Fast Luminance and Chrominance Correction based on Motion Compensated Linear Regression for Multi-view Video Coding

Wei-Yin Chen, Li-Fu Ding, and Liang-Gee Chen

DSP/IC Design Lab,
Department of Electrical Engineering and Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

Luminance and chrominance correction (LCC) is important in multi-view video coding (MVC) because it provides better rate-distortion performance when encoding video sequences captured by ill-calibrated multi-view cameras. This paper presents a robust and fast LCC algorithm based on motion compensated linear regression which reuses the motion information from the encoder. We adopt the linear weighted prediction model in H.264/AVC as our LCC model. In our experiments, the proposed LCC algorithm outperforms basic histogram matching method up to 0.4dB with only few computational overhead and zero external memory bandwidth. So, the dataflow of this method is suitable for low bandwidth/low power VLSI design for future multi-view applications.

Keywords: illumination compensation, multi-view video coding, MVC

1. Introduction

Nowadays, multi-view video coding and processing have become a trend in the related visual processing field, and the 3D or free-view point TV applications are emerging as well [1]. In this circumstance, the issue of multi-view video encoding attracts much effort to improve its power-rate-distortion efficiency and functionality. The main difference between multi-view video coding systems and normal video coding systems is the inter-view prediction, so the effectiveness of inter-view redundancy removal is the most critical part affecting the rate-distortion (R-D) performance. However, due to the badly calibrated camera array, not all inter-view redundancies can be easily removed by the current disparity estimation (DE)/disparity compensation (DC) techniques. Therefore, to utilize maximal inter-view correlation, the image sources from different cameras should be adjusted to the same tone and illumination before the estimation and compensation steps.

For all the various coding tools developed in this decade, weighted prediction (WP) [2] is suitable in this scheme. The original usage of WP is to compensate the luminance and chrominance difference between current frame and the temporal reference frames, such as flashes, fade-in/fade-out, and changed lighting conditions. It can be easily adapted for the inter-view reference frames because the underlying representation method of the frame reference structure is similar. Furthermore, the extent of the luminance/chrominance distortion of inter-view reference frames is averagely higher than that of temporal references, so this coding tool can have better use in this application.

According to many research results based on this coding tool, the global (frame-wise) linear model of WP defined in AVC is sufficient in most cases [3, 4]. In this model, there are two parameters for each reference frame that describe the slope (ratio) and intersection (offset) of the linear transform formula. Defining a model is one thing, but finding the best parameters in the model is another. There exists a trade-off between the simplicity of the searching algorithm and the accuracy of the model. The simplest way to get the linear model parameters is implemented in the reference software, JSVM [5]. In this algorithm, the luminance/chrominance DC values (frame average) of the current frame and the reference frames are calculated, and the parameters are derived from the ratio or difference of these DC values. Because this naïve method cannot get an accurate model and has few coding gain, some other methods are

Visual Communications and Image Processing 2007, edited by Chang Wen Chen, Dan Schonfeld, Jiebo Luo,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6508, 650823, © 2007 SPIE-IS&T · 0277-786X/07/$18

proposed to achieve higher accuracy and robustness, like linear regression of co-located pixels [6], linear regression of motion compensated pixels [4], histogram matching [7, 8], and translational global disparity compensated histogram matching [3]. The methods listed above more or less get better model parameters, but they either neglect those non-global motions ([3, 6, 7, 8]) or have high computational complexity ([4]). Since the motion estimation (ME), or equivalently the disparity estimation (DE), is an essential step in all the video encoders, we can leverage this information without much additional effort to retrieve the relationship of paired pixels between current and reference frames, so the matching assumption of linear regression method can be fulfilled.

The rest of this paper is organized as follows. The conventional and proposed algorithms are described in section 2, and the experimental results will be shown in section 3. Finally, section 4 contains the conclusion.

2. The Proposed Algorithm

2.1. Conventional encoding process with weighted prediction

To illustrate the proposed algorithm clearly, the conventional coding process with WP is explained first below. As the video coding standard only defines the behavior of the decoder, the description of decoder below is mandatory, but the behavior of the encoder introduced below is only one of many possible implementation methods.

