Grid based image warping for hole prevention in 3D applications

Author1: Mr.S.Yoganandhan,
Department of computer Science & Engineering, Akshaya college of Engineering & Technology, Anna University,Chennai, India.
Email: yoganandhan.cse@gmail.com

Author2: Ms.B.Vinitha Subashini,
Department of computer Science & Engineering, Assistant Professor, Akshaya college of Engineering &Technology, Anna University,Chennai, India.
Email: vinithasubashini@gmail.com

Abstract: A popular way to convert video is given by depth image-based rendering methods, in which a depth map that is associated with an image frame is used to generate a virtual view. We focus on the problem to generate the virtual videos of different viewpoints from a given video and associated depth maps at a viewpoint so that the viewers can realistically sense the depths. A spatio-temporal global optimization approach is proposed to synthesize the virtual views from a single video-plus-depth video sequence. Because of the lack of knowledge about the 3D structure of a scene and its corresponding texture, the conversion of 2D video, inevitably, however, leads to holes in the resulting 3D image as a result of newly-exposed areas. The new algorithm exploits the smoothness of a typical depth map to reduce the complexity of the underlying optimization problem that is necessary to find the deformation, which is required to prevent holes. Our proposed approach for the pre-processing of depth maps to prevent holes during the view synthesis. We propose to generate the virtual videos of different viewpoints from a given video and associated depth maps at a viewpoint so that the viewers can realistically sense the depths. We also propose a spatio-temporal global optimization approach that formulates an energy function considering depth map, image structure, texture information and patch shift. We also propose a new and efficient adaptation of the regular grid-based warping.

Key Terms—2D-to-3D-conversion, depth image-based rendering, hole-filling, image warping, optimization.

Overview of proposed Method: Our proposed approach for the pre-processing of depth maps to prevent holes during the view synthesis. We propose to generate the virtual videos of different viewpoints from a given video and associated depth maps at a viewpoint so that the viewers can realistically sense the depths. We also propose a spatio-temporal global optimization approach that formulates an energy function considering depth map, image structure, texture information and patch shift. Proposed method allows for a high-resolution representation of image regions where necessary and low-resolution where possible. We also propose simple novel energy functions with added benefits, such as effective prevention of warp inversions.

I. Introduction:

Technical commitment that was required for the presentation of 3D video was significant, which limited its benefits to a circle of professionals and burdened its general acceptance by a broader audience. However, as a result of a series of technical advances over the past years, 3D displaying devices are becoming increasingly popular in consumer products, for example in television sets, home projectors, or even hand-held devices like smartphones. This development has lead to a growing demand for 3D content. Despite the diversity of technologies for the generation of 3D video for stereoscopic or multi-view displays, the majority of them relies on dense depth maps. As such, the actual 3D image can be created directly on the device. The processes can be split into two steps: (1) depth estimation and (2) view synthesis. In the first step,

the depth structure of the corresponding monocular image is reconstructed, either automatically or with the interaction of an expert. The first step usually implies the generation of a depth map or a 3D model for each frame in a video sequence. The generation of depth maps can be roughly classified into semi-automatic (i.e. techniques that require user-interaction to a certain degree) and fully-automatic techniques [1]. Semiautomatic techniques are very time-consuming, and, thus, cost-intensive, since objects have to be segmented (rotoscoping) [2]–[5] and the depth of each object has to be modeled (depth grading). In [6]–[8] 2D images are converted to 3D images by matching parts of 2D images to a repository of known stereo images, followed by fusing the known depth structures of the matched stereo images to create depth structures for the 2D images.

