Multiview video quality enhancement

Ljubomir Jovanov
Hiêp Luong
Wilfried Philips
Multiview video quality enhancement

Ljubomir Jovanov,* Hiêp Luong, and Wilfried Philips
iMinds-IPI-Ghent University, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium

Abstract. Realistic visualization is crucial for a more intuitive representation of complex data, medical imaging, simulation, and entertainment systems. In this respect, multiview autostereoscopic displays are a great step toward achieving the complete immersive user experience, although providing high-quality content for these types of displays is still a great challenge. Due to the different characteristics/settings of the cameras in the multiview setup and varying photometric characteristics of the objects in the scene, the same object may have a different appearance in the sequences acquired by the different cameras. Images representing views recorded using different cameras, in practice, have different local noise, color, and sharpness characteristics. View synthesis algorithms introduce artifacts due to errors in disparity estimation/bad occlusion handling or due to an erroneous warping function estimation. If the input multiview images are not of sufficient quality and have mismatching color and sharpness characteristics, these artifacts may become even more disturbing. Accordingly, the main goal of our method is to simultaneously perform multiview image sequence denoising, color correction, and the improvement of sharpness in slightly defocused regions. Results show that the proposed method significantly reduces the amount of the artifacts in multiview video sequences, resulting in a better visual experience. © 2016 SPIE and IS&T [DOI: 10.1117/1.JEI.25.1.013031]

Keywords: multiview; denoising; restoration; sharpness improvement; color matching.

1 Introduction

We have witnessed rapid developments in visual communication devices in the last decades, and each of the steps within this evolution has brought improved visual user experience, such as color, increased resolution, progressive image analysis, and stereovision. Moreover, multimedia devices, like notebook PCs, tablets, and smartphones, have proven capable of synthesizing/streaming and displaying high-quality visual content. Thereon, the next big steps toward achieving entirely realistic visualization are multiview autostereoscopic displays, which provide multiple distinct views at the scene, for multiple viewers or one moving viewer.

To achieve a realistic and visually pleasing viewing experience, the number of views should be as large as possible. Moreover, view transitions should be seamless, the geometry of the scene should remain intact and compression and other artifacts should not be visible. While such display systems already exist and provide high visual quality if computer-generated content or light-field approach for static scenes capture are used, video content captured using multicamera systems is still not at the quality level desired.

While multiview autostereoscopic displays are capable of displaying hundreds of views, the multicamera capture system cannot provide a sufficient number of video streams. Moreover, in order to achieve a seamless transition, these views should be as close as possible; due to the finite dimensions of cameras in the system, captured video streams are at a much larger distance than the one required for a seamless view transition. In order to feed the display with the required number of views, nonexistent views have to be generated using the existing ones as an input, as shown in Fig. 1. While three-dimensional (3-D) capture sensors like Kinect are already widely used for 3-D capture, multiview video streams are typically captured using equidistant linear arrays of conventional color cameras. Alternatively, cameras may be positioned on an arc. It is desirable here that the cameras are placed on a smallest possible distance, which is not always possible, especially in the case of high-quality broadcast cameras. Missing views are typically synthesized using depth image-based rendering techniques or by relying on view synthesis techniques based on warping.

View synthesis techniques are very sensitive regarding the quality of input video streams, and both groups of techniques require estimation of disparity fields between cameras. While numerous techniques can already achieve excellent performance levels in controlled conditions and on standard datasets, disparity estimation algorithms, in reality, often fail to achieve satisfactory results.

Noise in video sequences influences the visual quality in multiple ways; first, it directly deteriorates the quality of video streams captured using the existing cameras in the setup; second, the performance of the disparity estimation algorithms is degraded in the presence of signal dependent noise. The presence of noise increases uncertainty during the search process and masks salient features used for matching. This is additionally complicated by the fact that the noise variance is signal dependent. Finally, in the view interpolation step, views synthesized from noisy input video streams using approaches based on warping may suffer from correlated noise, since the warping approaches are often based on a weighted average using the reliability of the disparity estimation.

The next factor that deteriorates the visual quality of the interpolated views is color variation between cameras. Due to the different parameters of cameras in the setup, like gain,
offset, and gamut, the same objects and regions in video streams from different cameras may have different color appearances. Even in the case when cameras are calibrated using color reference, the color of the object might still change from camera to camera.

Color mismatch may also have an adverse affect on disparity estimation algorithms, since the appearance of a certain object in the scene will not be exactly the same in video streams from different cameras. Unmatched characteristics of the cameras also have an adverse effect on the view interpolation, since colors in interpolated views will change in an unnatural way that is not expected by the user. Color changes as user moves from one camera to another should only appear due to the different light reflections and the directional characteristics of different materials within the scene.

The visual quality of multiview video streams can also suffer from differently focused cameras in the array. Due to the different distances of camera objectives from objects in the scene and different focus settings, objects that are in focus in one view may appear out of focus in others. Similarly, as in the case of the previous types of distortions, these differences in sharpness influence both the performance of disparity estimation algorithms and the visual quality of the interpolated views.

For our research, a very important practical case where this type of distortion is present is microstereopsis cameras, due to the optical characteristics of the beam splitter or prism. Since most of the disparity matching algorithms rely on high-frequency image content, such as sharp edges, their blurring significantly reduces the accuracy of the stereo matching. Wrongly estimated disparity maps in combination with differently focused views result in an even lower quality of the interpolated views.

