TEMPORALLY CONSISTENT HANDLING OF DISOCCLUSIONS WITH TEXTURE SYNTHESIS FOR DEPTH-IMAGE-BASED RENDERING

M. Köppel[1], P. Ndjiki-Nya[1], D. Doshkov[1], H. Lakshman[1], P. Merkle[1], K. Müller[1], and T. Wiegand[1,2]

[1]Image Processing Department
Fraunhofer Institut for Telecommunications
Heinrich-Hertz-Institut
Einsteinufer 37, 10587 Berlin, Germany

[2]Image Communication Chair
Department of Telecommunication Systems
Technical University of Berlin
Einsteinufer 17, 10587 Berlin, Germany

ABSTRACT

Depth-image-based rendering (DIBR) is used to generate additional views of a real-world scene from images or videos and associated per-pixel depth information. An inherent problem of the view synthesis concept is the fact that image information which is occluded in the original view may become visible in the “virtual” image. The resulting question is: how can these disocclusions be covered in a visually plausible manner? In this paper, a new temporally and spatially consistent hole filling method for DIBR is presented. In a first step, disocclusions in the depth map are filled. Then, a background sprite is generated and updated with every frame using the original and synthesized information from previous frames to achieve temporally consistent results. Next, small holes resulting from depth estimation inaccuracies are closed in the textured image, using methods that are based on solving Laplace equations. The residual disoccluded areas are coarsely initialized and subsequently refined by patch-based texture synthesis. Experimental results are presented, highlighting that gains in objective and visual quality can be achieved in comparison to the latest MPEG view synthesis reference software (VSRS).

Index Terms— View synthesis, inpainting, texture synthesis, 3D video, Depth-image-based rendering.

1. INTRODUCTION

The popularity of 3D video, free viewpoint television and 3D displays is growing significantly and many 3D video products are currently entering the mass market [1]. Autostereoscopic displays provide a 3D perception without the need to wear additional glasses. Such a display shows many slightly different views (e.g. 5, 8, 9 or 22) at the same time. However, due to the physical limitation of cameras and the bandwidth of communication channels, only a limited number of original views can be stored and transmitted. Hence, the need to render additional “virtual” views arises, in order to support autostereoscopic multi-view displays.

Depth-image-based rendering (DIBR) is a technology for synthesizing novel realistic images at a slightly different view perspective, using a textured image and its associated depth values. A critical problem is that the regions occluded by foreground (FG) objects in the original view may become visible in the synthesized view. In the literature, two basic options are described, addressing this problem. Either the missing image regions may be replaced by plausible color information [2] or the depth map is preprocessed in a way that no disocclusions appear in the rendered image [3]. One disadvantage of existing approaches is that they support only small baselines for rendering. Furthermore, most approaches render the new images frame by frame, ignoring the temporal correlation of the filled area [2], [3].

An appropriate technique to fill missing image regions with known information is texture synthesis [4], [5]. This method operates in parametric [6] or non-parametric [4], [7] modes. Although parametric methods are faster, non-parametric methods result in better visual quality [8].

In this paper, a new approach for handling disocclusions in 3D video is presented. The method is based on non-parametric texture synthesis. Temporal correlations between different pictures are taken into consideration via a background (BG) sprite. A robust initialization gives an estimate of the unknown image regions, which is refined during the synthesis stage.

2. VIEW SYNTHESIS FRAMEWORK

Fig. 1 outlines the proposed framework as a block diagram. We present our approach assuming that the BG is static. For moving BG, a motion estimation stage needs to be included to compensate for the BG motion. Then, texture filling can be conducted as described in this paper.

To compute the depth maps (DM) the method presented in [9] is used. The method introduced in [10] is utilized to warp an original image, using the associated DM into the new “virtual” viewing position. The displacement for scene content from an original to the new viewing position depends on the depth value. FG and BG objects are therefore warped differently, causing uncovered image regions.

The goal of the new view synthesis algorithm is to fill these disocclusions (holes), which become visible in the “virtual” view and the DM, in a visually plausible manner. Furthermore, the synthesis should be temporally stable, meaning that information from previous frames must be taken into consideration. This is

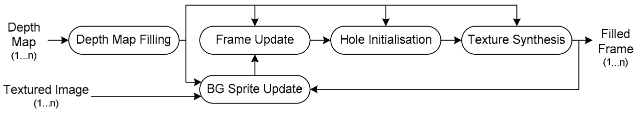


Fig. 1. Block diagram of the proposed approach. First, disocclusions in the DM are filled. Next, the BG sprite is updated with original BG data and the holes in the current picture are filled from the BG sprite. Then, the remaining holes are initialized and refined with texture synthesis. Finally, the BG sprite is updated with the new synthesized texture.

