

Novel View-Synthesis from Multiple Sources for Conversion to 3DS

Francesco Malapelle¹, Andrea Fusiello¹,
Beatrice Rossi², and Pasqualina Fragneto²

¹ Università di Udine - DIEGM, Via delle Scienze 208, Udine, Italy

² STMicroelectronics - AST Lab, Via Camillo Olivetti 2, Agrate Brianza (MB), Italy

Abstract. In this paper we confront the problem of uncalibrated view synthesis, i.e. rendering novel images from two, or more images without any knowledge on camera parameters. The method builds on the computation of planar parallax and focuses on the application of converting a monocular image sequence to a 3D stereo video, a problem that requires the positioning of the virtual camera outside the actual motion trajectory. The paper addresses both geometric and practical issues related to the rendering. We validate our method by showing both quantitative and qualitative results.

Keywords: image-based rendering, view-synthesis, 3DS video

1 Introduction

View Synthesis (VS) or *Image Based Rendering* (IBR) is the generation of novel images as if they were captured from virtual viewpoints, starting from a set of actual images or frames. Applications include the generation of a 3D stereo (3DS) video from a monocular one [24, 14, 3, 26] and the upsampling of video sequences in order to achieve slow-motion effects (e.g., [4, 17])

The rendering of virtual images requires some geometry information, either explicit (depth) or implicit (depth-proxies), and suitable warping functions. When cameras are *calibrated*, i.e. when both internal and external parameters are available, given the depth of an image point, it is straightforward to compute the position of the point in virtual image from any viewpoint. Techniques based on this paradigm, known as *Depth Image Based Rendering* (DIBR), have been extensively studied and several solutions are available in the literature ([27] and references therein).

When dealing with the 3DS conversion problem, calibration data is hardly available. Despite this, many works addressing this application ([24, 14, 3, 26]) assume some knowledge on the camera parameters, and fall within the DIBR family described above. The *uncalibrated* view synthesis (UVS) is less explored and more challenging for several reasons.

First of all, depth cannot be used in uncalibrated situations, and suitable depth-proxies must be defined, together with proper warping functions based on fundamental matrices [16], trilinear tensors [2], or plane-parallax representation [13, 22].

Second, specifying the external *orientation* (position and attitude) of virtual views is unnatural, since they are embedded in a projective frame, linked to the Euclidean one by an *unknown* projective transformation. Only few works address this problem. In [7] an automatic method based on the planar parallax as a geometry proxy is presented: given two or more reference images, the possible uncalibrated orientations describe a 1-parametric trajectories obtained interpolating or extrapolating the relative motion among reference images. This approach is expanded in [10] by extending to 3-parametric trajectories, thus allowing additional positions along and orthogonally the line of sight, and in [6] by defining a 1-parametric rectified trajectory that, when deroctified, is compatible with the one in [7], and is more resilient to errors induced by poor epipolar geometry estimation. In the upsampling of video sequences the virtual views are always very close to the reference view, hence simple interpolation along motion vectors is widely used.

Finally – but this issue is shared with DIBR technique, several sub-problems have to be addressed when applying warping functions: *folding*, which occurs when two or more pixels in the reference image are warped to the same pixel in the virtual image, *holes*, which may be caused either by occlusions or by missing geometric information, and *resampling*, due to the discrete nature of digital images.

Contributions. In this paper we present a fully automatic UVS pipeline which addresses most of the critical problems arising in uncalibrated scenarios. The method takes inspiration from the pipeline presented in [8], but instead of transferring points directly from the reference image to the novel one (forward), we use a *backward mapping* strategy which yields finer results. Moreover, we combine information coming from *multiple reference images*, blending several parallax maps into one. At last, we propose a simple and suitable method to fill holes in the final virtual image and cope with resampling artefacts.

2 Background

In this section we cover some of the background notions that are needed to understand the method. In Subsection 2.1 we introduce planar parallax as an uncalibrated proxy for the depth. In Subsection 2.2 we describe the theory of the uncalibrated camera orientation specification, based on [7].

2.1 Planar Parallax

Planar parallax represents the displacement in the apparent position of objects imaged from different points of view with respect to a reference plane, and can be computed from stereo correspondences.