In this paper, we present a method for simultaneous noise reduction, sharpness correction, and color matching, with the paper organized as follows: in Sec. 2 we describe the architecture and the components of the method, in Sec. 3 we describe the experimental results. Finally, in Sec. 4 we conclude the paper.

2 Method Description

In this section, we describe, in detail, the proposed method and its components. To reduce the influence of the artifacts mentioned in Sec. 1 on the performance of the multiview system, we propose a new method for the simultaneous correction of these artifacts, shown in Fig. 2. Outputs of each of the cameras are first denoised using the motion compensated wavelet-based method, in order to facilitate sharpness and color correction. Thanks to the denoising step, the performance of the next processing steps is significantly improved.

The structure of the proposed multiview enhancement algorithm is depicted in Fig. 2. Each of the algorithms comprising the proposed method shares the following processing blocks: wavelet transform, motion/disparity estimation, and segmentation algorithm. Wavelet coefficients are first used by the video denoising algorithm, which then forwards denoised wavelet coefficients to the color correction block and further to the sharpness correction block. Motion vectors estimated by motion estimation block are utilized in the video denoising step, while disparity estimates between different cameras are used by color and sharpness correction for warping operations between different views. The results of the segmentation are utilized in all processing steps. More detailed description of the proposed method is given in Secs. 2.1, 2.2, and 2.3.

2.1 Multiview Video Denoising Method

As mentioned earlier, multiview video is captured using conventional color or microstereopsis cameras. Thus, the multiview video shares the problem of noise originating from different sources with two-dimensional (2-D) video, e.g., shot noise, dark current noise, fixed pattern noise, amplifier noise, and quantization noise. Due to the increased resolutions and frame rates, the quantity of light reaching each pixel is reduced, which makes the amount of noise higher and more pronounced.

While image and video denoising have received a lot of attention in the past few decades, the first methods on
multiview image denoising have been published only recently. One of the first such approaches was presented in Ref. 10, where Zhang et al. proposed an approach to capture multiview images using an array of pinhole cameras, with a small aperture and short exposure, to achieve minimal optical defocus and motion blur. As a result of such camera settings, the incoming light has a low intensity, resulting in very noisy images. The main idea in this paper is to use the depth map of the scene to search for similar image patches in other camera views. Moreover, Zhang et al. incorporated realistic signal-dependent noise models and adapted to the spatially varying noise parameters in different regions of the image.

Dai et al. 11 presented a different approach that does not rely on the existing depth map. In the first processing step, images from the multiview set are denoised using BM3D algorithm. 12 Next, feature points that are projected to other views are identified using the PMVS approach from Ref. 13. Feature points determined this way are consequently used as centers of patches used for collaborative denoising. These patches are then organized in a graph and dissimilarity measures, based on the geodesic distance calculated between each pair of patches in the graph are calculated. The patches with the smallest dissimilarity measure are then selected for Wiener filtering.

One of the first multiview video denoising methods was presented in Ref. 14. First, for each denoised patch, the algorithm searches for eight most similar patches in previous frames, as well as in other views. For this purpose, a coarse-to-fine procedure is used to improve accuracy and robustness. After these patches are collected, denoising is performed using thresholding in a 3-D discrete cosine transform (DCT) domain.

Finally, Luo et al. 15 proposed an adaptive nonlinear means (NLM) based method for the denoising of video sequences captured by a multiview system. To achieve major improvements over the baseline NLM algorithm, Luo et al. proposed certain adaptations, such as a robust joint-view distance metric, in order to measure the similarity of patches and an adaptive procedure derived from the statistical properties of patches in order to determine the optimal number of patches used for denoising. Due to these adaptations, noise is removed uniformly from the whole image without excessive blurring.

2.1.1 Proposed multiview video noise reduction method

In this section, we present the new multiview video denoising method. In this regard, video sequences contain a large amount of redundancy, since every subsequent frame can be approximated as a geometric transformation of previous frames in the sequence. Each pixel in a certain frame can be observed as one noisy realization of a measurement; thus, if we can find all the noisy versions of the current pixel, the estimate of the “noise-free” value of the pixel can be found efficiently. Displaced locations of each pixel here are usually found using motion estimation or through search for the most similar patches around the current pixel in a few neighboring frames. Such an approach was applied in some of the recent and powerful techniques, such as 12,16 in pixel and DCT domain.

While block-based approaches can remove noise efficiently, due to the use of blocks and imperfections of search procedures, the end result might suffer from some blocking and ringing artifacts. Moreover, imprecision of block-based greedy motion estimation methods make such methods highly inefficient in the case of correlated noise. In order to avoid these artifacts and perform denoising efficiently in the case of more realistic correlated noise possible, we propose the use of more robust and precise optical flow from Refs. 17 and 18.

This method combines the principles of local optical flow methods such as the Lucas–Kanade technique, with the principles of global methods such as the Horn/Schunck approach. Local optical flow methods provide more robust motion estimation, while global methods provide a dense motion field. Moreover, this approach combines a brightness

Fig. 2 The proposed multiview enhancement algorithm.
and gradient constancy assumption and a discontinuity-preserving spatiotemporal smoothness constraint. In this way, the robustness of the proposed denoising method to noise is ensured, while achieving highly precise motion estimation, which facilitates highly efficient noise reduction.