We would like to thank the Gwangju Institute of Science and Technology in Korea and the Electronic and Telecommunications Research Institute / MPEG Korea Forum for providing the "Newspaper" and "Lovebird1" sequence respectively.

achieved with an image called a BG sprite, which is of the same size as a frame and stores BG information from previous frames. In a first step, the disoccluded areas in the DM are filled. The BG sprite is then updated with known BG information from the current picture. Next, the holes in the current picture are updated from the BG sprite. The remaining holes are initialized from the spatially adjacent original texture, providing an estimate of the missing information. In the next step, patch-based texture synthesis is used to refine the initialized area. The BG sprite is finally updated with the synthesized image information.

3. FILLING DISOCCLUSIONS IN THE DEPTH MAP

From the inherent properties of the applied warping, larger uncovered areas mostly belong to BG objects. The DM is an 8 bit gray scale image, denoted as D in the following The closest point to the camera is associated with the value 255 and the most distant point is associated with the value 0 (cf. Fig. 2 (a)-(c)). The uncovered areas in the DM are denoted as Ψ. Due to inaccuracies in depth estimation, FG objects may be warped into Ψ (cf. Fig. 2 (a)). Therefore, small blobs up to γ pixels in Ψ are assigned to Ψ because they are assumed to correspond to noise and may otherwise lead to noticeable inaccuracies in the post-processed DM (cp. Fig. 2 (b) and (c)). The holes in the DM are filled line-wise from the BG. The filling direction for the example in Fig. 2 (a) is marked with red arrows. One possibility is to fill the last known BG depth value D_i line-wise into Ψ (Fig. 2 (c)). However, relying on a single D_i value can be error-prone. For that reason, in this work, the spatial neighborhood at a location i is clustered into two centroids, c_{min} and c_{max}, representing FG and BG. These are computed via k-means clustering ($k = 2$). The neighborhood is given by an squared area of $M \times N$ pixels, centered at location i. Given c_{min} and c_{max}, the selection criterion for the depth values to be filled into Ψ is defined as follows:

$$D_{\mathrm{j}} = \begin{cases} D_i \text{ if } D_i \leq \mathrm{c}_{min} \\ \mathrm{c}_{min} \text{ otherwise} \end{cases}, \quad j \in \Psi; i \in D/\Psi. \tag{1}$$

4. SPRITE AND IMAGE UPDATION

The BG information and its associated depth values from previous pictures are stored as the BG sprite S and DM sprite G. These sprites accumulate valuable information for rendering subsequent images. In fact, by referencing the sprite samples for filling unknown area in the current picture, the synthesis is temporally stabilized.

4.1. Sprite updation

For each new picture, the depth values of all pixel positions are examined. All pixels with a depth value below $\overline{\mathrm{c}_{min}}$ are considered for sprite update, where $\overline{\mathrm{c}_{min}}$ is the median c_{min} value in the current picture. Depth values below $\overline{\mathrm{c}_{min}}$ describe the BG, while the remaining are assigned to the FG. Depth estimates at BG-FG transitions and along the uncovered area Ω, are considered unreliable. Therefore, a two pixels wide area around the unreliable regions is not considered for sprite update. The remaining locations with $D_i < \overline{\mathrm{c}_{min}}$ are stored in the BG and DM sprites, respectively. After the synthesis step (cf. Sec. 5 and Sec. 6), newly synthesized textures and depths are incorporated in the sprites as well.

4.2. Image updation

Every picture is updated from the BG sprite, whereby pixel positions corresponding to pixels in the BG sprite with non-assigned background information are ignored. The pixel positions in S to be used for updating the current picture P, are selected as follows:

$$\mathrm{P}_{\mathrm{j}} = \begin{cases} S_j \; if \; D_j < G_j + \beta \\ 0 \text{ otherwise} \end{cases}, \quad \forall j \in \Omega, \tag{2}$$

where S_j and G_j represent the gray level at location j in the BG and the DM sprite, respectively. The parameter β allows for some variance in the local BG depth label. Eq. (2) is applied to the chroma channels in the same way. In order to account for illumination variations, the covariant cloning method [13], [7] is utilized to fit the BG sprite samples to the color distribution in the relevant neighborhood of the current picture. The cloning method is adapted to the given view synthesis framework by ensuring that only BG pixels in the current picture are considered as valid boundary conditions.