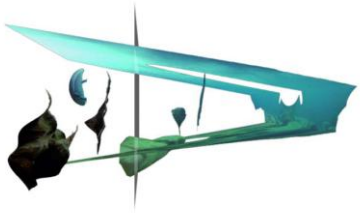
A. Depth Image-Based Rendering**:**

The synthesis algorithm used in the proposed application, depth image-based rendering (DIBR) [11], exploits dense depth maps in order to synthesize a virtual view. It assumes an 1D parallel arrangement of the virtual camera relative to the first camera and identical intrinsic parameters of both cameras. This permits the synthesis of the second view by horizontally shifting pixels according to their corresponding disparity values [32]. In contrast to the binocular disparity, which is considered to be an angle, the disparity used in stereo conversion is measured in pixels. Additionally, only the horizontal disparity is used for view synthesis, as vertical disparity has been found to cause nausea and fatigue [33]. Consider a metrical depth map **Z**, i.e. a map where each pixel holds the Euclidean distance from the projection center of the camera to the corresponding object point of that pixel in the scene. Next, the map **Z** is converted into a disparity map $\tilde{\mathbf{D}}$, such that the value of each pixel i of the depth map **Z**, denoted as $\mathbf{Z}i$, is inversely proportional to the corresponding disparity value $\tilde{\mathbf{D}}i$, i.e. $\tilde{\mathbf{D}}i \propto 1/\mathbf{Z}i$. Since the size of 3D displays is not constrained, the relative difference of disparity values of a scene are more important than the absolute disparity values. This is reflected by normalization of disparity map $\tilde{\mathbf{D}}$ to a unit less gray-value map **C** containing values in the range [0,*C*max] where large values correspond to objects close to the camera and small values correspond to more distant objects. Depending on the screen size and distance of the audience to the display, the maximum and minimum disparity values of a scene are controlled by two scalars, far shift d_f and near shift d_n, respectively, and are specified in pixels. The *screen specific* disparity map **D** can now be expressed as $\mathbf{D}i = d_n - d_f \, C\max \, \mathbf{C}i + d_f$, (1) where the value of element $\mathbf{D}i$ denotes the extent to which a pixel i appears in front of or behind the plane of a 3D display. Please note that $d_n < d_f$ has to be valid in order to prevent background objects from appearing in front of foreground objects. For the remainder of this paper the unit-less gray value map **C** will be referred to as depth map. The effect of varying values.

(a)

(b)

(c)



(d)

Fig. 1. Example of a scene with varying depth budget, (a) a 2D input image, (b) depth image-based 3D model of the scene viewed from above, (c) the

same model as in (b) viewed from the side, and (d) again the same scene from the
side but with a larger depth budget. gray plane in (b) denotes the zero parallax plane. This plane is visible as vertical gray line in (c) and (d).

B. Image Warping:

A $m \times n$ image is represented as a directed graph or mesh grid $G = (V, E, Q)$, where $V = [\mathbf{v}0, \ldots, \mathbf{v}N]$ is a list of vertices $\mathbf{v}i \in$ R2, the set of directed edges is denoted as E, and quads or faces as Q. In this setting each pixel in the image is represented by a vertex in G. However, this grid size can be chosen arbitrarily from sub- to uperpixel resolution, depending on the application and computational feasibility. The image is deformed according to a set of warping parameters _ = [**p**0, . . . , **p***N*], where **p***i* ∈ R2, to yield a new set of vertices *V*_ = [**v**_ 0, . . . , **v**_ *N*] such that **v**_ *I* = **v***i* + **p***i* .

C. Regular grid based Approach:

The approach for hole-filling using image warping, which is portrayed in this section, is partially based on previous work of [28]. However, in contrast to their method, our approach is not required to approximate a disparity map, since it is readily available. Our algorithm, as schematically outlined in Fig. 4(b), works as follows. Initially, the pipeline of DIBR is followed. Using the depth-to-disparity mapping defined in (1) and a specified depth budget *b*, the depth map **C** is converted to a disparity map **D**. Then, depending on the selected source camera *c*, two warping maps ˜_ *L* and ˜_ *R* are computed from **D** as defined by (2). An analysis of the warping maps yields a pair of binary masks, **H***L* and **H***R*, which indicate where the texture output images, **I***L* out and **I***R* out, will have holes. This information is stored in the coordinate frame of the input image **I**in and can be directly incorporated into an optimization problem, which will alter the initial warping maps to yield a new pair of warping maps, _*L* and _*R*. These updated maps are applied in the remaining DIBR pipeline to produce a render output that is visually very similar to that of the initial warping maps, but does not contain holes.

(a) input image **I**in

(a) depth map **C**

(c) output image **I***L*out

Definitions and Notation:

Following the notation introduced in Section II-B, an image **I** is represented as a regular, 4-connected grid with directed edges where each image pixel corresponds to a vertex in the grid. The objective is to find a pair of warping maps _*L* and _*R*, such that no holes occur in the textures of the left and right stereo view **I***L* out and **I***R* out while at the same time the render output is still visually similar to the output that is obtained when rendering with the initial warping maps ˜_ *L* and ˜_ *R* while also preserving the disparities **D**. The warping maps modify only the *x*-coordinates of pixels, since a parallel aligned camera arrangement is assumed. As in [28], to find these warps, an energy minimization problem is defined.