Let us consider a 3D point \mathbf{M} belonging to some plane Π and its projection $(\mathbf{m}_r, \mathbf{m}_i)$ onto the image planes I_r and I_i respectively. There exists a non-singular linear transformation, or homography, that maps \mathbf{m}_r onto \mathbf{m}_i , i.e.

$$\mathbf{m}_i \simeq H_{\Pi} \mathbf{m}_r \quad (1)$$

where H_{Π} is the homography induced by plane Π and \simeq means equality up to a scale factor. For 3D points \mathbf{M} not belonging to plane Π , the following more general relation holds:

$$\mathbf{m}_i \simeq H_{\Pi} \mathbf{m}_r + \mathbf{e}_i \gamma \quad (2)$$

where \mathbf{e}_i is the epipole in I_i and γ is the *planar parallax* (or, simply, *parallax* if the context is clear) of \mathbf{M} , which can be interpreted as the displacement between the point $H_{\Pi} \mathbf{m}_r$ mapped via the homography H_{Π} and its actual corresponding point \mathbf{m}_i . With a little abuse of notation, the term parallax is also used to denote the magnitude γ of this displacement (the direction is always towards the epipole \mathbf{e}_i).

Note that, when two image planes are coplanar (i.e., motion is along X axis, up to coordinates change) and the reference plane is at infinity, then H_{Π} is the identity and the epipole is $\mathbf{e}_i = [1 \ 0 \ 0]^{\top}$, thus parallax γ in Eq. (2) results to be proportional to binocular disparity.

Given point correspondences and a plane homography H_{Π} , parallax values can be obtained for each pixel of the reference frame by solving for γ in Eq. (2):

$$\frac{1}{\gamma} = \frac{(\mathbf{e}_i \times \mathbf{m}_i)^T (\mathbf{m}_i \times H_{\Pi} \mathbf{m}_r)}{\|\mathbf{m}_i \times H_{\Pi} \mathbf{m}_r\|^2}. \quad (3)$$

It can be demonstrated that i) γ is proportional to the inverse of the depth of points, and ii) γ depends only on the orientation of I_r and the plane Π , and not on the orientation of I_i . For this reason we will call I_r the *reference* image and I_i the *auxiliary* image. Hence, parallax maps computed with multiple pairs (I_r, I_i) - where i varies - are commensurate, provided the reference plane is fixed.

Please note that this is not true for disparity. Since disparity is proportional to the reciprocal of the depth and the depth is defined with respect to the focal plane, there must be a common focal plane in order for disparities to be commensurate. This can always be achieved for $N \leq 3$ cameras by rectification (rotating the focal planes around the optical centres until they coincide with the plane defined by the three centres), but cannot be guaranteed for more cameras, unless camera centres lies on a plane (see [18] for more details).

In summary, the parallax can be seen as a useful generalization of the depth in the uncalibrated case, and as a generalization of disparity to unrestricted camera configurations. In the case where camera calibration is unavailable and the camera undergoes a general motion, *planar parallax* [21], can be profitably employed as a depth-proxy.

2.2 Uncalibrated motion description

If the parallax γ is known, since it does not depend on the auxiliary image, it is possible to exploit Eq. (2) in order to map points from the reference image I_r onto a virtual image I_v . Specifically, one must substitute H_{Π} with the homography of the plane Π between I_r and I_v , namely H_{Π}^{rv} , and \mathbf{e}_i with the epipole in I_v , namely \mathbf{e}_v .

In [8] it is described how to specify the so-called *uncalibrated motions*

$$D_{rv} = \begin{bmatrix} H_{\Pi}^{rv} & \mathbf{e}_v \\ 0 & 1 \end{bmatrix} \quad (4)$$

when the reference plane Π is the plane at infinity (i.e. $H_{\Pi} = H_{\infty}$) and D_{rv} is the orientation of the virtual camera.