5. INITIAL FILLING OF TEXTURED IMAGES

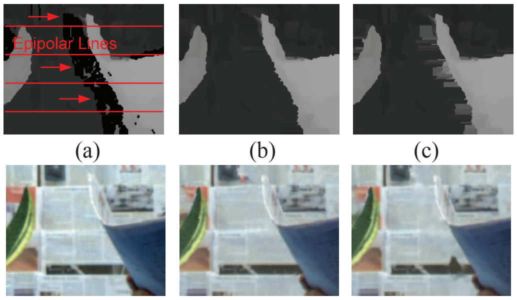
In a first step, the Laplacian equation [11] is used to fill small holes in the current image. For the reconstruction of smooth regions this method gives satisfactory results. Good visual results are observed for holes smaller than γ pixels (e.g. 50 pixels), where Laplace cloning is about 10 times faster than patch-based texture synthesis. Hence, small holes are regarded as finally filled and are not considered in the texture refinement step.

For holes larger than γ pixels, we have shown in our previous work [12] that the visual results of texture synthesis can be improved by using an initial estimate of pixel values. In this paper, we present an initialization method that is based on the statistical properties of known samples in the vicinity of Ω. Commonly, the known samples constitute valid BG pixels, but in some cases the depth information at the FG-BG transition are not reliable. Hence, the probability distribution of known BG pixel values in the spatial neighborhood of the hole area is observed to be skewed. To obtain the BG value from the spatially adjacent samples, the Median estimator is used, which is the standard measure of (end value) location used in case of skewed distributions.

A window of samples centered on the pixel to be filled is considered. For each unknown pixel, a measure N_{BG} is set equal to the number of known pixels that are classified as BG in the current window. The unknown pixels are visited in decreasing order of N_{BG}. A 2D median filter operates on the BG pixels in the current window and the filtered output is used to initialize the unknown pixel. The filtering operation can be viewed as the process of extracting a valid BG value from the spatially neighboring samples. This serves as a coarse estimate for the texture synthesis stage that can recover the details in the unknown region. Using the described initialization scheme, the sensitivity of the texture synthesis stage to outliers in the transition region is significantly reduced.

6. TEXTURE REFINEMENT VIA SYNTHESIS

In texture synthesis techniques the unknown region is synthesized by copying content from the known parts ($\mathrm{P} - \Omega$) to the missing parts (Ω) of the image. Patch-based texture synthesis is used in this work to refine the initialized areas. To determine the patch filling order, the method introduced in [4] is utilized and additionally enhanced in two ways. Firstly, already estimated samples through initialization are regarded in further processing steps. Hence, the gradient is calculated for all the samples in the current patch, which leads to a better isophote direction (Please refer to [4] for the original algorithm). Secondly, the filling order is steered such that the synthesis starts from the BG area towards the FG objects. After these modifications, the remaining locations in the border of



(a) (b) (c)

(d) (e) (f)

Fig. 2. Results for frame 1 of the "Newspaper" sequence for the proposed DM and texture filling approach. (a) DM with disoccluded area marked black (filling direction given by red arrows). (b) Result of proposed DM filling approach. (c) Line-wise filling of DM without blob removal. (d) Original reference image. (e) Result of the proposed approach. (f) Result of MPEG VSRS.

the hole are assigned filling priorities according to [4]. In the following, the patch at the current location to be filled is denoted as $\boldsymbol{c}$. Its center is denoted as $\boldsymbol{c}_{center}$. An area of $5M \times 5N$ around $\boldsymbol{c}_{center}$ is defined to be the source area $\boldsymbol{s}$. The filling algorithm now searches for a patch $\boldsymbol{x}$ in $\boldsymbol{s}$ that is similar to $\boldsymbol{c}$. In the matching routine only the luminance channel is considered. Given the filled DM, the depth value of $\boldsymbol{c}_{center}$ is always known. All pixel positions in $\boldsymbol{s}$ with depth values higher than $D_{\boldsymbol{c}_{center}} + \beta$ are excluded from the search area. Therefore, patches will not be taken from areas with depth values much higher than the current region to be filled, such that foreground objects are actively excluded. To speed up the matching procedure the source area is sub-sampled by a factor of 2. The remaining source positions are used as center positions for $\boldsymbol{x}$. The best continuation patch out of all candidate patches in the source area is found by minimizing the following cost function:

