Automatic, Efficient, Temporally-Coherent Video Enhancement for Large Scale Applications

Anonymous
Anonymous Institution

Anonymous
Anonymous Address

ABSTRACT

A fast and robust method for video contrast enhancement is presented. The method uses the histogram of each frame, along with upper and lower bounds computed per shot in order to enhance the current frame. This ensures that the artifacts introduced during the enhancement is reduced to a minimum. Traditional methods that do not compute per-shot estimates tend to over-enhance parts of the video such as fades and transitions. Our method does not suffer from this problem, which is essential for a fully automatic algorithm. We present the parameters for our methods which yielded the best human feedback, which showed that out of 208 videos, 203 were enhanced, while the remaining 5 were of too poor quality to be enhanced. Additionally, we present a visual comparison of our work with the recently-proposed Weighted Thresholded Histogram Equalization (WTHE) algorithm [6].

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.5.1 [**Information Systems and Presentation**]: Multimedia Information Systems

Keywords

video enhancement, temporal coherence, histogram equalization

1. INTRODUCTION

Histogram stretching and histogram equalization are basic image processing techniques which have been widely applied to images. Videos may contain transitions and other special effects, such as fade-in/fade-out. Applying any image enhancement algorithm on these portions of the video yields undesirable effects, such as blocking, and increasing chroma noise, in the end destroying the effect that the video creator intended.

An algorithm that can be applied to a very large number of videos needs to satisfy several stringent requirements: 1) It has to be *very fast*, since it is important to be able to handle a large volume of videos with as few machines as possible; 2) It has to be *robust* – the cost of degrading the quality of a video is high – since the users expect an enhancement; 3) The algorithm must be *automatic*, meaning that the parameters have to be consistent across many videos as human intervention is not feasible. These requirements mean that any algorithm that applies local operators (spatio-temporal) cannot be used, as they tend to use a large amount of processing power. Therefore, global enhancement algorithms can be used.

The closest published approach to what we propose here was developed by Qing and Ward [6]. Their approach, however is susceptible to corrupting video transition effects, and introducing feedback loops when the input video contains content recorded with cameras that employ automatic gain control. Zheng and Liao [7] use the HSL color space, in which they perform the equalization the L component, and adjust the S separately, while having a relatively simple coherency metric. We will discuss a more consistent metric which does not depend on the underlying encoding of the video. Unlike Mittal *et. al.* [3], we found that using the *La*b**, and simply enhancing the *L* component yields poorer quality video than using HSV due to out of gamut problems which we counter due to the fact that the color space is defined assuming a CIE D50 illuminant [2]. This assumption does not hold for a large number of videos, since not all videos are shot using bright sun light. Moreover, digitally created videos have RGB values that do not have any physical correspondence, thus making the mapping to the *La*b** color space meaningless.

The task of enhancing video is challenging due to the following reasons: 1) Videos may contain various *edits* that can “fool” enhancement algorithms. For example, the user may want a portion of the video to look as if it were filmed with a webcam in a low light environment. This will be corrected by the algorithm, making the user unhappy. 2) Many videos contain *multiple shots*. Each shot will need to be treated separately from the rest of the video. However, shot detection is still not a solved problem. Bad shot boundaries may yield bad output. 3) *Transitions* are hard to handle: fade-in/fade-out effects must not be corrected, and yet, if any enhancement algorithm analyzes each frame individually, it will try to enhance them, thus removing the effect. While doing this, it will likely emphasize compression artifacts present in the darker frames of the sequence. 4) *Automatic gain control* is present on most consumer cameras. This causes many enhancement algorithms to introduce flickering (in the best case, making it out of phase), since the statistics of the input frames change rapidly over time, while the content of the frame remains relatively static.

