Automatic Colour Enhancement and Scene Change Detection of Digital Video

Korpi-Anttila, J.
Helsinki University of Technology, Department of Automation and System Technology, Laboratory of Media Technology
P.O.Box 6400, FIN-02015 HUT, Finland, http://www.media.hut.fi
E-mail: jyrki.korpi-anttila@hut.fi

ABSTRACT

Basically digital video is a sequence of still images, displayed at a constant frame rate. Simplest adaptation of still image colour correction algorithm into the digital video is to use the same algorithm frame by frame in the video sequence. However, this kind of approach does not lead to satisfactory results. The sources of the problems are the temporal continuity of the video frames and different gradual transitions, for example editing effects and camera and object movement. Visual defects, which appear in the corrected video are for example flickering, hue changes, brightness changes and edit effect cancellations. These defects are the result of the still image algorithm, which has internally different kinds of thresholds, which adjust correction parameters. There are always small changes in the video sequence. The problems occur when there are small changes in the original video and the changes are still large enough for the correction algorithm to produce little differently corrected image. Even though the changes in the original video are unnoticeable, the changes in the corrected video may be visually noticeable.

The temporal continuity of the correction must be made certain and solution is utilizing scene change detection. The object is to detect scene changes and assure that single scene is corrected without any large changes of the correction parameters. The original contrast enhancement and colour balance correction algorithms were based on the RGB-histograms and parameters calculated from the histograms, for example white and black points of the image. It is important that the scene change detection algorithm detects the changes, which results to the eventual changes of the actual correction parameters. In this work, the scene change detection is based on the wavelet transformed histograms. The wavelet coefficients of the histograms of the consecutive frames are compared to calculate difference of the frames. Also separate edit effect detection based on the histogram distribution movement is conducted. Overall result of the scene change detection module is a value between [0, 1], which represents the probability of the scene change.

The final correction parameters are the combination of the parameters calculated from the current frame and the parameters used to correct previous frames, i.e. the correction parameters are adaptive recursive filtered before the correction is made. The probability of the scene change is used as an adaptive filter coefficient. If the probability is high then the weight of the parameters calculated from the current frame is significant. If the probability is low then the temporal continuity of the correction is assured by using larger weights for the parameters used in the previous frames.

The still image algorithms may be used in the digital video if temporal continuity of the correction is ensured. The filtering of the correction parameters based on the scene change detection is proved to be workable solution. Visual tests of the application presented in this study
have indicated that most of the defects, which arise when the correction is made only frame by frame, can be eliminated.

INTRODUCTION

A research project to develop automatic algorithms to enhance visual quality of colour images, has been active for several years at the Laboratory of Media Technology. The results of this project include many algorithms for contrast enhancement, colour balance correction, sharpening, noise suppression and skin tone correction [1], [2], [3]. These algorithms were originally developed for still images only. First real time prototype application of contrast enhancement and colour balance correction for video was implemented in the Laboratory of Media Technology in 1997. The application used still image algorithms as such, calculating corrections frame by frame without taking into account any temporal dependencies between adjacent frames [4].

The behavior of different colour enhancement algorithms when applied to moving pictures has not been widely studied. The purpose of this work [5] is to address problems, which arise when these algorithms are used in moving pictures. Implementations of automatic contrast enhancement and colour balance correction of moving pictures are presented. These utilize adaptive recursive filtering of correction parameters according to accurate, histogram based scene change and transition effect detection.

CONTRAST ENHANCEMENT AND COLOUR CORRECTION

The principal objective of enhancement techniques is to process a given image so that the result is more suitable than the original for a specific application. In this application the objective is to enhance the visual quality of digital video. The original application for contrast enhancement and colour correction algorithm was simple still image. A brief overview of the original still image algorithm is given, before the actual application of moving pictures is presented.

Contrast stretching (often called normalization) is a simple image enhancement technique that attempts to improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values, e.g. the full range of pixel values that the image type concerned allows. It differs from the more sophisticated histogram equalization in that it can only apply a linear scaling function to the image pixel values. As a result the `enhancement' is less harsh.

Before the stretching can be performed it is necessary to specify the upper and lower pixel value limits over which the image is to be normalized. Often these limits will just be the minimum and maximum pixel values that the image type concerned allows. For example for 8-bit gray-level images the lower and upper limits might be 0 and 255. Call the lower and the upper limits \(a\) and \(b\) respectively.