In this paper, we propose to use a dense motion field estimation to find the most similar pixels to the one currently denoised in neighboring frames, where the proposed algorithm recursively operates on a frame buffer of seven frames. In the initialization phase, we first estimate pairwise motion fields, from the first until the fourth frame, and, similarly, in the opposite direction, from the seventh until the fourth frame, for each pixel, as shown in Fig. 3.

By cumulatively using motion fields estimated in this way, we perform warping of all frames in the buffer to the middle, fourth frame in the buffer. For denoising of all consecutive frames, it is only necessary to estimate new motion fields between the third and fourth, and seventh and sixth frame in the buffer. When block-based motion estimation methods are used, the quality of registration degrades with the temporal distance of the frame. These degradations especially occur around the borders of the objects, where large occlusions appear. Bidirectional warping can correct these occlusions using frames from the opposite direction, with a limited success. The proposed multiview denoising method is performed along temporal motion trajectories in order to enforce temporal coherence and avoid flickering of the denoised multiview video sequence. Such flickering may occur if the denoising is performed per frame, for all cameras in the system.

Despite highly accurate motion estimation, errors can still occur in the presence of strong noise. These errors in the estimated motion field often manifest as artifacts in the processed frame, especially in the case of pixel-based denoising methods. To reduce the visibility of these artifacts, the proposed method operates in the multiresolution domain, where it is possible to separate the base layer, which corresponds to low-frequency content, from remaining layers containing high-frequency details. In this paper, in particular, we propose the use of second-generation wavelets, so-called “edge-avoiding wavelets,” in all processing steps.

In order to avoid artifacts around edges, which often occur when first-generation wavelets are used, the support of edge-avoiding wavelets is constructed taking into account edges in the image and avoiding the use of pixels from either side of the edges. Thanks to this strategy, edge-avoiding wavelets show better decorrelation of the data compared to classical wavelets. Since the interscale correlation is reduced, halo artifacts can be avoided without taking any special steps for their suppression. The remaining processing steps are also performed using edge-avoiding wavelets in order to fully exploit their benefits.

If we denote an unknown noise-free $k$th frame of $m$th view as $x_m(k)$, its noisy version captured by a camera as $y_m(k)$ and signal-dependent noise as $\sigma(x)n_m(k)$, it can be written as

$$y_m(k) = x_m(k) + \sigma(x)n_m(k).$$

Noise in video sequences mainly originates from the amplifier and shot noise, and also largely depends on the transfer characteristics of processing blocks in the camera. Our goal is to estimate the noise standard deviation without knowing the parameters of the camera. To achieve this often means recording a number of calibration video sequences of a static scene or a slowly varying scene. In the case of a static scene, we record a sufficient number of frames (>200) and calculate per-pixel standard deviation. This per-pixel standard deviation is equivalent to noise standard deviation, since the scene is static and the only variation is that which originates from the noise.

However, it is highly desirable to have the ability to adapt to the changes of camera settings and, therefore, the noise characteristics over time, without having to stop the recording. In order to achieve this, we perform segmentation of the video scene onto segments with approximately constant color, as described in Ref. 20. The main advantage of such segments is that they perfectly follow the shape of the objects in the scene, as shown in Fig. 3, which enables us to track the constant color segments more efficiently over time, following each of the segments along the estimated motion trajectories in the buffer. For each of the segments, we calculate the

Fig. 3 Segment-based temporal filtering.
average and the standard deviation per pixel along the motion trajectory. In order to estimate the characteristics of signal-dependent noise, we first remove higher residual values originating from motion compensation errors.

Motion compensation errors from locations with less accurate motion estimation or occluded areas have significantly higher values than those originating only noise. A scatter plot of the dependence between noise standard deviation and pixel value is shown in Fig. 4. The upper bound on noise standard deviation is shown using the black line in the same plot. The shape of the scatter plot and the upper bound agree with the visual observations that the noise variance is higher in regions with lower illumination. In Fig. 5, we show upper bounds on the noise standard deviation plots for all three color channels. As we can see from Fig. 5, noise standard deviation is highest in the blue channel and much lower in green and red channels. To facilitate the noise level estimation for each pixel of each color channel, we approximate the upper noise standard deviation bound using second-order polynomials. An example of a fitted polynomial for noise values larger than a saddle point in the graph is shown in Fig. 6. To calculate the noise standard deviation for each pixel, we evaluate the estimated polynomial model using a value of low-frequency band of the wavelet decomposition. The main reason for this is that low passband contains less noise and less temporal variation than the noisy video frames. Once noise parameters are determined, we perform denoising in the wavelet domain, first performing wavelet transform on each of the registered frames. We observe $3 \times 3$ patches around each of the wavelet coefficients, in the currently denoised frame, and patches in other frames in the buffer, $p_l$ and $p_m$, and calculate the dissimilarity measure between them as

