TEMPORAL DETECTION AND PROCESSING OF TRANSPARENT OVERLAYS FOR VIDEO INDEXING AND ENHANCEMENT

Ahmet Ekin and Radu Jasinschi

Video Processing Group, Philips Research, Natlab, Eindhoven, The Netherlands
{ahmet.ekin, radu.jasinschi}@philips.com

ABSTRACT

This paper develops a theoretical model for the formation of transparent overlays and proposes a temporal algorithm to detect them independent of their degree of transparency. The proposed algorithm exploits our novel observation that the appearance of a transparent overlay results in a proportionally constant decrease in the intensity variance. In order to detect transparent regions, we first compute intensity variances about each pixel. After that, the ratios of the variances between the pixels of the consecutive frames are computed to form variance ratio images. Because the degree of transparency is unknown and may vary, we generate binary images by thresholding variance ratio images for every possible fine interval of the degree of transparency. Various morphological, textural, and contextual information are applied to every candidate binary image to detect spatial location of transparent overlays. We can also accurately detect the color and the degree of transparency of the transparent overlay so that we can remove the transparency or apply user-specific enhancement operations. We also demonstrate the application of the algorithm to video indexing and retrieval.

1. INTRODUCTION

During content editing, broadcasters overlay a variety of graphics, such as logos and text, to video to supplement it with additional information. The overlaid content is valuable to the viewer due to its information content, as it is in the case for subtitles, stock price tickers, and captions, and to the broadcasters due to its advertisement value, as it is in the case for logos and textual/graphics commercials. This requires easily readable and visually high-quality overlaid content. Furthermore, in some cases, simultaneous visibility of the actual video content and the overlaid content is desired. Transparent overlays are frequently used by production teams to serve both purposes: 1) Overlaid graphics that are superimposed on transparencies have higher readability and quality than those without transparencies [1], 2) Transparent overlays preserve the visibility of the original scene. Please note that the transparent overlays do not have any connotations to the actual transparency of the overlaid graphics. As in most cases, opaque graphics are added onto the transparent masks as shown in Figure 1.

Ironically, in certain situations, transparent overlays, which are introduced to increase the readability and the quality of overlaid content, may cause problems for video enhancement algorithms. If the video content behind the transparent mask is moving while text, graphics, and other overlaid content over the transparent overlay is stationary, motion estimation algorithms may produce inaccurate and inconsistent motion vectors. The incorrect motion vectors will

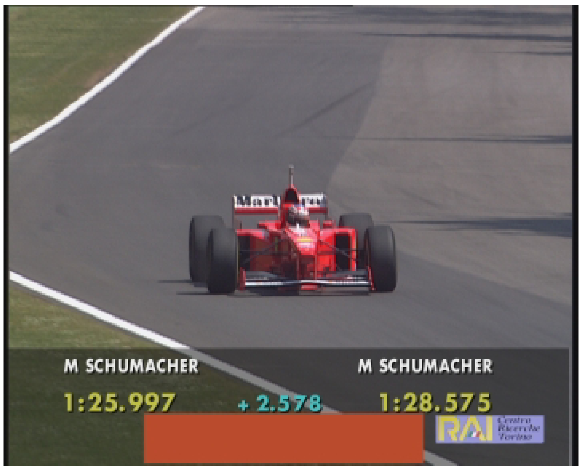

Figure 1: The use of transparency overlay mask to allow for the visibility of background and to increase the readability of text

degrade the performance of motion-based de-interlacing algorithms, such as the Philips Natural Motion [2]. Due to the common use of transparent regions and the severity of the degradations in the visual quality of the de-interlaced content, it is highly desirable to automatically detect the locations of transparent overlays in video and optimally enhance these regions. Although transparent overlays are frequently used, to the best of our knowledge, there is no algorithm in the literature that deals with the detection of these overlays. With regard to this, this paper proposes a novel algorithm to automatically detect transparent overlays in video.