The simplest sort of normalization then scans the image to find the lowest and highest pixel values currently present in the image. Call these \(c\) and \(d\). Then each pixel \(P\) is scaled using the following function

\[
p_{\text{out}} = \left(p_{\text{in}} - c\right) \left(\frac{b - a}{d - c}\right) + a
\]

(1)

Values below 0 are set to 0 and values above 255 are set to 255.
The problem with this is that a single outlying pixel with either a very high or very low value can severely affect the value of $c$ or $d$ and this could lead to very unrepresentative scaling. Therefore a more robust approach is to first take a histogram of the image, and then select $c$ and $d$ at, say, the 5th and 95th percentile in the histogram (that is, 5% of the pixel in the histogram will have values lower than $c$, and 5% of the pixels will have values higher than $d$). This prevents outliers affecting the scaling so much.

Another common technique for dealing with outliers is to use the intensity histogram to find the most popular intensity level in an image (i.e. the histogram peak) and then define a cutoff fraction, which is the minimum fraction of this peak magnitude below which data will be ignored. The intensity histogram is then scanned upward from 0 until the first intensity value with contents above the cutoff fraction. This defines $c$. Similarly, the intensity histogram is then scanned downward from 255 until the first intensity value with contents above the cutoff fraction. This defines $d$.

The original still image algorithm includes similar linear adjustment described above for each colour component. This is basically a stretching of the dynamic range to maximum. However, instead of trivially mapping the histogram endpoints to 0 and 1, a more careful analysis is employed to prevent over adjusting. When the adjustment curves are the same for each component, hues are approximately preserved, i.e. there is "contrast adjustment" only (this does not hold exactly because the target gamma is not unity). When the curves differ from each other, hues are also altered, so there is ‘colour balance adjustment’ as well. Either contrast or colour balance adjustment, or both, may be moderated depending on the image. The basic features for determining the adjustment curves are:

- The essential minima and maxima of the colour components,
- The "black" and "white" points, i.e. the darkest and brightest colours in the image.

**STILL IMAGE ALGORITHM AND MOVING PICTURES**

Basically digital video is a sequence of still images, displayed at a constant frame rate (for example 25 frames/second). Simplest adaptation of still image colour correction algorithm to use in moving picture, is to use same algorithm frame by frame in video sequence.

The viewer of the corrected still images does not have any reference images when the corrected image is viewed (maybe only the original). In moving pictures the situation is very much different. All previous frames are considered as a reference to the current frame. If current frame is corrected even slightly differently than the previous one (when scene content is considered constant), different kinds of very noticeable “flash” effects will appear in the corrected video.

Also different transitions and edit effects are problematic for the automatic colour enhancement. For example fade-out and fade-in effects where a scene gets darker or brighter. In the best situation, contrast enhancement corrects these transitions in a way that transition time is much faster (scene gets dark abruptly instead of constant fade-out). An even worse situation may arise when correction produce abrupt brightening in a middle of fade-out. This is seen as a very distinct flash in a corrected movie.

Frequently used edit effect in the movie edition is cross-fade. This may be very problematic for automatic colour correction. If the adaptation of the colour correction algorithm is not good, this
may lead to very curious corrections in the neighborhood of cross-fade effect, because of contribution of colour content of both previous and current scenes, in the middle of the transition.

Another typical problem, which is characteristic for automatic colour enhancement algorithms for moving pictures, is gradual scene changes, for example camera pans and moving objects. If a colour correction algorithm does not maintain continuity through scene, some hue changes may appear. For example, a bouncing red ball should not change the hue of green grass in the middle of the scene.

Simple still image colour correction algorithm must be enhanced before it can be applied to moving pictures. The main requirements are the following:

- Correction must be stable, if there are only small scene changes
- However, adaptation of correction must be quick in abrupt scene changes, for example in direct cut, and
- Correction must handle different edit effects (fades etc.) without canceling or making other defects in transitions

**SCENE CHANGE DETECTION**

Problems discussed in the previous section, are solved in this study with scene change and edit effect detection. Many scene change detection are known in the literature, but none of them seemed to be suitable as such, to this problem. It is to be observed, that scene changes are not necessarily the same, which are designed by film director. Because colour correction algorithms in this application are based on RGB-histograms, the interesting scene changes are those in which histogram content of two successive frames has influence on the colour correction. If two successive scenes have similar histogram content, the actual colour correction does not change and scene change is not important for correction purposes.

The scene change detection method presented in this work is based on RGB-histograms and can be described as a hybrid of wavelet analysis and statistical goodness of fit test.

The problem which is common to statistical goodness of fit tests is that they give only a difference estimate which is calculated as a sum of bin-wise differences. The problem is illustrated in Figure 1. The result of the bin-wise goodness of fit test is that the difference between histograms 1 and 2 is (almost) the same as the difference between histograms 1 and 3. If images of such histograms are viewed side by side, of course the image of histogram 1 (black) differs more from the image of histogram 3 (white) than from the image of histogram 2 (gray).

![Figure 1](image)  
Problem of statistical goodness of fit tests.
One very good solution to overcome this problem is the wavelet (multiresolution) analysis. The idea behind this is that, by analyzing wavelet decomposition coefficients at different levels, the multiresolutional information about histogram content is obtained. Figure 2 shows that these three histograms can be easily differentiated at wavelet decomposition level 8.