$$d(p_l, p_m) = \sum_{i=1}^{N} ||p_{li} - p_{mi}||^2,$$

where $p_{li}$ denotes $i$‘th coefficient of the patch around the currently denoised coefficient at the location $l$ and $p_{mi}$ is the $i$‘th coefficient of the patch around the warped coefficient location $m$ from the other frames in the buffer and $N$ is the number of coefficients in the patch. As noted earlier, the use of coefficients from certain areas in warped frames would create artifacts in denoised frames, due to the occlusions. In order to avoid them, we observe dissimilarity measures from Eq. (2) and compare them to a predefined threshold THR. If the value of Eq. (2) is higher than the threshold, we do not use the pixel located at $m_i$ for denoising; such cases can certainly cause lower denoising performance in the occluded regions. To improve the denoising performance, especially in occluded regions, we allow additional coefficients to be included in the set of the candidates for denoising, by also relying on the segmentation of the input frames, where segments tend to follow contours of the objects containing pixels with approximately constant color or texture. The first way of extending the candidate set is by searching

![Fig. 4 Joint plot of pixel values and noise standard deviation for the red color channel.](image-url)

![Fig. 5 Upper bounds of noise standard deviation for all three color channels.](image-url)

![Fig. 6 Fitted noise standard deviation for the red color channel.](image-url)
the most similar segments in the current frame/camera view, which belong to the same object, follow the same edge, have similar texture, and so on, in the local neighborhood of the currently denoised wavelet coefficient. For example, in the case of occluded regions we would like to use only the coefficients from the background region of the current frame. For this reason, using the pixels inside each of these segments for denoising will significantly contribute toward noise removal without introducing any additional blur to the edges. We further improve denoising in the occluded regions by identifying frames in the frame buffer which contain them. For example, if any region covered in the previous frame subsequently becomes visible in the current and future frames, we cannot use affected regions from the previous frame in order to denoise the current frame. On the other hand, the affected region becomes visible in the future frames, so it is possible to use them for denoising. We fully exploit this bidirectional motion estimation in the proposed method. The second way of extending the candidate is using segments from neighboring views, in which case the estimated noise-free wavelet coefficient is obtained using the following expression:

\[ s_{l,o,c} = \frac{1}{Z} \sum_{t=1}^{T} w_{l,t,o,c}s_{l,t,o,c}, \tag{3} \]

where \( s_{l,o,c} \) is the value of the wavelet coefficient at the location \( l \) in the current central frame, \( w_{l,t,o,c} \) is the significance/weight of the warped coefficient at the location \( l \) from the frame \( t \). \( s_{l,t,o,c} \) is the value of the coefficient from the \( l \)th frame in the buffer, \( o \) and \( c \) are the orientation and the scale of the wavelet band being denoised, and \( Z \) is the normalization constant. Local weights used for denoising are defined as

\[ w_{lm} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(p_{lm}-m_{lm})^2}{2\sigma^2}}, \tag{4} \]

where \( \sigma \) is estimated local standard deviation of noise. We perform the filtering from the Eq. (3) on all wavelet bands and orientations. In order to avoid temporal flickering of the low spatial frequency image content, we also apply the filtering step from Eq. (3) to the scaling coefficients. In this case, we adapt the estimated noise variance used for the calculation of weights in Eq. (4), since the calculation of the low-frequency “base layer” already performs adaptive low pass filtering. Noise standard deviations are first estimated for each spatial location and color in pixel domain. However, after calculating wavelet coefficients, noise standard deviations in different wavelet bands differ significantly. For example, to extract low-frequency “base layer,” adaptive low-frequency filtering is performed on image pixels. Due to this low-pass filtering, the amount of the noise in the base layer is lower than the initial estimate. To obtain the noise standard deviation (std) in the base layer, we utilize initial noise std estimate and the filter coefficients used to calculate low-frequency coefficients of the base layer.

The set of pixel neighborhoods used in Eq. (4) is extended until noise is reduced sufficiently, without introducing excessive blur, similarly as in Ref. 21. Finally, we calculate the inverse wavelet transform and obtain the denoised frame, where the amount of noise in the individual video streams is significantly reduced after the denoising step, as we will show in Sec. 3.

### 2.2 Color Correction Algorithm

Typically, color correction approaches rely on the use of a known target, a so-called “color chart” to calibrate color response characteristics of the cameras in the system, like in Refs. 22 and 23. These methods adjust camera parameters based on the known properties of the color chart. However, due to the practical limitations, it is not desirable to use color charts for every camera in the multiview setup. Methods that do not use color charts can be subdivided into global transformation techniques, and nonlinear modification techniques. Local methods are favored over global ones because of the accuracy of the estimation and content dependencies. For example, the method from Ref. 26 relies on spatiotemporal information, but also requires computation of disparity maps. Color matching/correction is also applied in multiview video coding in order to increase encoding gains, but the use of macroblocks results in lower precision. In Ref. 28, real-time color correction method is presented using fast 3-D lookup tables. In the case when the reference view is missing, the views can be corrected to match the average color of the set of views.