The final energy *E* function will be composed linearly of three energy terms. The first term, *Es* , preserves the visual appearance by creating a bonding energy between neighbouring pixels. The second term, *Ed* , will force the stereo disparity to be as close as possible to the disparity map **D**, and the third term, *Ea*, makes sure that the absolute warping values remain similar to those defined in the initial warping maps. Finally, these energies are combined with the information about holes gained from the mask **H***L* and **H***R* to find the warp that avoids holes. Additionally, a saliency map is used to indicate the importance of pixels. Basically, it is the idea to hide the image deformation in those regions of an image that are not that important while leaving important (salient) regions intact. Generally, the closer an object is to the user, the more salient it is. This heuristic is used in a simple mapping to transform the depth map **C** into a saliency map **S**, which is defined as **S***i* = *s*max – *s*min *C*max **C***i* + *s*min (5) where *C*max is the maximum allowed depth and the scalars *s*min and *s*max denote the minimum and maximum allowed saliency, respectively. In our setting we use *s*min = 0.1 and *s*max = 1. Choosing a positive, non-zero value for *s*min prevents image regions to be transformed arbitrarily, which could happen in case they had no importance at all.

D. Hole-Filling:

Having identified those regions, in which holes will occur, the next step will be to convert the holes masks into a series Fig. 5. Depth discontinuities mask **G** is the result of an edge detection applied to disparity map **D** and indicates depth changes in the scene. In conjunction with the holes masks **H***L* and **H***R*, these mask are used to identify those edges between neighbouring pixels that may be deformed. The proposed adaptive warping algorithm exploits these masks and the depth structure in order to generate a quadtree grid over the image, which is depicted in (d). The masks depicted here are associated to the input image in Fig. 3. of spans, in a fashion similar to run-length encoding (RLE). A span is defined as a sequence of consecutive hole pixels in a row, for which the location of the first pixel before and after the hole is stored as starting *a* and ending point *b*. Additionally, either the starting or the ending point will be marked as fixed, depending on which is closer to the viewer, i.e. *φ(a, b)* = _*a*, if **C***(a)* > **C***(b) b*, otherwise.(7) Fixing one side of a span is necessary, as the other side will be pulled towards it, which closes the hole. The pulling of pixels is achieved by assigning the warping value of the fixed span point to all the other pixels in that span. Let *(a, b)* denote a hole span and assume for the moment that *a* is marked as fixed, we then define the hole-filling boundary constraint for the current span as _*i* = ˜_*a*, (8) where $a \leq i \leq b$. This means, in other words, that the warping values of pixels inside a hole span are fixed to the warping value of *a*, which guarantees that these pixels will maintain a relative pair-wise distance of one pixel after warping, which will avoid any holes. The error that is introduced by this constraint will be hidden by spreading it over neighboring non-hole pixels as will be explained later. The case, where *b* is fixed, can be constructed analogously. Hole spans of the left holes mask **H***L* define only boundary constraints for the left warping map _*L*, and spans in **H***R* only for the right warping map _*R*. These constraints will transform exactly those pixels that are marked as holes in **H***L* and **H***R*. However, shifting these marked pixels will not automatically pull along their neighboring pixels outside of the span, which are not marked as holes explicitly. As a result new holes may occur along the sides of hole spans. The next sections will address this problem by introducing a kind of cohesive attraction between pixels. This attraction is expressed as energy terms and ensures that the background is stretched to the foreground in such a way that new holes in the texture are avoided while also preserving disparity.

E. Preserving Smoothness:

Similar to [28], a smoothness preserving energy term *Es* is used. This term serves to keep the deformation of the regular grid to a minimum and directs errors introduced by the deformation, which are inevitable, to less salient regions in the image by penalizing local distortion everywhere except at overlaps. Overlaps can be regarded as the opposite case of holes. They occur when foreground objects are shifted less than background objects or even into colliding directions Please note that the information on overlaps is contained in a mask **G**, which stores depth discontinuities. The depth discontinuities are extracted by applying a standard edge detection algorithm on the disparity map **D**. Overlaps can then be conveniently identified in **G** as those pixels with non-zero values which are not marked in holes masks **H**. An example of a depth discontinuity map is depicted in Fig. 5. The edge between pixels *i* and its neighboring pixel *j* is allowed to be deformed, if it is an overlap and if it is located on a depth discontinuity.