Let us consider the uncalibrated motion of the reference image wrt the auxiliary one:

$$D_{ri} = \begin{bmatrix} H_{\Pi}^{ri} & \mathbf{e}_i \\ 0 & 1 \end{bmatrix}. \quad (5)$$

It can be noticed that D_{ri} resembles a rigid transformation matrix

$$G_{ri} = \begin{bmatrix} R_{ri} & \mathbf{t}_{ri} \\ 0 & 1 \end{bmatrix} \in SE(3, \mathbb{R}). \quad (6)$$

Specifically D_{ri} and G_{ri} are related by $D_{ri} = \hat{K}G_{ri}\hat{K}^{-1}$ with K being the unknown matrix of the internal parameters of the camera and $\hat{K} = \begin{bmatrix} K & 0 \\ 0 & 1 \end{bmatrix}$. Thanks to this identification (actually it is an isomorphism), and since $SE(3, \mathbb{R})$ is a Lie group, it is possible to continuously parametrize the uncalibrated motion of the virtual camera as:

$$D_{rv} := D_{ri}^t := \exp(t \log(D_{ri})) \quad t \in \mathbb{R}. \quad (7)$$

Varying the value of t , we obtain a 1-parameter family orientations which naturally interpolates/extrapolates the orientations of the of reference and the auxiliary cameras along a geodesic path.

The infinite plane homography H_{Π}^{rv} , along with the epipole \mathbf{e}_v needed in Eq. (2) for the view synthesis can be extracted from D_{rv} according to Eq. (4).

3 Proposed method

In this section we describe the steps of our method. The input is a set of reference images I_r and a set (not necessarily disjoint) of auxiliary images I_i . For each of the reference images one or more parallax maps are computed with the support of auxiliary images I_i (the ones with the highest overlap with I_r). For example, in the 3DS application there is a single reference image and typically two auxiliary images (previous and subsequent frame). In the upsampling case there are two reference images (the two frames that are to be interpolated) which – in turn – play the role of the auxiliary ones (see Fig. 1). These parallax maps are transferred (forward) to the virtual image I_v and merged together. The resulting map is then used to synthesize I_v by (backward) mapping to the "right" pixel in the "right" source image.

Please note that the use of parallax instead of disparity is crucial to allow the fusion of multiple parallax maps, as discussed in Sec. 2.

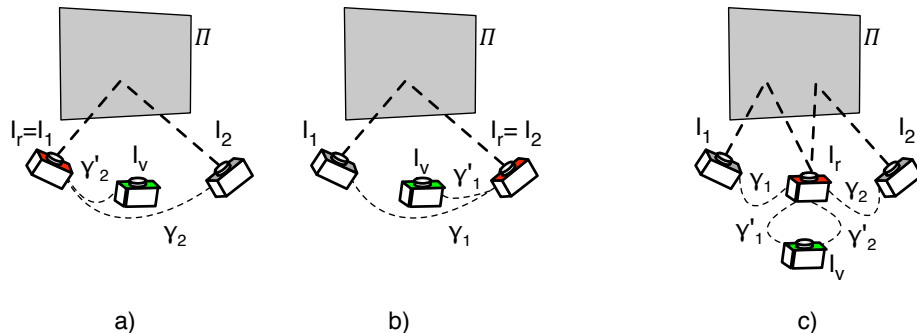


Fig. 1. a) and b) represent the interpolation scheme, with I_1 and I_2 exchanging the role of reference and auxiliary image. c) is the 3DS scenario where there is one reference frame and the neighbours are the auxiliary images. In both cases the the parallax map for I_v results from the fusion of γ'_1 and γ'_2 .

3.1 Virtual camera orientation

In the upsampling application the orientation of the virtual camera interpolates between two actual ones, hence it can be specified by computing D_{rv} using Eq. (7) and selecting $t \in [0, 1]$.

In the 3DS conversion application, on the contrary, the virtual camera position is alongside the actual one, i.e., outside its trajectory. Assuming that the video has been shot with zero roll angle, i.e., the image rows are parallel to the horizon, H_{Π}^{rv} and \mathbf{e}_v can be specified as follows. Since there is no rotation between the reference image and the virtual one (images are coplanar), the infinite plane homography H_{Π}^{rv} is the identity matrix. As for the epipole, $\mathbf{e}_v = [t \ 0 \ 0]^T$ with $t \in \mathbb{R}^+$, since the virtual viewpoint is displaced horizontally. Thus the orientation of the virtual camera can be computed as

$$D_{rv} = \begin{bmatrix} 1 & 0 & 0 & t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, t \in \mathbb{R}^+. \quad (8)$$

3.2 Stereo processing

The purpose of this block of the algorithm is to compute the parallax value γ for each pixel of the reference image I_r , with the support of an auxiliary image to constitute a stereo pair.