#### 2.2.1 Proposed color correction method

Due to the different settings of the cameras, the color of the same object in the multiview setup often appears differently. In order to avoid further quality degradation mentioned in Sec. 1, we adopt an approach where the central view is set as a color reference, reusing the correspondences between views to perform warping of all frames to the central view. In order to obtain more accurate color correction, we perform the segmentation of each image onto segments located on a rectangular grid like in Ref. 20, preceded by detail reduction filtering in the wavelet domain. Due to the edge-avoiding wavelets, we can extract the base layer by removing details from the image, but, in the same way, preserving edges. Next, segmentation partitions the image into segments, which have slowly varying color, while at the same time follows contours of the objects. In general, color mapping in RGB color space requires nonlinear mapping functions. One way to use a simpler, linear, transfer function is to perform color mapping in an alternative color space, such as YUV, CIELAB, Lab, as in the methods proposed in Refs. 26, 30, and 31. However, in our case, slowly varying colors in each segment allow us to use linear color mapping functions. In other words, the proposed method performs local color correction, which additionally relaxes the nonlinearity of the transfer function. By relying on warping functions and segmentation, we estimate the parameters of color correction function locally for each color using linear regression. Local color values within each segment are mapped using the expressions

\[ R' = k^R_R R + c^R \]

\[ G' = k^G_G G + c^G \]

\[ B' = k^B_B B + c^B, \tag{5} \]

where \( R', G', B' \) are modified color values; \( k^R_R, k^G_G, k^B_B \) are slopes of linear function in the segment; \( c^R, c^G, c^B \) are...
offsets of linear function in the segment $l$; and $R, G, B$ are original values of colors of a coefficients from the segment $l$. One way to observe the proposed approach is as a global mapping function approximated by a number of local, linear mapping functions.

Due to the detail reduction, color variations inside each segment are reduced, which facilitates the estimation of color-mapping parameters. We perform color correction in the wavelet domain by mapping the colors in the base layer using the estimated color correction parameters and by restoring the detail coefficients.

For each segment $l$, we estimate the slopes and the offsets by relying on the values of the warped coefficients from the reference frame and coefficients from the frame which needs to be color corrected and by fitting a first-order polynomial to the data. Each segment typically contains sufficient number of points to perform this task reliably. First, values of the source coefficients that need to be corrected are normalized to the range [0,1]. Next, a Vandermonde matrix is constructed from the source coefficient, followed by QR decomposition to orthogonal matrices $Q$ and $R$. Finally, the values of the slopes and the offset of each correction function are obtained from QR decomposition and from the values of the warped coefficients from the current segment of the reference frame.

If applied globally, linear color correction functions may lead to uneven color correction, since the characteristics of the optics, sensor, and hardware processing inside the camera vary spatially in the image. By performing color correction locally, using the affine function it is possible to compensate spatial variability and nonlinearity of the correction function. To visualize the equivalent transfer function obtained by the proposed approach, we generate the equivalent transfer function for all three colors, using the microstereopsis pair shown in Fig. 7. In Fig. 8(a), we show the histogram of the blue color component of the image pair shown in Fig. 7. As we can see, coefficient values are concentrated in the range [0,150], which is also noticeable in the image, since dark tones are dominating the scene. As stated earlier, color correction in each segment is performed using locally affine transfer function of the form $y = ax + b$. In Fig. 8(b), we show average values of the estimated offset coefficient $b$ for each value of the input coefficient and each color. These curves are obtained by observing all segments and ranges of input values in each of them and calculating cumulative correction function using local correction functions as inputs.

One can notice that these offsets have the highest values for the green component of the image, over the whole value range, which corresponds to the real situation, since the source image has more pronounced green tones compared to the reference image. Corrections applied to other two channels are much smaller. In Fig. 8(c), we show cumulative values of the local slopes $a$ from the linear color correction model. For coefficient values higher than 60, the values of the slopes of the green color component tend to be higher than the values for other two colors. Finally, cumulative values of the correction function shown in Fig. 8(d) confirm that in total, the green color channel needs the most corrections in this practical case.

However, the most important conclusion that can be drawn from the graphs shown in Fig. 8 is that the highly nonlinear color correction function, as the one shown in Fig. 8(d), can be approximated using a number of locally affine functions. Moreover, these correction functions are applied locally, which facilitates correcting various nonlinearities of optics and processing algorithms in the camera.

### 2.3 Multiview Sharpness Correction

In order to improve the visual quality of each video stream within the multiview setup as well as the performance of disparity estimation and view interpolation algorithms, we perform sharpness correction. Certain areas in views are differently focused, which results in blur occurring in different regions for different views. In order to improve the sharpness of each view, we combine different views by selecting prominent edges and transferring them to blurred regions of other views. If noise is present in the sequence, its high-frequency content could be erroneously introduced into other views. This was the main motivation to perform the denoising step along motion trajectories for each stream, while also relying on high-frequency content of images corresponding to different views.

One of very few methods dealing with this challenge was presented in Ref. 32, where Doutre and Nasiopoulos propose a fast method in the DCT domain to improve sharpness in blurred regions of stereo images, a method that estimates the energy of the signal and noise in each band and scales the coefficients so that the two images have equal signal energy in each band. Another closely related reference was presented in Ref. 33, where Petrovic et al. propose a method for virtual view synthesis for undersampled light fields, where the fusion of differently focused regions in the image is also performed.