II. RELATED WORK

More and more 3D display applications can be found in the high-tech products, including 3D LCD/LED

displays, 3D laptops, 3D cameras, mobile devices and home video/games, etc. The suitable file formats to support these modern 3D devices have become one of the most important issues. The video-plus-depth format is commonly used in the 3DTV community. It consists of the color intensity and the associated per-pixel depth map at one view. Based on this format, the DIBR (Depth-Image-Based Rendering) system [4] produced virtual views based on the three steps: preprocessing of depth image, image warping and hole filling. However, the major problem in DIBR is how to fill the holes caused by the disocclusion regions in which the occluded pixels in the source view may become visible in the virtual views. Under the video-plus-depth format, some research works focused on the preprocessing of the depth image [5] to reduce the disocclusion regions. Others developed hole filling methods based on image inpainting techniques [6] to fill in the disocclusion regions. For preprocessing of the depth image, the common approach is to apply the smoothness filters, (e.g. Gaussian filter and average filter) to the depth image. After image warping with the smoothed depth image, the disocclusion regions may be split into several small holes. Then the color interpolation can be used to fill in the small hole regions. Zhang et al. [5] extended the idea of the depth preprocessing for the hole filling from the symmetric smoothing filter to the asymmetric smoothing in order to reduce the geometric distortion. For image inpainting, Oh et al. [6] proposed a hole filling method by using depth-based inpainting. This method is designed by combining the depth-based hole filling and the image inpainting technique. Because hole filling is the major problem in depth-imagebased rendering. Image inpainting is a technique widely utilized to recover the disocclusion regions. The objective is to fill the unknown regions in a plausible way. Recent exemplarbased approaches are commonly used in image inpainting. In [7], a fast algorithm was presented to propagate texture and structure in a small patch. The success of structure propagation was dependent on the order in which the filling proceeds. The confidence value in the synthesized pixel values was propagated in a manner similar to the propagation of information in inpainting. Contrary to the greedy methods, some approaches formulate the image completion as discrete global optimization problems [8] [9] [10]. In [8], image completion was automatically solved using an efficient BP algorithm. However, it did not consider structure information and thus the results may contain structure inconsistency. In [9], the image was completed with manually added structure information. Huang et al. [10] improved the hole filling method in [8] by adding the structure information into the global optimization formulation and solved the optimization problem with a two-step BP. In their method, only a single image was considered for the completion. Then, Liu et al. extended the method to the video completion [11] by adding the motion information to keep spatial and temporal coherence.

III. ADAPTIVE GRID BASED APPROACH:

The proposed algorithm is motivated by the observation that most depth maps **C** and their corresponding disparity maps **D** are very smooth for the most part and depth changes happen gradually. The only exceptions are found along depth discontinuities. Under the assumption that also the final solution, $\_L$ and $\_R$, will be comparably smooth in these regions, there is great potential to improve efficiency and reduce redundancy. Instead of representing each pixel as an individual variable in the optimization problem defined in (15), the idea is to combine pixels with similar disparity into larger regions of support and, thus, represent these pixels using fewer variables. This scheme allows for a high-resolution representation of image regions where necessary and low-resolution where possible. In theory, in order to reduce the complexity, the original square system of linear equations $\mathbf{Ax} = \mathbf{b}$ with n unknowns is transformed linearly into a problem of lower dimensionality. Following [31], this is achieved by substituting the solution Vector $\mathbf{x} = \mathbf{My}$ (18) so that the system of equations can now be rewritten to $\mathbf{AMy} = \mathbf{b}$ (19) Fig. 6. An excerpt an image grid along a hole (red pixels) superimposed by a quadtree structure. The pixel grid of the image is depicted by the white squares with black borders. The four corners of each quad are coinciding with pixel centers. While the full linear system represents each pixel in the image by one unknown, the reduced linear system defines variables only for those pixels that are corners of quad tree leafs (green/blue dots). In order to avoid artefacts, t-junctions corners t (blue dots) are forced to lie on the line that is defined its neighboring corners a and b. The quad corners and the green lines correspond to the set of vertices V and the set of edges E of the directed quad graph G, respectively. where $\mathbf{y} \in \mathrm{R}m$ and $\mathbf{M} \in \mathrm{R}n \times m$ denote the reduced solution vector and the transformation matrix, respectively. By squaring the problem now becomes $\mathbf{M}T\mathbf{A}T\mathbf{AMy} = \mathbf{M}T\mathbf{A}T\,\mathbf{b}$, (20) which is a $m \times m$ rather than a $n \times n$ problem and, hence, reduces the complexity significantly for $m \; n$. After solving the reduced problem, the solution is transformed back to the initial size using (18). In practice, the matrix **M** that is used to reduce the complexity of the optimization problem does not have to be expressed explicitly. This is because the transformation that is