2. METHODS

Formally, an image (or video frame) is represented as RGB triplets mapped onto a 2D grid: $I(x, y) \in \{0, 1, ..., N-1\}^3$, where $x \in \{0, ..., width-1\}$, and $y \in \{0, ..., height-1\}$. N defines the number of discrete colors allowed in the image. Most commonly, video frames have $N = 255$. In the HSV color space, the V component is defined as the maximum of the RGB values, thus $V(x, y) = \max(I(x, y))$ (element-wise). The probability mass function (PMF) for V is:

$$P(k) = Pr\left(V(x, y) = k\right), \sum_k P(k) = 1 \tag{1}$$

The cumulative distribution function (CDF) of V is:

$$C(k) = \sum_{i=0}^{k} P(i), k \in \{0, .., N\} \tag{2}$$

A simple form of Histogram Equalization, often referred by image editing software as "*Auto-Levels*", computes a linear transform for all colors in an image. The name derives from the fact that the algorithm tries to find a "level" value to which the white color should be mapped, and another value for black:

$$l_{out} = \frac{l_{in} - b_{est}}{w_{est} - b_{est}}(w_d - b_d) + b_d, \tag{3}$$

where l_{out} is the output luminance, l_{in} is the input luminance, b_{est} is the estimated black level, w_{est} is the estimated white level, w_d is the desired white level, and b_d is the desired black level. The luminance is usually computed as defined by the YUV, HSL or *La*b** formulations. Typically these values are chosen such that the output will cover the entire dynamic range of the color space in which the adjustment is made. Given a luminance, the advantage of this formulation is that it is color-space independent (the exact formulation does not really matter), and it is very fast as it is simply a linear transform.

Using the formulation from Eq. 2, a good estimate for w_{est} is such that $C(w_{est}) = 0.995$ and b_{est} is chosen such that $C(b_{est}) = 0.005$. This formulation would assign w_{est} and b_{est} values in the set $\{0, .., N\}$. It is more convenient, though to work with values in the [0..1] interval, and as such, we will assume that the variables we work with are normalized to this interval. The formulation for luminosity that we employ is the "value" component in the HSV color space. This decision was made experimentally, as it yielded the best enhancement results overall, and it is impossible to generate out of gamut colors by adjusting V.

Naïvely applying this formulation independently on each frame of the video is not desired, as in many cases the output will have a noticeable flicker due to jumps between nearby estimates of $b_{est}$and w_{est}. In order to alleviate this problem, we can compute these values per frame, but not use them directly.

Using a shot (or cut) detector [5], a better estimate can be computed based on the statistics of the entire shot. In order to have a conservative estimate that will not severely change outlier frames, it is possible to to use the 95^{th} percentile for w_{shot} and the 5^{th} percentile for b_{shot}. Simply using these values will yield a reasonable output, but unfortunately, the output will be too conservative to be practical. An alternative approach [6] is to use a moving average of w and b over a window of frames. This has the side effect of "delaying" fades, and enhancing flickering on videos that were filmed with cameras that employ automatic gain control, since the statistics will vary across the same shot.

A compromise between the estimates at frame level, denoted as b_{frame} and w_{frame}, and the shot level (b_{shot} and w_{shot}) can be obtained by:

$$\alpha = [1 - \mid b_{shot} - b_{frame} \mid]^{\gamma_{frame}}, \tag{4}$$

$$b_{est} = \alpha b_{frame} + (1 - \alpha) b_{shot}. \tag{5}$$

Using the same pattern, w_{est} can also be computed. The parameter γ_{frame} is used to decide how much influence does the local estimate have over the more global shot estimate. This formulation allows more frames to be corrected when compared to the per-shot parameters. At the same time, a smaller number of frames will be adjusted than when using the per-frame statistics. A good choice for γ_{frame} was found to be 0.5.

The algorithm thus far will work well in the general case. However, there is a class of frames for which it will not do very well: whenever there is a high contrast between the foreground and the background, and the foreground object is "dark", the above algorithm will make the object perceptually darker, and it will further increase the contrast. However, to the human eye, the end result will "seem" lower quality [1], because it is now much harder to distinguish any features in the dark object. In order to alleviate this problem, if the corrected color has a lower value than the original, then the output is computed as follows:

$$\delta_l = l_{out} - l_{in}, \tag{6}$$

$$l'_{out} = \begin{cases} l_{in} - \mid \delta_l \mid^{\gamma_l} & if \delta_l < 0; \\ l_{out} & otherwise \end{cases} \tag{7}$$

γ_l is a parameter which can be adjusted by the user for the desired output. If it set to one, then the default algorithm is applied. Experimentally, we found that 4 is a good choice of γ_l.