#### 2.3.1 Proposed sharpness correction method

The first step in the proposed method is the warping of neighboring views to the reference view. Similar to the first step—
to avoid artifacts due to the occlusion—we compare the differences between views and discard locations that would create artifacts, as in Eq. (2). Since warping is performed bidirectionally, pixels occluded in one view can often be found in its symmetrical view relative to the reference frame. Edges from regions in focus typically produce a larger wavelet coefficient, which we use here as a criterion for selection of edges in focus from different images.

We apply two different strategies for scaling and wavelet coefficients. In the low-pass bands corresponding to different views, we iterate through all views and select scaling coefficient with maximal amplitude as

$$b_v(i, j) = \arg\max_v |a_v(i, j)|,$$

(6)

$$A_v(i, j) = a_v[i, j, b_v(i, j)],$$

(7)

where $b_v(i, j)$ is the ordinal number of the view corresponding to the highest coefficient values and $a_v(i, j)$ is the value of the scaling coefficient at the location with coordinates $(i, j)$, belonging to the view $v$. This way of selecting scaling coefficients leads to improvements in contrast.

In wavelet bands that contain information about image details, we propose a different approach of selecting views that are in focus. Here, we again rely on the segmentation of the processed view. For each of the segments $p_i$ of each view $v$ in the system, we calculate the sum of absolute values of wavelet coefficients at the location $l$, orientation $o$, and scale $c$, $s_{c,l,o}$.

$$F_{p_i} = \sum_{l \in p_i} \sum_{c,o} |s_{c,l,o}|.$$  

(8)

In such a way, we obtain estimates of focus level per segment for each of the views, as shown in Fig. 9. For each of the segments, we find the view with the maximal value of the function defined as

$$v_{p_i}^{\max} = \arg\max_v \left( |F_{p_i}| + \alpha \sum_{k \in N} |D_{p_k} - D_{p_i}| \right),$$

(9)

Fig. 8 (a) Histogram of the base layer coefficients of the blue color. (b) Affine component of the color correction function averaged over all segments and input values. (c) Slope of the correction function averaged over all segments and input values. (d) Average value of the color correction for each value of the coefficient.
where $\alpha$ is regularization constant, $D_{p_i}$ is the sum of disparities in the current segment, and $D_{p_k}$ is the sum of disparities in segments belonging to the neighborhood $N$. We use the disparities here as a secondary feature for selection of focused segments. When the scene is recorded using multiple cameras, an object in the scene takes different positions in each of the images. The closer the object is to the camera, the larger the difference in its position in the images acquired using different cameras. If the object is far from the camera, the differences in its $x$-coordinate are much smaller. Therefore, the disparity field provides us with the segmentation of the scene into regions at different distances from the camera. Disparity fields give implicit information about the level of focus of different objects in the scene; in real scenes, the level of focus remains approximately constant per object, while objects at different distances are differently focused. The disparity fields we use in this method provide clear segmentation onto object depending on their distance. The second part of Eq. (9) has a goal to penalize the case when a certain segment has significantly different sharpness than the neighboring segments. Division of the image onto segments and finding maximal value per segment significantly accelerates the processing, since the average segment contains 256 pixels. The final value of the wavelet coefficients is obtained as a weighted sum of wavelet coefficients corresponding to different views

$$s_{est, o, p_i} (i, j) = \sum_{v=1}^{V} w_v p_i s_{v, o, x, p_i},$$

where the weights $w_v$ are defined as

$$w_v = \frac{1}{K} e^{-\frac{a}{s_{max} (i, j) - s_{v, o, x, p_i}}}.$$

where $s_v (i, j)$ is the wavelet coefficient from the view $v$ at the location $(i, j)$, $s_{max} (i, j)$ is the value of the wavelet coefficient corresponding to the view defined by Eq. (9), $a$ is the shape parameter adjusting the influence of wavelet coefficients from other views, and $K$ is the normalization constant equal to the sum of all weights. Wavelet coefficients belonging to the view with maximum value $F_{p_i}$ therefore receive the highest weight values. We finally obtain the result by performing the inverse wavelet transform of the selected coefficients and weights.
3 Experimental Results

In this section, we evaluate the proposed method using some standard multiview sequences like “Vassar,” “Ballroom,” and “Exit” and a microstereopsis video sequence. The results of the denoising algorithm are shown in Figs. 10–12. As we can see, the noise is more pronounced in darker regions, which is a challenge for denoising algorithms.

However, due to the local noise estimation incorporated in the proposed method and spatially adaptive denoising that adjusts to the local noise variance, noise is uniformly removed from the video sequence. Due to the highly accurate motion estimation and the use of edge-avoiding wavelets, the borders of the images and details are well preserved. Moreover, due to highly precise motion estimation, the proposed method is able to cope with the correlated noise. Moreover, we also evaluate the denoising performance using PSNR measure defined as

$$\text{PSNR} = 20 \log \frac{255}{\sqrt{\frac{1}{N} \sum_{x} \sum_{y} |I'(x,y) - I(x,y)|^2}}, \quad (12)$$