encoded in **M** can be understood as a linear interpolation of the data. And this, in turn, can be approximated by a quadtree [36]. Using bilinear (or cubic) interpolation, it is possible to reconstruct the values of entries enclosed by the four corners of a leaf quad. The construction of this tree is explained in the next section.

Error Minimization:

The optimization problem (8) can now be updated to argmin $\_L$, $\_R$ $E(\_L$,$\_R)$ subject to hole-filling constraints (8), t-junction constraints (24) (25) where the energy is defined as the weighted sum of energy terms in (16) but now using the updated energies (21)–(23). This problem is minimized identically to the regular gridbased approach, only that now a $2M \times 2M$ instead of an $2N \times 2N$ problem has to be solved. Since $M \ll N$, solving the reduced optimization problem is far more efficient. Fig. 7. The initial warping maps generated by the DIBR-algorithm, which produce holes during rendering, are depicted in the top row. The bottom row depicts the updated warping maps produced by the warping algorithm, which produce a visually similar output but less the holes. The solution vector is then mapped to the final warping maps $\_L$ and $\_R$ using bilinear interpolation to fill the values of the regions inside the quadtree nodes. Finally, the rendering algorithm from section II-A generates the stereo frame using the new warping maps. The entire algorithm is depicted schematically in Fig. 4(b). An example for the output of the adaptive warping algorithm is depicted in Fig. 7. Negative shift values in the warping maps are coloured in green, positive values are shown in red and the brightness of each color indicates the magnitude of the shift. The top row depicts the initial warp maps that were produced by the DIBR-algorithm for the images shown in Fig. 3. The bottom row depicts the updated warp maps that are produced by the warping algorithm. The updated warp maps are similar to the initial warp maps except at region that would create holes.

Visual Quality:

While the previous sections have demonstrated the complexity reduction of the adaptive approach with respect to the regular approach, this section evaluates the impact of the optimization on the visual quality. In general, multimedia quality can be evaluated either through subjective quality tests or based on objective quality metrics which try to predict the perceived quality. Since the output of the adaptive grid-based approach is designed to be as similar to the output of the regular grid-based approach as possible, the quality difference between the two approaches was assessed using objective quality metrics. Most of the proposed 3D quality metrics apply well-known 2D quality metrics to the individual views and depth map of the 3D content and combine the results according to some heuristic rules [40]. Since both approaches modify the warping maps in order to prevent holes, the most suitable Fig. 11. Comparison of hole-filling methods: (a) an input image (1920 × 1080 pixels), (b)–(f) synthesized left stereo frames. (b) DIBR without any hole filling. Holes are shown as green areas and measure approximately 30 pixels in width. In (c) the depth map is blurred with a Gaussian filter until all holes in the DIBR result have vanished. In (d) the holes of the result in (b) have been filled using the in-painting algorithm from [39], (e) the DIBR result after pre-processing using the regular warping method. Finally, (f) depicts the conversion result of DIBR if the depth structure is pre-processed with the proposed warping algorithm, (g) depicts the blurred depth map used in (c), and (h) depicts the depth map with corresponding quad tree as an overlay. way for the evaluation is to compare the final results after the view synthesis. The structural similarity index (SSIM) [41] is adopted to compare the synthesized view from the adaptive approach with the one from the regular approach since it has been shown to provide a good correlation with the perceived quality [42]. Fig. 12 depicts a 100 bin histogram of the SSIM scores across a series of image sequences. The values range from 0.891 to 0.987 with an average score of 0.943 which means that the result of the adaptive approach is structurally very similar to the regular approach. Combined with the second view for 3D restitution the deviations are even less noticeable due to a perceptual effect called binocular suppression. This can be visually verified by the results shown in columns three and four of Fig. 8. The differences between the output of the proposed warping method versus alternative pre- and post-processing methods for hole-filling are depicted in Fig. 11. In comparison to warping the blurring of depth maps as suggested in [11] is a cheap operation in terms of computational cost. However, it creates visible artefacts along object edges where foreground objects sink into the background and vice versa. Hence, these artefacts are referred to as sinking edges. In-painting [39] on the other hand may lead to halo effects along depth discontinuities and object boundaries since holes are filled by continuing and expanding the texture along the boundary of holes. Warping on the other hand does not exhibit either behaviour, since the depth structure of foreground objects is preserved, which avoids sinking edges, and no additional image texture is created as in the case of in-painting.