The proposed algorithm is based on the observation that the overlay of a transparent region over an image region causes a proportionally constant drop in the local intensity variances within that region. We further prove that this constant is a function of the degree of transparency of the overlay. Despite the lack of relevant work on transparent overlay detection, we have been inspired by the work on gradual shot transition detection. Specifically, dissolve type transitions can be detected by modeling the temporal change in frame intensity variances [3] [4]. The main differences between that approach and the proposed transparent overlay detection algorithm are:

- Dissolve detection cannot be performed between two frames because reliable dissolve detection requires a large temporal window, usually a window length of at least 15 frames. In contrast, we detect transparent region by analyzing only two frames.

- Dissolve detection is thoroughly a temporal process, i.e., identification of the temporal location of dissolves, while transparent region detection is both spatial and temporal process. The proposed transparent overlay detection algorithm is able to locate both the temporal frame number of the transparent overlay appearance as well as the spatial location of the transparent overlay region.
- Dissolve detection algorithms use frame intensity variance, so they compute one variance value globally for the whole frame. In contrast, in order to detect spatial location of the transparent overlay, we compute intensity variance values locally, usually within a window around each pixel.

In Section 2, we first explain the novel variance-based model of transparent overlay appearance. Then, in Section 3, we describe the proposed detection algorithm. In Section 4, the applications of the proposed algorithm to video indexing and enhancement will be discussed. Section 5 presents the results and Section 6 concludes the paper.

2. VARIANCE-BASED TEMPORAL MODEL OF TRANSPARENT OVERLAY

A graphics-overlaid image can be considered as a mixture of the natural content and the graphics overlay. Equation 1 formulates the mixing process for transparent overlays. In Equation 1, I_{t+1} is the graphics overlaid intensity image at time *t+1,* I_T is the image with only graphics content, and I_N is the image with only natural content, i.e., I_{t+1} without any editing effects. The intensity value of I_{t+1} at (x,y) gets contribution from both graphics content and the natural content. Because the contribution of the graphics content to the pixel (x,y) is β, we define the degree of transparency at (x,y) as β. The parameter β assumes a value in the range $[0,1]$. The value of zero refers to no transparency, i.e., no graphics is added, whereas the value of one completely hides the natural content resulting in an opaque overlay.

$$I_{t+1}(x,y) = \beta \times I_T(x,y) + (1-\beta) \times I_N(x,y) \qquad (1)$$

If we consider I_N and I_T in Equation 1 as the content of two shots having a gradual transition (dissolve) from one to the other, Equation 1 can be considered as a snapshot of the dissolve process. Thus, β does not change by time in transparent overlay composition process whereas β is a function of time during shot transition process; one shot fades in while the other fades out. Furthermore, unlike shot-boundary detectors that attempt to detect temporal instants of shot boundaries, we need to detect the transparent overlays both spatially and temporally.

If we compute the intensity variance $\sigma_{t+1}{}^2(x,y)$, as the intensity variance within an $M \times N$ window centered about (x,y) of the image I_{t+1}, and do the same for I_T and I_N to obtain Equation 2:

$$\sigma_{t+1}{}^2(x,y) = \beta^2 \times \sigma_{I_T}{}^2(x,y) + (1-\beta)^2 \times \sigma_{I_N}{}^2(x,y) \qquad (2)$$

In most cases, transparent overlays are locally non-textured. As a result, the variance of the transparent color $\sigma_{I_T}{}^2(x,y)$ can be safely assumed to be zero. Substituting this in Equation 2 and taking the variance ratio between I_{t+1} and I_N:

$$\frac{\sigma_{t+1}{}^2(x,y)}{\sigma_{I_N}{}^2(x,y)} = (1-\beta)^2 \qquad (3)$$

where $0 \le \beta \le 1$ and $0 \le (1-\beta)^2 \le 1$. Equation 3 indicates that if there is a transparency, i.e., $0 < \beta < 1$, the appearance of a transparent region at frame *t+1* results in a proportionally constant decrease in the variance (the difference of the logarithms of the variances is constant). In general, I_N is not observed; however, it can be approximately computed by registering the previous frame I_t onto the current frame I_{t+1}.