First the image pair is rectified. Since the internal parameters are unknown, we use the uncalibrated procedure described in [9], which relies on sparse correspondences. To this end, first SIFT features are extracted in both images and descriptors are matched, and then a RANSAC estimation of the fundamental matrix is performed in order to discard outlier matches. Uncalibrated rectification seeks the homographies T_2 and T_1 that make the original matching points satisfy the epipolar geometry of a *normal* stereo pair. Stereo matching produces two disparity maps, one referred to I_r and the other to I_i .

At this point we perform some post-processing on the disparity maps. First we use a simple hole-filling technique as the one suggested in [20]. Afterwards, we use anisotropic diffusion [19] to smooth out the maps without compromising the edges. At last we run a left-right consistency check to gather precious occlusion (i.e., visibility) information. The disparity maps are used to obtain a set of dense correspondences that are then deroctified using the inverse rectifying homographies.

Ultimately, parallax values are computed using Eq. (3) for each pixel: the dense set of correspondences $\mathbf{m}_r^k \leftrightarrow \mathbf{m}_i^k$ on the pair of images (I_r, I_i) is known from the stereo matching step; the collineation is $H_{\Pi}^{ri} = T_2^{-1}T_1$ and epipole \mathbf{e}_i is estimated from epipolar geometry. As a by-product of the rectification method [9], H_{Π} approximates the homography of the plane at infinity.

3.3 Forward mapping of parallax maps

Starting from a parallax map for the reference image I_r , we want to obtain a map referred to the virtual I_v instead.

First we generate a set of corresponding points between I_r and I_v by instantiating Eq. (2) as

$$\mathbf{m}_v \simeq H_{\Pi}^{rv} \mathbf{m}_r + \mathbf{e}_v \gamma \quad (9)$$

where H_{Π}^{rv} is the infinite plane homography between the reference image I_r and the virtual one I_v and \mathbf{e}_v is the epipole of I_v . Quantities H_{Π}^{rv} and e_v are specified through the parameter t in Eq. (7) which encodes the inter-ocular separation.

This process is a *forward mapping*: points in the original image are mapped forward, to the virtual image. Once we obtain the set of corresponding points among the reference image and the novel image we can compute parallax values with respect to our novel image using Eq. (3).

As it is well known, forward mapping raises some problems, for it is necessary to deal with small holes in the destination images due to the non-surjectivity of the map and to cope with its non-injectivity, causing the *folding* effect.

Probabilistic splatting In order to deal with the non-surjectivity issue, we developed a randomized technique that account for the quantization inherent to the forward mapping.

First, we generate noise in the form of random values drawn from the standard uniform distribution on the open interval $(-0.5, 0.5)$ (the amplitude of the interval is chosen to be equal to the maximum error introduced by the coordinates rounding). The noise is added to the non-integer coordinates \mathbf{m}_v in Eq. (9) which are then rounded to their closest integer value. The procedure is repeated for n times (we choose $n = 100$) and all the perturbed parallax maps are merged into the final one by averaging them. This approach has two main advantages: i) as n increases, the process will tend to approximate a proper linear interpolation between the neighbouring pixels, based on the distance from the integer values (i.e. the decimal parts of the coordinates) ii) this procedure fills holes in the map, since pixels with undefined value are likely to be filled with the value of the neighbouring valid ones.

Folding Folding occurs when different source pixels are mapped to the same destination pixel. This phenomenon is due to the modification of the viewpoint, when two points that were visible in the original image fall along the same line of sight in the new image. As most approaches in literature, we deal with this problem by selecting the pixel with the greater disparity which, by definition will be occluding the one with a smaller disparity value.

3.4 Merging of parallax maps

At this point we have a collection of independent parallax maps for I_v and the purpose of this step is to merge these maps into one, in order to reduce noise and fill holes.

Observe that even though these maps are commensurable, they differ by a global scale factor s . However, due to noise and outliers, the factor will not be unique for the entire image, thus we estimate it from the distribution of the pixel-wise ratios in a robust way using the Median Absolute Deviation and the x84 rejection rule [11].

Once the maps are brought to the same scale we merge them into a final one by keeping the highest parallax value in each pixel, where holes have conventionally assigned $-\infty$.

This fused parallax map is accompanied by a *source map* that records for each element of the parallax map, which of the reference images it originates from (in case of multiple I_r). Together they define a mapping from I_v to the reference images that will be exploited in the actual rendering of I_v , described in the next section.

3.5 Backward mapping of colour

In the final stage of the method, the pixel grid of the virtual image is used as a reference to determine corresponding point in the reference images.