![Figure 2](image)

**Figure 2** Wavelet decompositions of histograms at levels 8 and 6.

The big difference of the histograms at high wavelet decomposition levels indicates high probability of scene change.

The used wavelet basis, in this application, is Haar-wavelet, which is the simplest form of wavelets. The lowpass filter used in Haar-transform is a simple average of two adjacent samples. With 256 samples long histogram, it is possible to get 8 levels of wavelet decompositions. The first level consists of samples as such. The second level is 128 samples long, containing averages of two adjacent samples, and so on. Finally the 8th level contains two averages of 128 samples.

The difference value between wavelet decompositions of two successive frame histograms is calculated using simple statistical log-likelihood test.

\[
F_a = \sum_{n=0}^{N-1} A_n \log A_n, \quad F_b = \sum_{n=0}^{N-1} B_n \log B_n, \quad S = \sum_{n=0}^{N-1} (A_n + B_n) \log (A_n + B_n)
\]

(2)

Overall, the eight difference values represent the differences of all levels of the wavelet decompositions of the histograms. The final difference between the frames is calculated as a weighted sum of these eight decomposition differences. It is clear, that the difference values of higher levels of decompositions are more important than the lower ones. Weights are increased exponentially towards higher levels. The steepness of this exponential weight curve is adjusted by the effective width of histograms. The reason for this lies in the situations when histograms are very narrow but near each other. It is possible that small local disturbances in histogram content may result in high difference values at low levels of decompositions compared to higher ones, although there is no scene change.
Examples of decomposition difference weight coefficients (percentage), at different histogram widths are illustrated in Figure 3 (notice the logarithmic scale).

Before wavelet decomposition calculations are made, 5 percent of histograms ‘dynamic range’ is first clipped from bottom.

The reason for this is that, this way small disturbances in histogram content are eliminated, especially occurring in narrow histograms, and this also helps the calculation of effective width of histogram. After bottom clip, effective histogram width is calculated simply from 5 percent and 95 percent points of the histogram.

Weighted sum of wavelet decompositions difference values is calculated from all three RGB-histograms, and the final frame scene change value is calculated as a weighted sum of all three histogram differences.

\[
diff = 0.2125 D_R + 0.7154 D_G + 0.0721 D_B
\]  

(3)
Finally, a value between $[0, 1]$, which represents the probability of scene change ($PSC$) is calculated. According to our practical tests, quite reasonable value is

$$PSC = \min(1, 10 \cdot \text{diff})$$

(4)

This means that all difference values above 0.1 are treated as a certain scene cuts ($PSC$ is clipped to 1.0). The flow chart of the scene detection algorithm is presented in the thesis [5].

**TRANSITION AND EDIT EFFECT DETECTION**

The three basic transitions, that cause problems in automatic colour correction are fade-out, fade-in and cross-fade. During fade out and fade in scene is darkening or brightening gradually. This can be detected quite easily from tendency of the histograms to float towards black or white, simultaneously the dynamic range is either decreasing or increasing (histograms get narrower or wider). A simple detector, which observes tendency of movement directions and positions of histograms 5%, 50% and 95% points, proved to be suitable for this purpose. Cross-fade is treated as normal gradual scene change and it is not detected separately.

**Figure 5** Example of output of scene change detection.

**Figure 6** Histogram shrinkage in fade out.
**ADAPTIVE RECURSIVE FILTERING**

The still image colour correction algorithm is based on essential minima and maxima of the colour components and black and white points, calculated from RGB-histograms. The solution adopting still image algorithm in the video correction application is the adaptive recursive filtering of these four points according to the changes detected in the video stream. The filter equation is:

\[
C_n = W_x C_{curr} + (1 - W_x) C_{n-1}
\]

where

- \(C_{curr}\) is the parameter calculated from current frame alone
- \(C_{n-1}\) is the previous filtered parameter
- \(C_n\) is the current filtered parameter
- \(W_x\) is the adjusted filter coefficient

There are two different adaptive filter coefficients, one for filtering dark-end \((W_b)\) and another for bright-end \((W_w)\) points. In a normal situation both filter coefficients are the same as the scene change value, but in some detected transition (edit effect) states, the coefficients are adjusted. For example in fade-in, contrast enhancement is very critical, especially at the bright end, how the correction is made. If bright end is driven very fast at the maximum level, all headroom is lost, and correction is clipped when original fade-in continues brightening. Also the gradual fade-in effect is cancelled if the correction is too fast. On the other hand, severe flickering will arise, if there are even slight changes in constant brightening.

![Figure 7](image_url)  
Contrast enhancement in fade-in.

