foreground extraction using iterated graph cut. In: Proc. of ACM SIGGRAPH, pp. 309–314 (2004)
22. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence 23, 1222–1239 (2001)