3. DETECTION OF TRANSPARENT OVERLAY

Based on the theoretical model of the transparent overlays explained in Section 2, this section describes a detection algorithm. We follow the steps outlined below to detect transparent regions. Steps 1 to 4 are concerned with pre-processing, such as motion compensation and variance feature extraction. Step 5 is the detection stage with five sub-steps. Because we do not know β apriori, Step 5 involves the division of the expected range of $(1-\beta)^2$ in Equation 3 to multiple intervals and runs of the sub-steps for each interval.

1. Register the current frame I_{t+1} and the previous frame I_t by compensating for the motion between them
2. Identify the changed areas in the current frame by computing the absolute difference between the registered I_{t+1} and I_t
3. Compute the intensity variance value for each changed pixel, which is identified in the previous step, in I_t and I_{t+1} by using the neighboring pixel values. A window of size $M \times N$ is centered about each pixel to define the neighbors.
4. Compute the intensity variance ratio image, $V_{t+1}(x,y) = \dfrac{\sigma_{t+1}{}^2(x,y)}{\sigma_t{}^2(T(x,y)) + eps}$ where eps is a small number that is inserted to overcome numerical instabilities, and $T(x,y)$ maps the coordinate (x,y) in

I_t to the corresponding coordinate in I_{t+1} by the employed motion model (We use the two-parameter translational model)

5. Divide the expected range of $(1-\beta)^2$ in Equation 3 into k intervals, for example with equal-length and half–overlapping partitioning, and for each selected interval, follow the procedures below:
    1. Identify the pixels in V_{t+1} by assigning those whose variance ratio values are inside the current interval to one (shown white in Figure 3) and the others to zero (Figure 3 shows the pixel mask for one of the selected intervals, the original image is shown in Figure 2)
    2. In order to connect the split objects, apply morphological closing to the binary image computed in the previous step (Figure 4)
    3. Find connected-components, and remove the small-sized blobs to eliminate noise
    4. Compute the minimum bounding rectangle (MBR) and the *filling ratio* of each blob as a percentage of the number of white pixels within the blob MBR to the size of the MBR, and remove the blobs with lower filling ratio than a predefined number from further consideration. This step uses the observation that transparent overlays form compact blobs (Figure 5 shows the surviving blob, which is indeed a transparent overlay mask, after this stage)
    5. Verify the existence of horizontal and/or vertical edges at the boundaries of the blob based on the observation that a transparent overlay results in pronounced edges at the boundaries. For this purpose, compute the horizontal and vertical projections of the edge map for each blob, find the peak values of both projections, and then, verify that the peak location is close to the MBR coordinates and peak values are not less than 80% of the MBR height for horizontal projection and MBR width for vertical projection.

4. APPLICATIONS OF TRANSPARENT OVERLAY MASK DETECTION

Detection of transparent overlays can be useful for primarily two purposes: 1) video indexing and retrieval, and 2) video enhancement. In the next section, we first describe the possible uses of the proposed algorithm for video indexing and retrieval. As explained in the introduction, our main motivation is video enhancement. We will elaborate this in Section 4.2.

4.1 Video Indexing and Retrieval

Transparent overlays are displayed mainly to boost the readability of other graphical objects that are overlaid on them. As a result, they indicate the occurrences of other graphical objects, such as logos and video text. Because of the information content of those frames, they can be assigned as the key-frames of the video without even detecting text.