The clipping is not a problem in fade-outs, but flickering and the effect canceling problems are still valid, if correction is not gradual or is too strong. In detected fade-in and fade-out situations the filter coefficients are reduced from simple scene change value. Especially the bright-end filter coefficient is reduced considerably.

As can be seen from equation (5), the adaptive recursive filtering takes care automatically also direct cuts. In direct cut, the scene change value is 1, thus filtering equation discards all previous information in the recursion. Only information collected from current frame is used.
As all four RGB-points, crucial to the colour correction algorithm, are filtered, actual colour correction curves are calculated same way as in still image algorithm. To ensure continuity of correction, the endpoints of resulting correction curves are also filtered using the same filter coefficients. Thus, the input parameters and output curves of colour correction are adaptive recursive filtered before correction is made.

**IMPLEMENTATION AND APPLICATIONS**

Automatic contrast enhancement and colour correction of moving pictures, presented in this study, was implemented as an Adobe Premiere 4.2 filter plug-in. One starting point for development of the algorithms was demand for causality and fast processing. Of course scene change and transition detection could be much more accurate, if some amount of delay is allowed. In off-line environments, even infinite delay (all future frames are known) can be used.

The implementation of colour correction is causal (only current and past frames are used). Scene change detection and calculations of correction parameters are based completely on RGB-histograms, which makes the algorithm computational efficient. Most of the processing time is spent in calculating the histograms and making the final correction transform in the frame buffer. With slight optimizations, correction of full-resolution (640×480, 24bit, 25 frames/s) video can be calculated in real-time with current top PC-processors (1GHz PIII).

The presented colour enhancement scheme could be implemented as such in any other platform, which uses RGB-frame buffer to store video frames. For example digital TV receiver or set-top box would be ideal platform. The correction could be made also in the broadcasting end of the video production, for example in the digital video camera or in the video production board.

The Adobe Premiere is ideal platform for retail consumer application. Typical purpose of use could be situation when consumer takes private video shots with digital camcoder. Typically, peoples photographing skills are not very good, which results to the bad quality in the final movie. The quality could be noticeably improved using this kind of automatic enhancement application, which could be used without any further knowledge of photographing. Another typical application could be news reports. For example in live breaking news situation, there is not time to do any manual quality enhancement operations for the live video material. Also quality in the in the live sports broadcasts is often poor. In these kinds of situations automatic enhancement system is useful even in the professional broadcasting environment.

**EXPERIMENTAL RESULTS**

The original still image correction includes algorithms for colour balance correction, contrast enhancement, skin colour correction and noise suppression. In the previous study [1], the performance was tested with 15 colour images from different sources. The test results indicated that overall visual quality was improved significantly. The only problem observed was increase of visual noise level in some corrected pictures. The reason for this phenomenon was contrast enhancement, which also increases the noise in dark areas in the image [1].

The performance of the Adobe Premiere 4.2 automatic contrast enhancement and colour correction filter plug-in was tested with a movie set containing 15 different movies [6]. The test set contained different class of movies: sports, music video, cartoon, news clips, sitcom, soap etc. Also random transitions were added into test clips to test performance of correction in these
transitions. The different transition types added were: dissolves, cross dissolves, wipes and different block transforms. Original and corrected movie was displayed side by side and test was performed by visual observation by the author. This was considered sufficient because only qualitative results were sought. The objective was to confirm if overall visual quality of the corrected movie is better and also to detect small defects produced by enhancement algorithm.

In nine movies out of fifteen, the overall quality in corrected movie was clearly better than in the original. In three cases the quality did not change much and in three cases correction was little too strong and the colours became too flavored. In most cases there were defects in colours in some areas in the original video frame and the overall quality of the original video was also very poor. The correction tends to strengthen these discoloured details and compared to original this is sensed as a deterioration of quality.

The primary objective of this work was to test and develop methods to use still image algorithms in moving picture environment. Main problem is temporal continuity and this was also emphasized in the performance test. Visual quality in the transitions was examined delicately. The visual test proved that all transitions were corrected without visual defects. If video is watched in slow motion (frame by frame) then some transition changes may be seen, but in normal real time playback these defects are unnoticeable.

CONCLUSION

Automatic colour correction of moving pictures is not as straightforward as correction of still images. Two main problems arise: temporal continuity and adaptation of correction in gradual and abrupt scene changes. The objective of this work was to study how still image correction algorithms may be used in the digital video. Adaptive recursive filtering of correction parameters, according to scene change and transition detection, is proved to be workable solution for applying automatic colour correction algorithms onto moving pictures. Any kind of scene change and transition detection algorithm may be used. In this work, the detection algorithm was based on the same model as the actual correction algorithm. The approach is intuitively reasonable. However, better results may be achieved with other detection algorithms or using cluster of different algorithms and using combination of their outputs.

REFERENCES