where $x$ and $y$ are pixel coordinates, $N$ is the number of pixels in the image, $I'$ is the estimated image obtained using the proposed method, and $I$ is the reference noise-free image. We evaluate the performance of the proposed method using high-quality images from the light-field images database of Stanford University to which we add artificial noise, which corresponds to the characteristics of the real noise, described in Sec. 2.1. The quantitative results achieved by applying the proposed multiview denoising method are given in Table 1. In this experiment, we have applied the proposed method to 7 views corresponding to the one time instant. When the artificial noise is added to the “Tarot” multiview dataset, the PSNR value drops to 28.36 dB. After applying the proposed denoising method, the PSNR value increases to 36.71 dB. For the “Knights” multiview dataset, the PSNR of the noisy central image is 28.24 dB, while the PSNR of the denoised image is 37.43 dB. On the other hand, the NLM-based method achieves 35.21 dB on the “Tarot” dataset and 34.08 dB on the “Knights” dataset, which is significantly less than the proposed method. While removing noise efficiently, edges remain preserved, as can be seen in Fig. 10. An enlarged detail of the “Exit” sequence shown in Fig. 11(a) demonstrates that noise is nonwhite and has spatially variable standard deviation. Moreover, other artifacts, such as blocking, are visible in the noisy image. The denoised detail shown in Fig. 11(b) demonstrates that all the artifacts were uniformly removed while preserving edges and texture. To further evaluate the proposed method in video denoising tasks, we have applied it on a “Flower Garden” video sequence, containing panning camera motion of a constant speed. The results of this experiment are given in the Table 2, in the case when artificial noise with standard deviation $\sigma = 15$ is added. As we can see, the proposed method outperforms some of the state-of-the-art video denoising methods.

Next, we present the results of the proposed method applied on a microstereopsis video sequence for each of the problems treated. Due to the beam splitter inserted between the lens and camera sensor, the left and right views shown in Fig. 7 contain an increased amount of signal dependent noise; its color differs from the second view and certain regions of the image are blurred. A more detailed view of differences between the views is shown in Fig. 13. In Figs. 13(a) and 13(b), one can see that the colors and noise levels differ significantly between the two views. Moreover, noise levels also differ due to the prism inserted in front of the sensor. Figures 13(c) and 13(d) show significant

![Fig. 12](image)

(a) Noisy “Vassar” sequence. (b) “Vassar” sequence denoised using the proposed approach.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>“Tarot”</th>
<th>“Knights”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td>28.36</td>
<td>28.24</td>
</tr>
<tr>
<td>NLM</td>
<td>35.21</td>
<td>34.08</td>
</tr>
<tr>
<td>Proposed</td>
<td>36.71</td>
<td>37.43</td>
</tr>
</tbody>
</table>

Table 1: The performance of the proposed cross-view denoising algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>“Flower”</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRSTF</td>
<td>28.19</td>
</tr>
<tr>
<td>VBM3D</td>
<td>29.81</td>
</tr>
<tr>
<td>Proposed</td>
<td>30.15</td>
</tr>
</tbody>
</table>

Table 2: The performance of the proposed multiview algorithm as a video denoising algorithm.
differences in sharpness. Not only that these differences create disturbing artifacts but they also cause lower performance of disparity estimation algorithms. In the first processing step, we apply the temporal denoising described in Sec. 2.1. Relying on the segmentation obtained in the denoising step, we perform the color correction described in Sec. 2.2. The first step is the warping of the right view to the left, so that the objects in the scene become aligned. Next, using the left view and warped right view, selection color transfer is performed from the reference right to the corrected left image. The results of this processing step are shown in Fig. 14(a). As we can see, in the resulting image shown in Fig. 14, the color gamut of the left image is now very close to the gamut of the right reference view, while defocused regions are still visible.

We have also evaluated the color correction algorithm using the PSNR measure defined in Eq. (12). The performance of the proposed color correction is compared to two reference color correction methods. The first reference method, presented in Ref. 35, performs gamut correction in the YUV color space based on histogram matching as a pre-filtering step before multiview coding step, to improve coding gains. The second reference method26 performs global gamut correction based on linear regression in the YUV space. We give the results of the quantitative comparison of the methods in Table 3. The performance of the proposed gamut correction algorithm is evaluated, without performing

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEGC35</td>
<td>35.4968</td>
<td>37.7421</td>
<td>34.8006</td>
</tr>
<tr>
<td>GLRC26</td>
<td>36.3050</td>
<td>34.8288</td>
<td>32.0983</td>
</tr>
<tr>
<td>Proposed</td>
<td>36.9776</td>
<td>38.0370</td>
<td>36.0510</td>
</tr>
</tbody>
</table>

Fig. 13 (a) and (b) Differences in colors and noise levels. (c) and (d) Differences in sharpness.

Fig. 14 Images after the proposed processing steps: (a) resulting image after color correction processing step; and (b) left microstereopsis image after denoising, sharpness, and color correction.
the other two processing steps. Visual comparison of the results is shown in Fig. 15. The gamut of the noncorrected left image shown in Fig. 15(a) differs significantly from the gamut of the target image shown in Fig. 15(e). By applying global color correction methods from Refs. 26 and 35, the dominant green tone is corrected, as shown in Figs. 15(b) and 15(c). However, objects like bricks and the Lego toy box still have colors that differ from the desired colors shown in Fig. 15(e). By applying the proposed method, colors become more consistent with colors of the target image, which can be best observed by comparing images of the target image shown in Fig. 15(e) and the result of the proposed method shown in Fig. 15(d). These observations are confirmed by objective comparisons in terms of PSNR given in Table 3.