According to the video coding standard, the parameters of the linear correction model for WP consist of an integer offset and a fixed point fractional ratio for each reference frame, and the bit-width of the parameters can be adjusted according to the needed precision and the range of value. The WP parameters are embedded in the header of each frame, and each reference frame has its own WP parameters. When decoding a macro-block (MB), the luminance and chrominance of the referenced MB should be adjusted according to the WP parameters of the frame in which the MB resides. Bidirectional predicted MBs (B-MBs) are treated slightly differently. The WP parameters of the two reference frames are considered altogether to make a blended model for the averaged prediction. In this paper, we neglect B-MBs.

For the encoder, the parameters of the linear correction model for WP are determined before encoding the frame. After that, ME/DE and MC (motion compensation)/DC are conducted. In the ME/DE stage, the current block is reversely transformed according to the linear correction model before doing the block matching algorithm (BMA) rather than transforming the pixels in the searching area in order to save some computation while keeping reasonable accuracy. In the MC/DC stage, the best-matching block is transformed before compensation, like what should be done in the decoder side.

2.2. Problem definition

The disparity compensation with WP can be expressed as (1), where I^{comp} is the compensated frame, which should be close to the current frame, I^{cur}. The frame-wise parameters in LCC model are m and b in (1). I^{ref} is the reference frame, and (x'-x, y'-y) is the disparity vector (DV) of pixel (x, y) in I^{cur}.

$$I^{comp}(x, y) = m\, I^{ref}(x', y') + b \tag{1}$$

Ideally, the LCC parameters should be determined in order to minimize the difference between I^{comp} and I^{cur}. Sum of absolute difference (SAD) is one possible criterion, and the expression is written as

$$(m, b) = \arg\min_{m,d} \sum_{(x,y)} abs(I^{comp}(x,y) - I^{cur}(x,y)). \tag{2}$$

The exhaustive search is not feasible because the solution domain includes all the possible motion vectors (MVs) in each MB and the two LCC parameters, so the complexity is the complexity of ME times the possible range of LCC

parameters. One method to decrease the time complexity is decomposing the process into two phases, ME and color correlation.

2.3. Linear regression with reused motion information

Since the process is decomposed into two phases, we can reuse the motion information from the ME stage in the normal coding loop. In this way, the only additional computation is the color correlation. We use linear regression over all the motion estimated pixel pairs to calculate the color correlation. In order to find the LCC model between I^{cur} and I^{ref}, m and b can be calculated after linear regression is done on all the pixel pairs ($I^{cur}(x, y)$, $I^{ref}(x'', y'')$). It should be noted that the DV used in encoding (x'-x, y'-y), assuming it is the real motion, could be different from the disparity vector used to estimate the LCC model (x''-x, y''-y). The closer the two different DVs are, the more accurate LCC model could be got. If we refine the model iteratively, the two kinds of DVs can gradually become closer.

The original usage of the mono-view WP is to compensate the changing illumination. The WP parameters in the mono-view coding changes rapidly and the parameters in different frames are generally not correlated very much, so they should be separately estimated in each frame. However, as the mismatch between a camera pair is relatively stable over time, the LCC model only changes gradually between the same view pairs, so it is wasteful to calculate the model right from the beginning with huge computation resource like the original algorithm used for mono-view WP. Alternatively, the computation in the proposed algorithm is highly reduced and amortized to multiple iterations spread in consecutive frames, so only a low-cost refinement is performed in an iteration. Furthermore, it keeps the ME/DE module intact and reuses the motion/disparity information. Although the R-D efficiency could be worse than doing dedicated ME/DE as [4], the computational cost is largely reduced and the correction model still converges to the optimal value after reasonable iterations.

The more biased the tone between different views, the less accurate the MVs can be estimated in the first iteration, thus the color correlation calculated from the pixel pairs could be affected either. As a result, more iterations are needed if the color distortion is severe.

The coding flow is described in the pseudo code in Fig. 1. In the coding flow, the LCC model is fixed in the ME/DE and MC/DC stage. After the motion information is generated, the LCC model is updated for the next iteration.

```
(m, b) ← model from the previous iteration
for each current block {
    reversely transform the current block B^cur by (m, b)
    perform DE and get matched block B^ref
    accumulate linear regression registers with (B^cur, B^ref) pair
    compensate B^comp with forwardly transformed block B^ref
}
(m, b) ← new linear model from linear regression
```

Fig. 1. Pseudo-code of the proposed algorithm

Let's express the i^{th} frame in j^{th} view as V_jF_i, and the LCC model estimated at time i to predict view j from view k is $LCC(V_jF_i, V_kF_i)$. The proposed algorithm uses $LCC(V_jF_i, V_kF_i)$ as the initial value in the refinement process for