$$E = \sum_{i=1}^{K} \|\boldsymbol{x}_i - \boldsymbol{c}_i\|^2 + \omega_\Omega \sum_{j=1}^{K_\Omega} \left\|\boldsymbol{x}_j - \boldsymbol{c}_j\right\|^2 + \\ \omega_\nabla \sum_{i=1}^{K} \|\nabla\boldsymbol{x}_i - \nabla\boldsymbol{c}_i\|^2 + \omega_\nabla \omega_\Omega \sum_{j=1}^{K_\Omega} \left\|\nabla\boldsymbol{x}_j - \nabla\boldsymbol{c}_j\right\|^2 \tag{3}$$

where K is the number of original and K_Ω is the number of initialized pixels in $\boldsymbol{c}$. $\nabla\boldsymbol{c}$ is the gradient map of $\mathbf{c}$ and $\nabla\boldsymbol{x}$ is the gradient map of $\boldsymbol{x}$. ω_Ω is the weighting factor for the initialized values in Ω and ω_∇ is the weighting factor for gradient component in the matching process. In the last term the weight is given by $\omega_\Omega\omega_\nabla$ because here, the distance between $\boldsymbol{x}$ and $\boldsymbol{c}$ is determined both in Ω and on sample wise gradient map. To ensure smooth transitions between adjacent patches, an efficient post-processing method, based on covariant cloning, is utilized [7]. This post-processing approach is adapted to the framework in such a manner that FG objects are not considered as boundary pixels.

7. EXPERIMENTAL RESULTS

Table 1. PSNR and SSIM results by the proposed framework and the view synthesis reference software.

Seq.	Camera	PSNR (dB)		SSIM	
		Prop.	MPEG	Prop.	MPEG
Book.	8 to 9	**37.26**	36.06	**0.9836**	0.9828
Book	10 to 9	**35.63**	35.15	**0.9827**	0.9810
Book.	8 to 10	**31.28**	30.25	**0.9553**	0.9525
Book	10 to 8	**30.58**	30.30	**0.9551**	0.9524
Loveb.	6 to 8	40.90	**42.13**	**0.9302**	0.9284
Loveb.	8 to 6	**39.78**	38.54	**0.9444**	0.9425
Newsp.	4 to 6	25.30	**25.43**	0.8936	**0.8974**
Newsp.	6 to 4	**31.01**	30.37	0.9123	**0.9131**

We compared our results with those of the MPEG view synthesis reference software (VSRS) [10] version 3.6. For evaluating the proposed algorithm, three 3D video sequences, provided to MPEG by the Gwangju Institute of Science and Technology (Korea) are used: "Book arrival", "Lovebird1" and "Newspaper". They have a resolution of 1024 x 768 pixels. For every sequence, the rectified videos of several views with slightly different camera perspectives are available. The baseline between two adjacent cameras is approximately 65 mm. We consider 2 original but not necessarily adjacent cameras (cf. Table 1, "Camera" column) for every sequence. The following view synthesis operations are conducted: warping an original view (right and left) to the position of an adjacent view yielding a baseline of about 65 mm; baseline extension where the "virtual" camera position is 2 cameras away from the original camera location, giving a baseline of approximately 130 mm. For all experiments we choose the following parameters: $M = N = 32$ pixels, $\gamma = 50$ pixels, $\omega_\Omega = \omega_\nabla = 0.6$ and $\beta = 15$. The objective results given in Table 1 correspond to the mean PSNR and SSIM over all pictures of a sequence. The best result for every sequence is highlighted through bold face type. PSNR is computed locally, that is only for the defective area in the image, while SSIM is determined for the entire image. SSIM is used to assess the subjective visual quality [14], as PSNR is not an accurate measure thereof [14]. For the sequences "Book arrival" the proposed approach gives better SSIM and PSNR results than MPEG VSRS. For the "Lovebird1" sequence we obtain the best results in terms of SSIM. Nevertheless, the PSNR value of MPEG VSRS is better for the case "warping camera 6 to the position of camera 8" because although the VSRS rendering results show obvious artifacts, the color information in the synthesized area somehow correlates with the true information and is strongly blurred, while our result is sharper but quite noisy. For the sequence "Newspaper" VSRS gives the better overall results, because all our modules rely on the DM. However, the DM is particularly unreliable here, leading to visual and objective losses. Nevertheless, some visual and objective gains can be obtained for the case "warping camera 6 to the position of camera 4" (cf. Table 1 and Fig. 2 (d)-(f), electronic magnification may be required). Fig. 3 shows some visual and objective results. In Fig. 3 (a), the original reference picture 51 of the "Book arrival" sequence is shown. In Fig. 3 (b), the warped image is shown (baseline extension, warping camera 8 to the position of camera10). The disoccluded area is marked in black. In Fig. 3 (c) and (d), the final BG sprite and its associated DM are shown. The final rendering result by the proposed approach is shown in Fig. 3 (e). The result by VSRS is shown in Fig. 3 (f). Note that the proposed approach yields sharper edges than VSRS. Fig. 3 (g) and (h) are magnifications of the red squared area in Fig. 3 (e) and (f) where the result for the proposed algorithm is shown on the left side and the VSRS result on the right. Foreground information is correctly ignored for the recommended filling process (cf. Fig. 3 (g)). As it can be seen in Fig. 3 (h) on the poster in the background, details are well preserved by our method. Fig. 3 (i) and (j) show the objective results. Significant gains in PSNR and SSIM are achieved.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a new hole filling approach for DIBR. The algorithm works for large baseline extensions and the