Finally, we perform correction of defocused regions described in Sec. 2.3. The final result of the method with integrated focused regions from the right view is shown in Fig. 14(b). As we can see, the final results not only contain sharpened details from the right image as in the champagne label but also preserve details that appear sharper in the left image, like the thread coils in the upper part of the image. An enlarged image of one of the blurred regions from the original image around the label of the champagne bottle is shown in Fig. 16(a). If only color correction is performed, which is typically the case in multiview systems, the result will still contain noise and blurred regions, which is especially visible in Fig. 16(b). While color balance is important, due to the other two types of artifacts, visual quality is still far from satisfactory. The enlarged detail shown in Fig. 16(c) shows significant improvement after applying the proposed method. Only by applying the proposed noise reduction technique together with the sharpness enhancement, the color correction step yields the necessary quality improvements, which can be seen especially around the champagne label and around Lego bricks, where the resulting image becomes much sharper, while at the same time containing significantly less noise. As we can see from Figs. 7(b) and 16(b), all artifacts have been successfully removed and the image sharpness becomes uniform across the whole image. The proposed algorithm was tested using multiple sets of multiview images, containing different levels of noise, motion, and occlusions. Owing to the bidirectional motion and disparity estimation, the proposed method works well in the presence of occlusions and large motions. Increased noise levels will certainly cause lower visual quality in the result.

Following this, we show the influence of the proposed algorithm on the quality of the interpolated views. Due to the different color gamut, noise characteristics, and focus, interpolated views between two noncorrected views contain varying color gamut, blur, and noise level, as shown in Fig. 17. By applying the proposed method, these variations are significantly reduced, as shown in Fig. 18. In order to assess the improvements of the visual quality of the interpolated views, we show the enlarged details of the interpolated views in Fig. 19. If the view interpolation is performed on raw multiview pairs, the results shown in Figs. 19(a) and 19(b) will contain different levels of noise, since the image pairs are typically contaminated with noise of different intensities, which is best visible at the table surface, bricks, and the champagne label. Moreover, due to the color mismatch, the interpolated images will gradually change color. In the

![Fig. 15: The results of the proposed color correction algorithm compared to the results of the reference methods: (a) original image before the color correction, (b) the result of the method presented in Ref. 26, (c) the result of the method presented in Ref. 35, (d) the result obtained using the proposed method, and (e) the reference frame.](http://electronicimaging.spiedigitallibrary.org/fig15.png)
particular test sequence, one may notice that the example shown in Fig. 19(b) is dominated by light green tones, while the image in Fig. 19(a) is not biased toward green. The image shown in Fig. 19(b) also suffers from severely blurred regions, which is especially visible in regions which contain letters. Due to these distortions, the level of noise in interpolated images will gradually change, while certain regions will become (de)focussed, which is again best visible in regions containing details, like the box in the background. If we apply the proposed method, interpolated views will not contain color variations nor sharpness will change as we change the virtual viewpoint. Details remain intact as shown in Fig. 19(c).

We have implemented the proposed method in MATLAB®, while the optical flow was implemented in mex c-file. The processing time of our method per one camera, for the frame resolution of $640 \times 480$, and on a computer based on Intel i7 860 CPU at 2.8 GHz, using a seven-frame temporal buffer and four additional views corresponding to different camera as an input is 5.5 s. Owing to the shared processing blocks, the
processing time is significantly reduced. The most time-consuming operation in the processing chain is certainly disparity/motion estimation. Due to the highly optimized code, it takes 2.5 s to compute the optical flow, denoising takes another 1.5 s, sharpness correction additional 0.89 s, and color correction 0.61 s.

4 Conclusions and Future Work

In this paper, we have presented a method for simultaneous denoising, sharpness improvement, and color correction in multiview systems. The proposed approach significantly improves the visual quality of the videos, while keeping the complexity low. In future work, we will focus on accelerating the method and achieving real-time operation, while investigating the influence of the proposed method on the visual quality of the interpolated views and coding gains achieved by applying the proposed methods as a compression preprocessing step.

Acknowledgments

This work was supported by the iMinds through the ICON ASPRO+ project and the CATRENE through the CA308 ICAF project.

References


**Ljubomir Jovanov** received his dipl. ing. and mr. ing. degrees in electrical engineering from the University of Novi Sad, Serbia, in 2000 and 2005, respectively. In 2011, he earned a PhD in applied sciences from Ghent University, Belgium. His main research interests are in the field of quality enhancement of color and depth video, including denoising, super-resolution, sensor fusion between the two modalities and multiview video processing.

**Hiêp Luong** received his BSc and MSc degrees in computer science engineering from Ghent University, Belgium, in 2001 and 2003, respectively. In 2009, he received his PhD in computer science engineering at Ghent University. His main research interests are in the domains of image and video restoration, including super-resolution, deblurring and denoising, biomedical imaging, depth and multiview processing, remote sensing and document processing.

**Wilfried Philips** received his diploma degree in electrical engineering in 1989 and in 1993 his PhD in applied sciences, both from Ghent University. Since November 1997, he has been a senior full-time professor at Ghent University. His main research interests are in the domains of image and video restoration, segmentation, and analysis with applications in multimedia, biological, and medical image processing and real-time collaborative computer vision for surveillance and security.