$LCC(V_jF_{i+1}, V_kF_{i+1})$, since the LCC model between the same view pair in consecutive time slots should be similar. When encoding frame V_jF_{i+1}, each blocks are reversely transformed by the initial model $LCC(V_jF_i, V_kF_i)$. In the estimation stage, the reversely transformed current block are compared with many candidate blocks in the reference frame V_kF_{i+1}, and the best matching block pair between V_jF_{i+1} and V_kF_{i+1} are found. The best matching block in V_kF_{i+1} is forwardly transformed to be the compensated block. The model $LCC(V_jF_i, V_kF_i)$ is adjusted by the statistical data gathered from the motion/disparity information, and then saved as $LCC(V_jF_{i+1}, V_kF_{i+1})$ to be used at the next time prediction view j from view k.

There is a single-frame latency between the estimation and the application of LCC models, so the performance can be degraded when the LCC model changes rapidly.

To sum up, the proposed algorithm is an iterative refinement process of the model parameters. When encoding a frame with the previously calculated LCC model, the pixel pairs in matched blocks are fed in the linear regression module. When all the blocks are finished, the LCC model can be refined according to the result of the linear regression, and the new model is used as the initial model when encoding frames from the same view next time.

2.4. Computational cost and bandwidth overhead

Compared with the conventional encoder, the proposed algorithm only additionally requires the linear regression computation, which consists of three multiplications and five additions per pixel per color channel per reference frame, and five multiplications, three subtractions, and two divisions per color channel per reference frame. This computational cost is negligible for a video encoder compared with ME/DE. The linear regression is done at the same time when the best-matched block is loaded in the compensation stage, so it causes no bandwidth overhead to the external memory, or equivalently higher cache hit-rate in software model. In the perspective of VLSI design, low external bandwidth leads to low power design, and the hardware area cost is only several adders and multipliers, and around 150 bits of registers.

When encoding a video sequence in D1 size, 30 fps, the three kinds of algorithms can be compared as follows:

Algorithm	Histogram-based	Proposed algorithm	Linear regression with dedicated ME
Computational cost (million arithmetic operations per second)	> 15.5	124.4	2073.6 (assuming 100 ME candidates)
R-D performance	lowest	higher	highest

3. Experiment

3.1. Environment setup

JSVM version 3.5 is modified to support our LCC model refinement algorithm. Several multi-view video sequences are encoded to compare the R-D efficiency of multiple LCC model searching algorithms, and some R-D curves are shown in section 3.3. The prediction structure shown in Fig. 2 is used in order to focus on the efficiency of inter-view prediction.

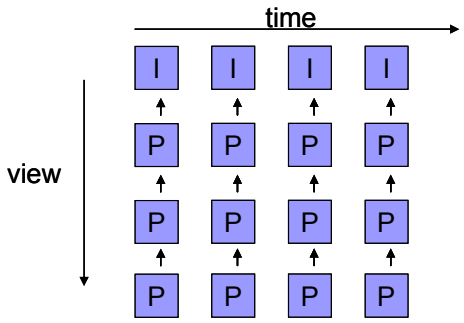

Fig. 2. Multi-view prediction structure in the encoder

3.2. Initial correction models

In this experiment, two initial LCC model values are used. One of them is the default algorithm of the WP implemented in JSVM 3.5, which uses the ratio of the DC values of the current frame and the reference frame as the slope of the linear model and set the intersection to zero. Another is histogram matching, which minimizes the area of absolute difference of the accumulated histograms of current frame and reference frame by tuning the LCC model.

3.3. Experimental result

The experiment result is shown below. The label "no WP" means the sequence is encoded without WP, "DC" means the initial value of the correction model is calculated by the ratio of DC values, and "HM" means the initial value is from histogram matching. The label ended with "n iter" means n refinement iterations of proposed algorithm are done after the initial model.

Compared with the HM algorithm, the proposed algorithm has coding gain from 0.1 to 0.4 dB in different bitrate ranges in Fig. 3. It is shown in Fig. 4 and Fig. 5 that the PSNR increases with number of iterations. If the initial value is as good as HM model, then the correction model would converge after only three iterations. Even if the initial value is worse than "no WP" like "DC", the proposed algorithm can still improve it to nearly the same PSNR as that using "WP" as initial value, and the difference is smaller than 0.1dB. So it can be concluded that the initial model is not absolutely needed and m=1, b=0 can be used. A worse initial model only increases the number of iterations required to converge to steady state, but it doesn't severely affect the quality of that steady state.