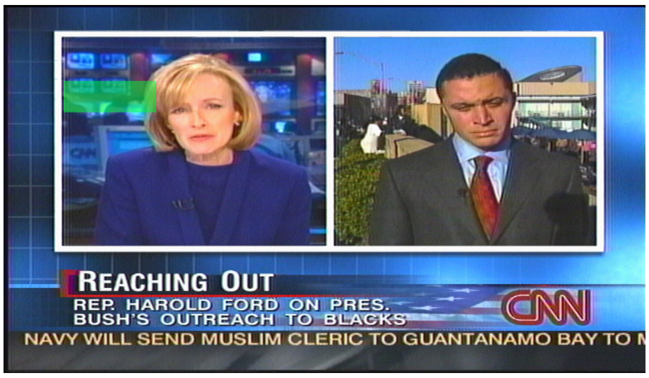
REACHING OUT
REP. HAROLD FORD ON PRES. BUSH'S OUTREACH TO BLACKS
CNN
NAVY WILL SEND MUSLIM CLERIC TO GUANTANAMO BAY TO

Figure 2: Transparent overlay added to the image (on the left of the anchor's head)

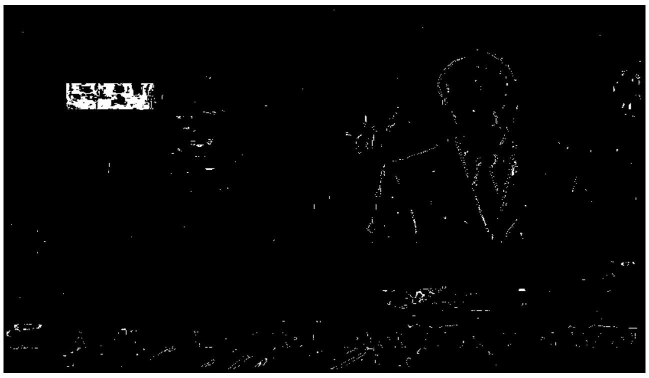

Figure 3: Pixels satisfying the variance ratio in the current interval

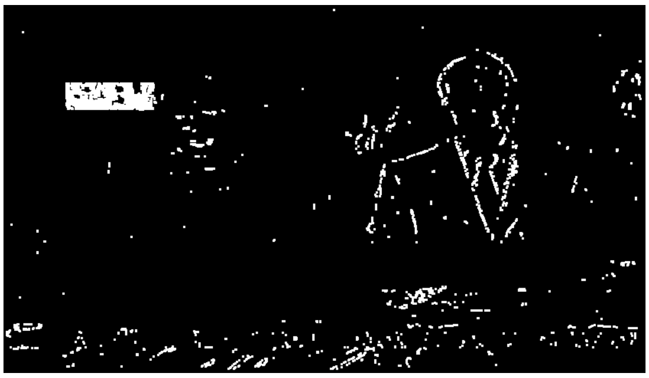

Figure 4: Closing operation applied to Figure 3

Because transparent overlays are used as background for video text, their detection can also be used as an additional cue for video text detection. Reducing the thresholds used in text detection algorithm within the transparent overlay regions may also improve the text detection accuracy.

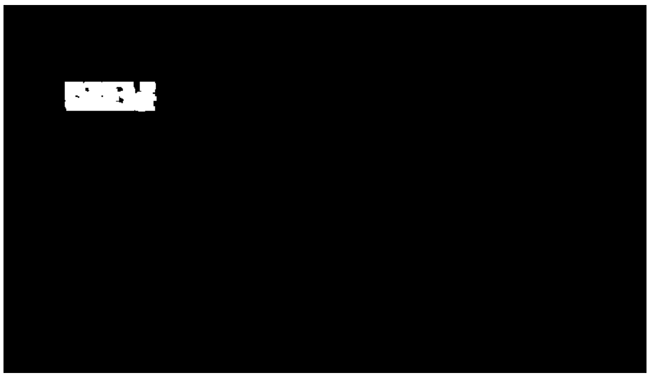

Figure 5: Detected transparent overlay region

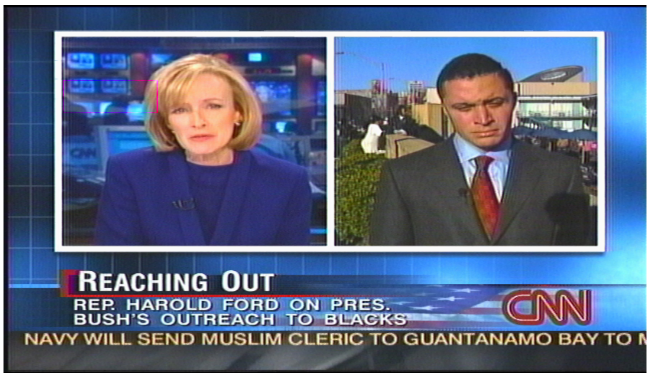
REACHING OUT
REP. HAROLD FORD ON PRES. BUSH'S OUTREACH TO BLACKS
CNN
NAVY WILL SEND MUSLIM CLERIC TO GUANTANAMO BAY TO M