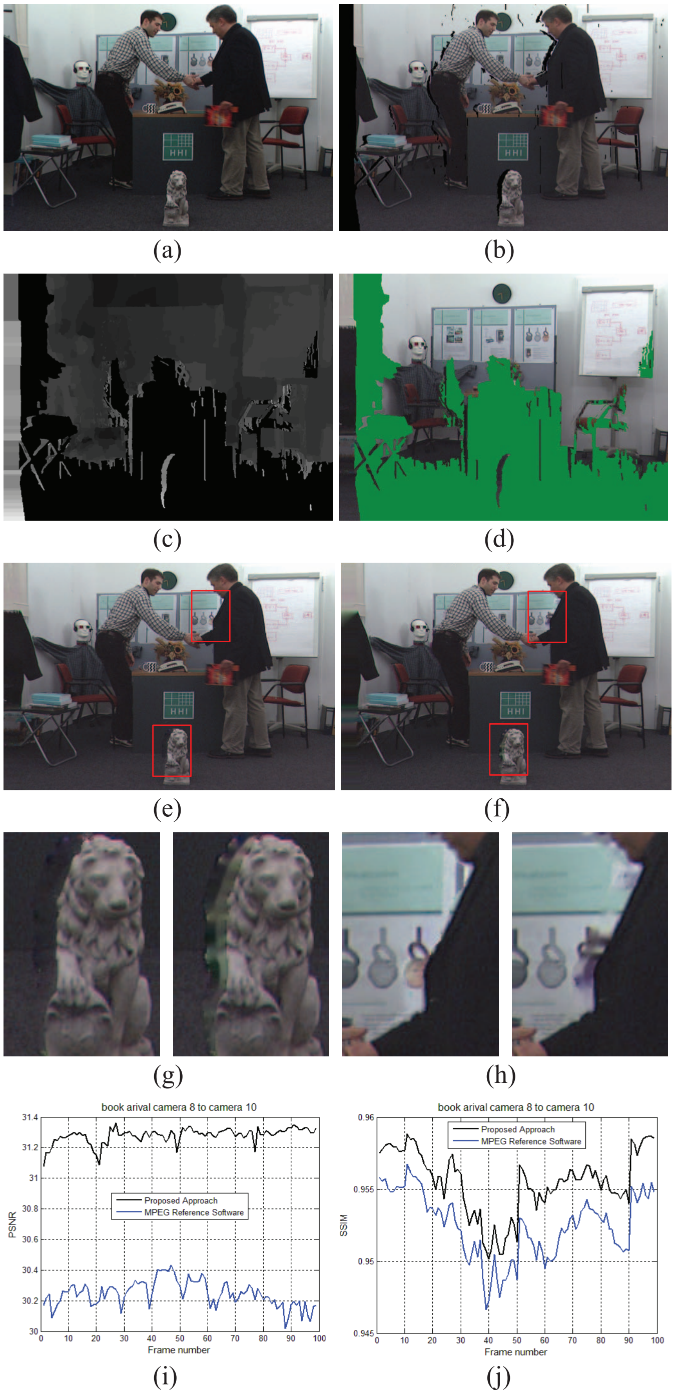

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 3. DIBR results for the "Book arrival" sequence. (a) Original reference image. (b) Warped image with uncovered area marked black. (c) Final BG sprite with unknown area marked green and its associated DM (d). (e) Result of picture 51 by the proposed approach. (f) Result of MPEG VSRS for the same picture. (g) and (h) Magnified results. Left, proposed approach and right, MPEG VSRS. (i) and (j) objective results for all frames of the sequence.

rendering results are temporally and spatially stable. Image information from previous pictures is considered via a background sprite from which further subsequent pictures are updated. The remaining uncovered areas are first coarsely estimated in an initialization step and subsequently refined via patch-based texture synthesis. We have shown that the presented approach yields both subjective and objective gains compared to the latest MPEG VSRS, given reasonable depth maps. However, depth estimation inconsistencies especially at foreground-background transitions may lead to considerable degradation of the rendering results. In future work, this dependency will be relaxed. Additionally, the problem of global and local background motion will be addressed.

9. REFERENCES

[1] A. Smolic, K. Müller, and A. Vetro, "Development of a New MPEG Standard for Advanced 3D Video Applications," *In Proc. of IEEE Int. Symp. on Image Signal Processing and Analysis (ISPA'09),* Salzburg, Austria, September 2009.

[2] C.-M. Cheng, S.-J. Lin, S.-H- Lai, and J.-C. Yang, "Improved Novel View Synthesis from Depth Image with Large Baseline," *In Proc. of Int. Conf. on Pattern Recognition (ICPR'08)*, Tampa, USA, December 2008.

[3] C. Fehn, "Depth-image-based Rendering (DIBR), compression and transmission for a new approach on 3D-TV," *In Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XI*, pp. 93-104, January 2004.