The final luminance value of the pixel, $\tau(l'_{out})$ is defined as:

$$\tau(v) = \begin{cases} 0 & \text{if } v < 0; \\ v & \text{if } 0 \leq v \leq 1; \\ 1 & \text{if } v > 1. \end{cases} \tag{8}$$

An intriguing aspect of video enhancement is the Abney effect [4]. An increase in lightness causes a perceived decrease in saturation. Therefore, in order to preserve the perceived saturation, we modify the saturation channel s_{out} as follows:

$$s_{out} = \tau\left(s_{in} + \sigma\delta_l\right), \tag{9}$$

where σ is a parameter, which yields a good compromise in enhancement at 0.15. This unique formulation of our algorithm modifies saturation as a function of luminosity difference in order to increase the perceived clarity of the image.

Some videos may not contain enough information for our algorithm to be effective. This class of videos can easily be detected. If the dynamic range of the luminance is too small, then no global enhancement algorithm can work. Experimentally, we found that if the input has fewer than ten unique luminance values, the output will likely not be visually pleasing (although, it may present more details).

Currently, no shot detector that can segment the video with 100% accuracy, as evidenced by the continuing TRECVid challenge for video shot boundary detection. Since our algorithm uses a blend of local and shot estimates, it works well as long as the shot estimate isn't corrupted. The shot estimate is corrupted only if it is calculated over the span of two statistically different shots. Thus, if the shot detector is calibrate such that it has high recall, the fact that it has low precision does not matter with respect to the output our algorithm generates.

In order to prevent frames that have a very low dynamic range from being processed, if the number of distinct V values in a frame is less than 20, the frame is ignored.

3. EXPERIMENTAL RESULTS

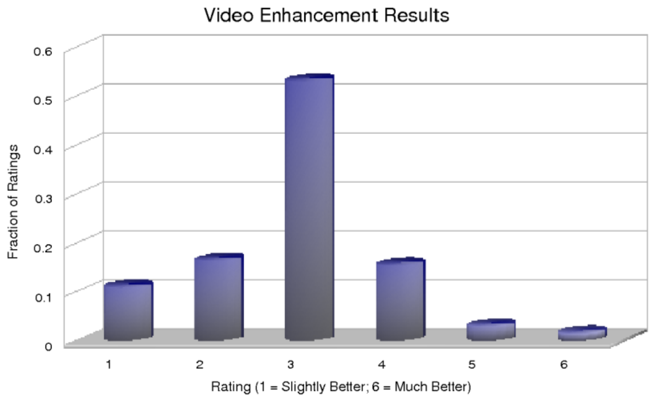

Figure 1: Aggregate human evaluation results.

In order to quantify the effectiveness of the video enhancement algorithm, we designed a human evaluation experiment. We chose to do this over using other metrics because it is very difficult to automatically determine whether artifacts are generated as a result of the enhancement, especially if the artifacts are subtle.

We ran the algorithm over a corpus of 1000 YouTube videos out of which we selected 208 that had at least one frame with a RMS of over 10% between the original and the enhanced version. The enhanced video and the original video were played side-by-side while being synchronized at the frame level. The location of the enhanced video was chosen randomly for each video pair.

The raters were asked to choose which side looks better overall, and to note which side had artifacts. The available ratings ranged from "Much worse", equating a score of -2, "No change" having a score of 0, and "Much better" having a score of 2. In order to reduce rater bias, we used 208 raters, with each rater being presented 3 random video pairs. At the end of the experiment, each video was rated by 3 raters. For each video, we compute a score that is the sum of all 3 scores, the summary of which is depicted in Fig. 1.

203 enhanced videos obtained a positive rating. 5 had a combined rating of 0 (two raters had conflicting opinions, and one was neutral). The enhancement algorithm emphasized the outline of the already existing blocking artifacts, making half raters choose the original. However, the decision was not clear cut, as other raters chose the enhanced version for the same videos. The overall rater agreement for the experiment was 0.75.

In addition to the overall evaluation, we also analyzed the comments from the raters. The comments for the enhanced version did not imply that any new artifacts were introduced in the video. This was a crucial finding, as a large number of videos on YouTube contain fade effects and wide variety of transitions that tend to make frame-based algorithms fail.