The inter-view mismatches in sequence Exit are lower, so WP is not very helpful. However, from Fig. 7, the coding gain of HM is negative, while that of proposed algorithm is still positive, and this shows the robustness of our method.

Form Fig. 8 to Fig. 11, we can see that the color distortion in sequence Ballroom and Breakdancers is even lower. No matter how many iterations are applied after the "DC" initial condition, the R-D curve is still lower than that of "HM". This seems contradictory to the previous claim, but the difference is less than 0.05dB.

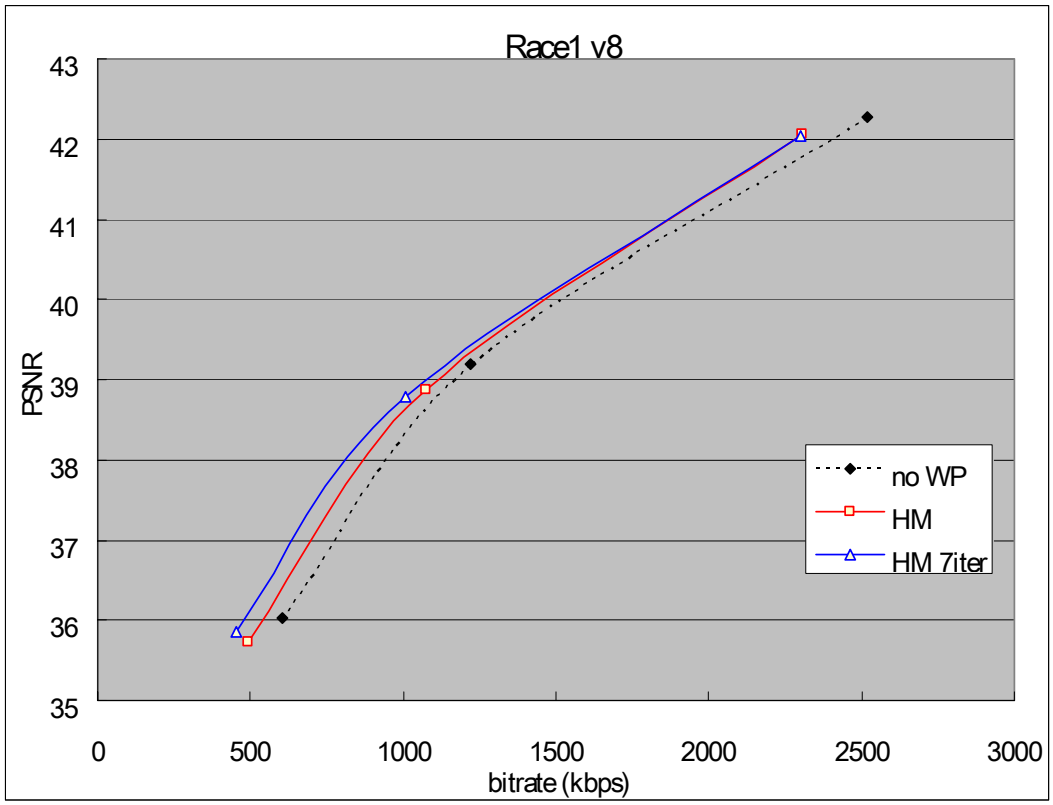

Fig. 3. RD curve of Race1

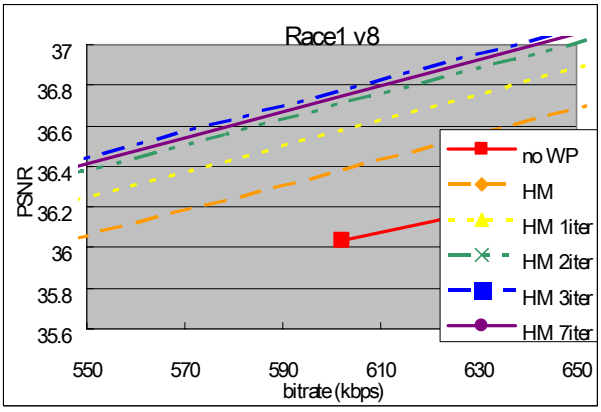

Fig. 4. Magnified RD curve of Race1, HM as initial value

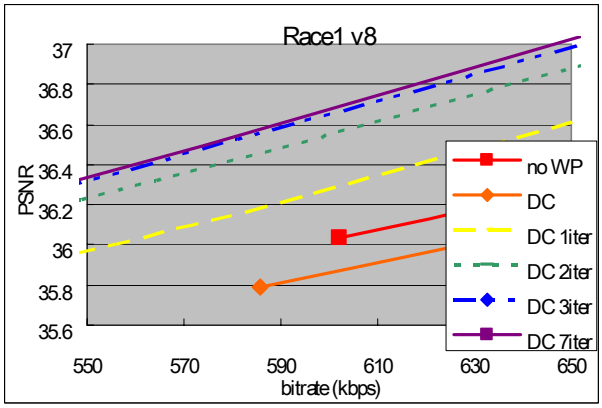

Fig. 5. Magnified RD curve of Race1, DC as initial value

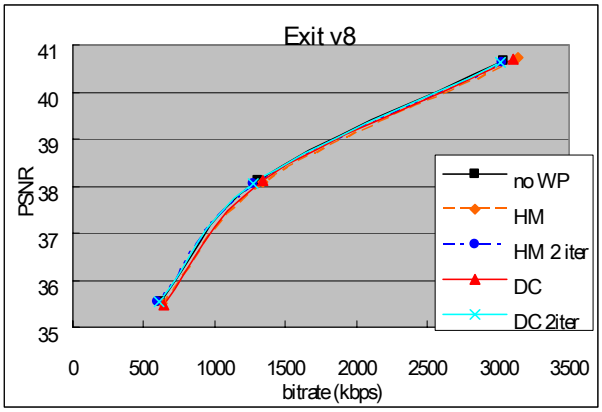

Fig. 6. RD curve of Exit

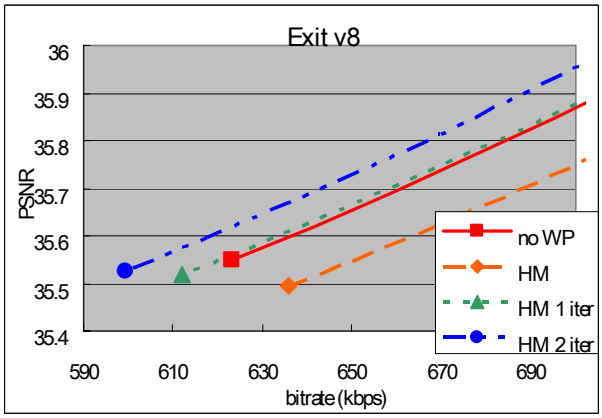

Fig. 7. Magnified RD curve of Exit, HM as initial value

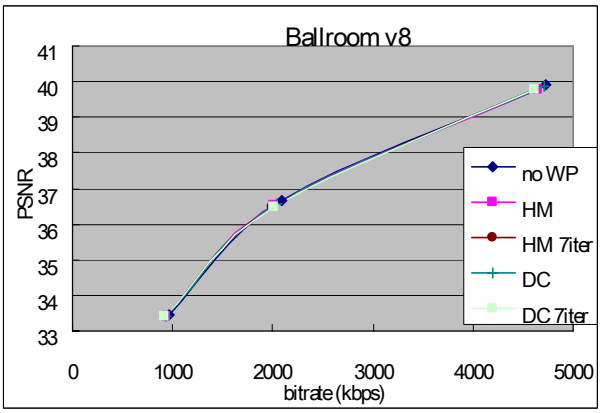

Fig. 8. RD curve of Ballroom

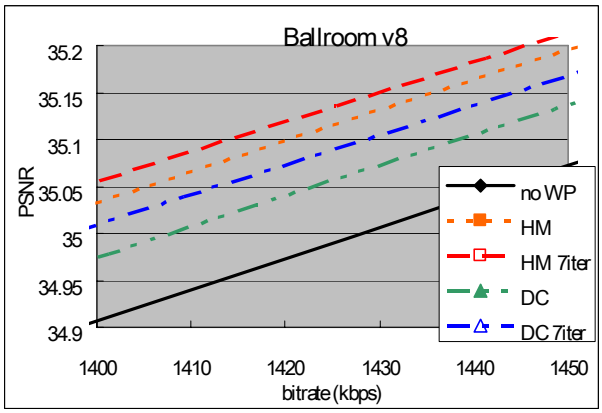

Fig. 9.Magnified RD curve of Ballroom

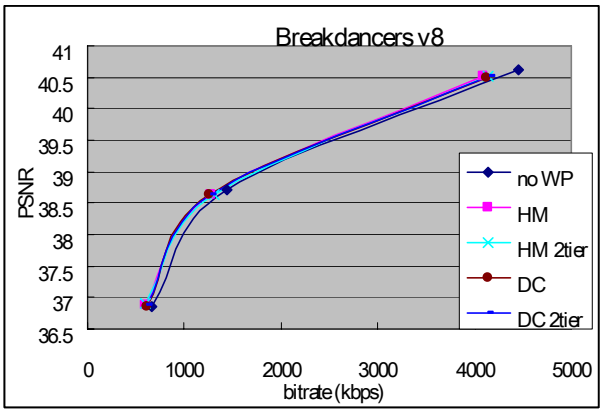

Fig. 10. RD curve of Breakdancers

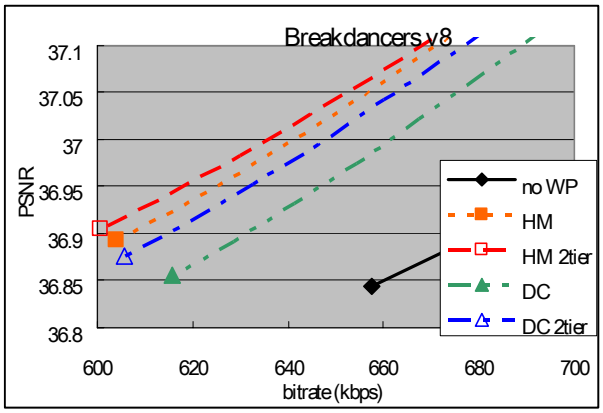

Fig. 11.Magnified RD curve of Breakdancers