[4] A. Criminisi, P. Perez, and K. Toyama, "Region Filling and Object Removal by Exemplar-based Inpainting," *In IEEE Trans. on Image Proc.* vol. 13, no. 9, pp. 1200-1212, January 2004.

[5] J. Hayes, A. Efros, "Scene Completion Using Millions of Photographs," *In Proc. of ACM SIGGRAPH,* San Diego, USA, August 2007.

[6] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic Textures," *In Int. Journal of Com. Vision*, pp. 91-109, February 2004.

[7] P. Ndjiki-Nya, M. Köppel, D. Doshkov, T. Wiegand, "Automatic Structure-Aware Inpainting for Complex Image Content," *In Proc. of Int. Sym. on Visual Computing*, Las Vegas, USA, December 2009.

[8] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, "State of the Art in Example-based Texture Synthesis," *In Proc. of EUROGRAPHICS 2009, State of the Art Report, EG-Star*, Munich, Germany, 2009.

[9] M.Tanimoto, T. Fujii, and K. Suzuki," Depth Estimation Reference Software (DERS) 5.0," ISO/IEC JTC1/SC29/WG11 M16923, Lausanne, Switzerland, October 2009.

[10] M. Tanimoto, T. Fujii, and K. Suzuki, "View Synthesis Algorithm in View Synthesis Reference Software 2.0 (VSRS 2.0)," ISO/IEC JTC1/SC29/WG11 M16090, Lausanne, Switzerland, February 2008.

[11] P. Pérez, M. Gangnet, and A. Blake, "Poisson Image Editing," *In Proc. of ACM SIGGRAPH*, San Diego, USA, July 2003.

[12] H. Lakshman, M. Köppel, P. Ndjiki-Nya, and T. Wiegand, "Image Recovery using Sparse Reconstruction Based Texture Refinement," *In Proc of IEEE Int. Conf. on Acoustic Speech and Signal Proc.*, Dallas, USA, March 2010.

[13] T.G. Georgiev, "Covariant Derivates and Vision," *In Proc. Europ. Conf. on Comp. Vision (ECCV)*, Graz, Austria, 2006

[14] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, "Image Quality Assesment: From Error Visibility to Structural Similarity," *In IEEE Trans. on Image Proc*, vol. 13, no. 4, pp. 600-612, April 2004.