In order to qualitatively measure the performance of our algorithm we present frames enhanced using our algorithm. Fig.2 depicts a frame taken from a fade transition. The user expectation is that fades be preserved. Our algorithm correctly preserves the fade. A per-frame enhancement algorithm (Fig.2(b) completely removes the transition. Fig.3 depicts a typical enhancement result of our algorithm applied on a frame that is not part of a transition. Additionally, Fig.3(e) contains the WTHE [6] output.

We implemented the algorithm in C++. This base implementation, which does not use lookup tables, processes 640x480 video over three times faster than real time on an Intel$^{®}$ Core^{TM}2 Duo 6600. This includes the decoding of the video MPEG4 stream, processing, and encoding it into an output file. It is possible to speed up the algorithm by computing a per shot lookup table for the value-saturation pairs.

4. CONCLUSION

We presented an algorithm that is capable of enhancing the quality a large number of videos, while not decreasing the quality of any videos it processes. This algorithm can be directly applied on a large scale for mass-processing of videos with little to no human supervision. Although our algorithm has parameters which could be tuned by humans, we presented the best enhancement parameters which we found experimentally.

5. REFERENCES

[1] A. Calabria and M. Fairchild. Perceived image contrast and observer preference: II. empirical modeling of perceived image contrast and observer preference data. *The Journal of imaging science and technology*, 47(6):494–508, 2003.

[2] D. Judd, D. MacAdam, and G. Wyszecki. Spectral distribution of typical daylight as a function of correlated color temperature. *Journal of the Optical Society of America*, 54(8):1031–1040, 1964.

[3] G. Mittal, S. Locharam, S. Sasi, G. R. Shaffer, and A. K. Kumar. An efficient video enhancement method using La*b* analysis. In *Proc. IEEE International Conference on Video and Signal Based Surveillance*, page 66, Washington, DC, USA, 2006. IEEE Computer Society.

[4] R. Pridmore. Color variability of wine in a glass and general comments on contrast effects. *Color Research and Application*, 2(30):146–149, 2005.

[5] J.-C. Ren, J. Jiang, and J. Chen. Determination of shot boundary in MPEG videos for TRECVID 2007. In *TREC Video Retrieval Evaluation Online Proceedings*, 2007.

[6] Q. Wang and R. Ward. Fast image/video contrast enhancement based on weighted thresholded histogram equalization. *IEEE Transactions on Consumer Electronics*, 53(2):757–764, May 2007.

[7] Y.-C. Zeng and H.-Y. Liao. Video enhancement based on saturation adjustment and contrast enhancement. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 3550–3553, May 2008.