Figure 6: Automatically removed transparent overlay

4.2 Video Enhancement

We envision two applications of the proposed algorithm in the domain of video enhancement. First, as explained in Section 1, the algorithm can be employed to update the confidence values of the motion vectors in transparent regions so that visual artifacts resulting from inconsistent and inaccurate motion vector estimation in transparent regions can be prevented. In the areas where transparent overlays are detected, we reduce the confidence of the motion vectors, and/or, according to the settings, the whole transparent region can be safely assumed to be stationary and de-interlaced.

A second video enhancement application relates to automatic modification of the transparency factor of the transparent mask. In this case, the transparency value β must be computed. For each detected transparent overlay, we solve Equation 1 to find β and I_T by using multiples of the pixels in the transparent overlay. We compute these two unknowns iteratively by first starting with an estimate of β, finding a least square solution for I_T and then updating β and continuing with the iteration until the difference between the current frame pixel values and the estimations based on β, I_T, and the previous frame becomes less than a certain threshold. Figure 6 demonstrates the result of transparent overlay removal from Figure 2.

5. RESULTS

Due to the difficulty of forming a database of natural video sequences that show the appearance of transparent overlays, we used synthetic sequences in our experiments. An advantage of the use of synthetic sequences is that the true values of β and I_T are already known.

In our experiments, we set k = 10 and used half overlapping windows. That is, each window of interval is of length 0.2 and is displaced 0.1 at each step to cover the expected range of $(1-\beta)^2$. We also changed β to find the limits of detectable transparency.

In over 20 sequences, the algorithm could detect transparent overlays with almost same accuracy (85 %) as we varied β from 0.2 to 0.5. β values greater than 0.5 resulted in the loss of transparency assumption.

In the current implementation, the location of the transparency is important. We observe degradations in the performance over a partially smooth area or an area where the transparency and background colors are close to each other. We are currently experimenting with a larger dataset to evaluate the dependency of the algorithm to the difference between the transparency and the background color.

6. CONCLUSIONS

We presented a novel model for transparent overlays and a robust image processing algorithm to realize the proposed model in practice. We have also demonstrated a video enhancement application by automatically detecting and removing the transparency. Two possible cases may require a revision of the proposed algorithm in spite of the validity of the theory. In the first case, the transparent overlays may be overlaid on a smooth surface that will nullify the method based on pixelwise division of the variances in Section 2. We can still detect such overlays, albeit with less accuracy, by modifying the intensity variance computation to energy computation around each pixel. The second case involves the text or graphics in the transparent overlay regions. For this case, the modification of the filling ratio threshold is necessary. It is also possible to couple the algorithm with a rough text detection algorithm such that text blocks, which can be identified as high-variance blocks, can be removed from further consideration and may be assumed as white regions in Step 4 of the algorithm in Section 3.

REFERENCES

[1] L. F. V. Scharff and A. J. Ahumada, "Predicting the readability of transparent text," *Journal of Vision,* vol. 2, no. 9, pp. 653-666, 2002.

[2] http://www.research.philips.com/newscenter/dossier/naturalmotion/index.html

[3] A.M. Alattar, "Detecting and compressing dissolve regions in video sequences with DVI multimedia image compression algorithm," in *Proc. IEEE ISCAS*, vol. IV, pp.13–16, 1993.

[4] Rainer Lienhart. "Reliable Dissolve Detection," In *Storage and Retrieval for Media Databases 2001*, Proc. SPIE 4315, pp. 219-230, Jan. 2001.