4. Conclusion

Luminance and chrominance correction for multi-view video coding can boost the rate-distortion curve considerably when the images from the camera array are not perfectly adjusted. The distortion of luminance and chrominance is modeled as a linear transform, and the global WP model in H.264/AVC is used in our coding system. The proposed algorithm amortizes the computation of multiple ME/DE-color correlation refinement iterations to multiple frames in the same view, and reuses the motion information from the encoder to calculate accurate LCC models with negligible computation overhead. The dataflow is also suitable for low power VLSI implementations because it requires zero additional external memory bandwidth. The coding gain of the proposed LCC model refinement algorithm compared with HM model is up to 0.4 dB. Due to the higher robustness of the proposed method, the coding gain for those sequences with little mismatch is generally positive, while that of HM is sometimes negative. Besides coding efficiency, the LCC model embedded in the encoded bitstream can be further utilized by the decoder to do the post-processing if homogeneous tone and illumination is preferred for the quality of subjective view.

5. References

1. T. Fujii, M. Tanimoto, “Free-Viewpoint TV System Based on Ray-Space Representation”, SPIE ITCom Vol. 4864-22, pp.175-189 (2002)
2. Boyce, J.M., “Weighted prediction in the H.264/MPEG AVC video coding standard,” Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on Volume 3, 23-26 May 2004 Page(s):III - 789-92 Vol.3
3. Yushan Chen, Jianle Chen, Canhui Cai, “Ni Luminance and Chrominance Correction for Multi-View Video Using Simplified Color Error Model,” PCS 2006
4. Kamikura, K.; Watanabe, H.; Jozawa, H.; Kotera, H.; Ichinose, S., “Global Brightness-Variation Compensation for Video Coding,” Circuits and Systems for Video Technology, IEEE Transactions on Volume 8, Issue 8, Dec. 1998 Page(s):988 – 1000
5. Weighted prediction for SVC, “JSVM 3.5 Software,” JVT-P064
6. Sang Hyun Kim; Rae-Hong Park, “Fast Local Motion-Compensation Algorithm for Video Sequences with Brightness Variations.” Circuits and Systems for Video Technology, IEEE Transactions on Volume 13, Issue 4, April 2003 Page(s):289 - 299
7. ISO/IEC JTC1/SC29/WG11, “Luminance and Chrominance Compensation for Multi-View Sequences Using Histogram Matching,” Nice, France, 2005
8. U. Fecker, M. Barkowsky, A. Kaup, “Improving the Prediction Efficiency for Multi-View Video Coding Using Histogram Matching,” Proc. Picture Coding Symposium (PCS 2006), Beijing, China, Apr. 2006