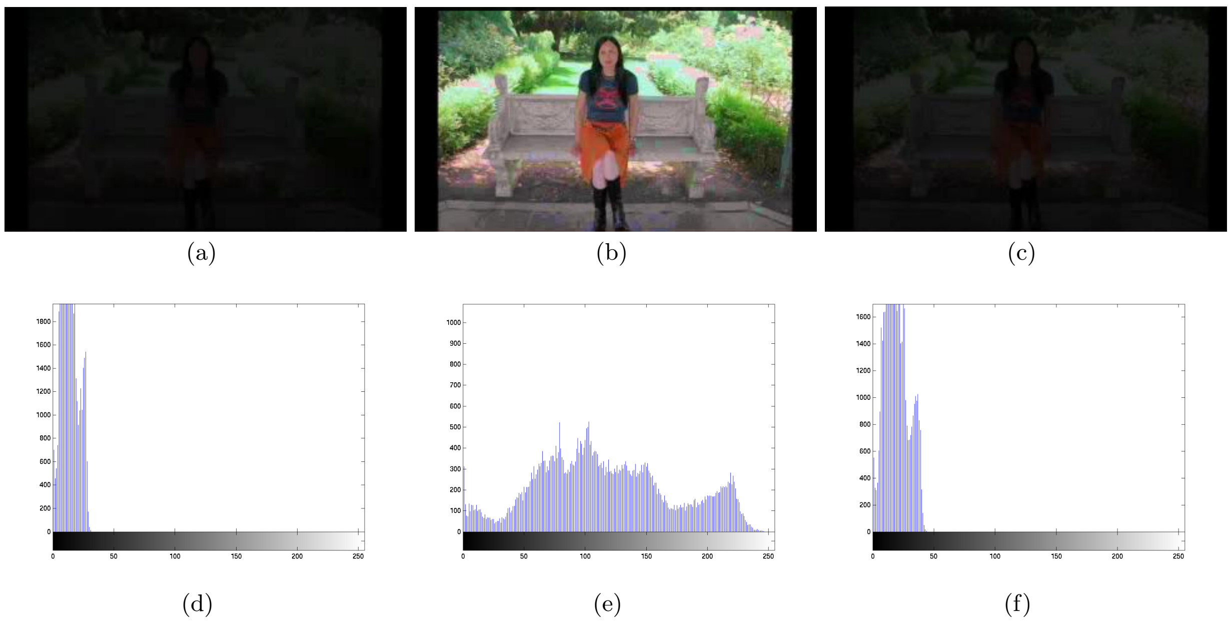

Figure 2: Example frame from a fade-in transition. A consistent enhancement algorithm should not change this frame much, as it is important to preserve the video author's intent: (a) Original frame; (b) HSV-enhanced frame (the entire transition is destroyed); (c) Temporally-smoothed HSV enhanced frame using our algorithm. Notice that the fade is still present as the output is dark. The corresponding luminance histograms are depicted in (d),(e), and (f).

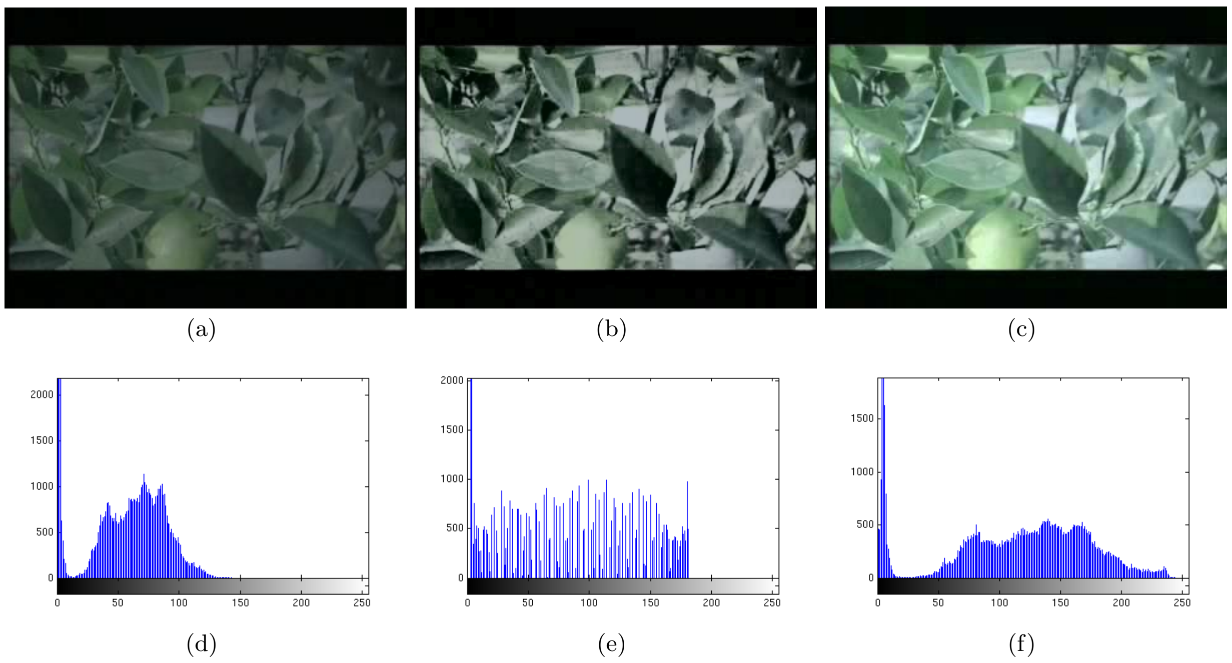

Figure 3: (a) Original frame and (d) its luminance histogram; (b) The Weighted Thresholded Histogram Equalization with parameters found through a grid search of the r and v with the corresponding histogram (e); (c) Our algorithm with the parameters presented in this paper, and its histogram (f). Notice that our algorithm produces a much more visually pleasing image. This is due to the fact that while increasing the overall luminance of the image, we also increase the saturation.