IV. CONCLUSION:

In this paper we have described the problem of holes that occur in the process of depth image-based rendering, due to the lack of knowledge about the 3D structure and texture of a scene. We proposed an image warping algorithm based on the work of [28] that represents an image as a regular grid and alters a pair of warping maps such that the final render output will be hole-free. We also proposed a new algorithm based on an adaptive grid-structure, which facilitates an approximation of the first algorithm and increases the efficiency significantly. The effectiveness of the proposed method has been shown by a series of experiments which demonstrate that the proposed adaptive algorithm produces visually very similar results at a fraction of the computational cost that is required by the regular grid-based approach.

V. REFERENCES

A. Smolic, P. Kauff, S. Knorr, A. Hornung, M. Kunter, M. Müller, and M. Lang, "Three-dimensional video postproduction and processing," Proc. IEEE, vol. 99, no. 4, pp. 607–625, Apr. 2011.

C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': Interactive foreground extraction using iterated graph cuts," ACM Trans. Graph., vol. 23, no. 3, pp. 309–314, Aug. 2004.

J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen, "Interactive video cutout," ACM Trans. Graph., vol. 24, no. 3, pp. 585–594, Jul. 2005.

A. Myronenko and X. Song, "Global active contour-based image segmentation via probability alignment," in Proc. IEEE Conf. CVPR, Jun. 2009, pp. 2798–2804.

X. Bai, J. Wang, D. Simons, and G. Sapiro. (2009). "Video SnapCut: Robust video object cutout using localized classifiers," in ACM SIGGRAPH 2009 Papers [Online]. Available: http://doi.acm.org/10.1145/1576246.1531376

A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 824–840, May 2009.

Konrad, M. Wang, and P. Ishwar, "2D-to-3D image conversion by learning depth from examples," in Proc. IEEE Comput. Soc. Conf. CVPRW, Jun. 2012, pp. 16–22.

B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in Proc. IEEE Conf. CVPR, Jun. 2010, pp. 1253–1260.

P. Didyk, T. Ritschel, E. Eisemann, K. Myszkowski, and H. Seidel, "Adaptive image-space stereo view synthesis," in *Proc. Vis., Model., Visualizat. Workshop*, 2010, pp. 299–306.

Agarwala, "Efficient gradient-domain compositing using quadtrees," *ACM Trans. Graph.*, vol. 26, no. 3, p. 94, Jul. 2007.

D. Tian, P. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," *Proc. SPIE*, vol. 7443, pp. 74430T-1–74430T-11, Oct. 2009.

J. Choi, D. Kim, S. Choi, and K. Sohn, "Visual fatigue modeling and analysis for stereoscopic video," *Opt. Eng.*, vol. 51, no. 1, pp. 017206- 1–017206,-12, Feb. 2012.

S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proc. 23rd IEEE Conf. CVPR*, Oct. 2010, pp. 2376–2383.

M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu, "Global contrast based salient region detection," in *Proc. 24th IEEE Conf. CVPR*, Jun. 2011, pp. 409–416.

H. Samet, "The quadtree and related hierarchical data structures," *ACM Comput. Surv. CSUR*, vol. 16, no. 2, pp. 187–260, Jun. 1984.

C. Dyer, "The space efficiency of quadtrees," *Comput. Graph. Image Process.*, vol. 19, no. 4, pp. 335–348, Aug. 1982.

D. Moore, "The cost of balancing generalized quadtrees," in *Proc. 3rd ACM Symp. Solid Modeling Appl.*, 1995, pp. 305–312.

A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, 2004.

Q. Huynh-Thu, P. Le Callet, and M. Barkowsky, "Video quality assessment: From 2D to 3D—Challenges and future trends," in *Proc. 17th IEEE ICIP*, Sep. 2010, pp. 4025–4028.

Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

P. Hanhart and T. Ebrahimi, "Quality assessment of a stereo pair formed from decoded and synthesized views using objective metrics," in *Proc.*