This process is a *backward mapping*, since points in the virtual image are mapped (backward), to points in the reference images to get a colour assigned. Again, we rely on Eq. (2) and we rewrite it as

$$\mathbf{m}_r \simeq H_{II}^{vr} \mathbf{m}_v + \mathbf{e}_r \gamma \quad (10)$$

Where H_{II}^{vr} and \mathbf{e}_r are obtained according to Eq. (4), but this time from matrix D_{rv}^{-1} , where D_{rv} is the one defined in Eq. (7).

The formula is applied pixels-wise using as the reference image the one specified in the source map. Bilinear interpolation is used to assign values to non-integer coordinates.

Occlusion filling. There can be points that are visible in the novel image, but for which a parallax value could not be computed, because they are not visible in the reference images or because of failure of the stereo matching. Such holes can only be filled heuristically. In our method, we build a binary map that

estimates local foreground/background segmentation: on the disparity map, for each unassigned pixel we compute the variance of its neighbourhood. A high variance indicates the presence of multiple depth layers, thus it is likely that an object in the foreground is occluding the background and the pixel is marked as background. Otherwise it is marked as foreground. This procedure is based on the idea presented in greater details in [20]. Once the binary map is built, we use it to fill holes on the virtual view, using the average colour of the neighbours that fall within the same class.

4 Experiments

We report two sets of experiments that validate our approach. The first set focuses on the forward vs backward mapping issue, and shows quantitative results by comparing our rendered images against ground truth images. The second set of experiments shows the visual results of the rendering of a virtual frames in a 3DS conversion scenario, where the position of the virtual camera is set alongside the actual one.

First we validate the choice of the backward mapping (BWM) approach – described in Sec. 3.5 and previous ones – against the forward mapping (FWM). FWM, used e.g. in [8], is the fusion of two virtual images obtained from the reference images. Both approaches are evaluated before the hole filling post-processing step.



Fig. 2. Parallax maps obtained from the stereo matching.

In order to factor out the inaccuracies of the stereo matching we used the Middlebury 2006 dataset ([12]) which provides ground truth disparity maps. Each sequence of the dataset is composed by seven frames, we used the second and the sixth frame as the reference pair to synthesize the middle frame, which corresponds to the fourth frame. We then compared the ground truth image with the virtual one and obtained the results reported in Tab. 1. As figures of merit we considered the structure similarity index (SSIM) [23], the signal-to-noise ration (SNR) and the absolute error rate (ABS) where pixels that differ from the true value for more than 1 pixel and unassigned ones are counted as errors. The result is that BWM consistently outperforms FWM showing its better ability to recreate the virtual image.

Finally, we report some qualitative results of the 3DS conversion in Fig. 3. The images are taken from [25] and from [12]. Stereo matching have been carried out with the implementation of [15] available in OpenCV; maps are reported in

Table 1. Comparison of performances of our BWM approach against the FWM employed in [8]. SSIM: and SNR the higher the better. ABS: the lower the better. See text for further explanations.

Sequence	BWM			FWM		
	SSIM	SNR	ABS	SSIM	SNR	ABS
Aloe	0.77	-33.59	33.74	0.72	-34.15	37.25
Baby1	0.89	-30.48	33.83	0.87	-30.80	34.79
Baby2	0.90	-26.91	31.67	0.88	-27.04	33.75
Baby3	0.88	-25.80	21.73	0.86	-26.45	23.35
Bowling1	0.88	-31.80	46.56	0.87	-31.90	46.99
Bowling2	0.87	-30.81	37.62	0.85	-31.23	39.04
Cloth1	0.96	-20.94	39.59	0.93	-19.77	40.74
Cloth2	0.91	-27.83	42.95	0.86	-28.75	44.66
Cloth3	0.91	-26.37	37.21	0.87	-27.20	39.08
Cloth4	0.88	-31.44	35.91	0.82	-32.17	39.33
Flowerpots	0.89	-28.56	28.45	0.87	-28.73	30.23
Lampshade1	0.83	-33.99	24.36	0.82	-34.41	25.04
Lampshade2	0.84	-33.92	22.90	0.83	-34.36	23.33
Midd1	0.91	-29.85	26.57	0.89	-30.10	28.56
Midd2	0.90	-30.01	44.92	0.89	-30.24	46.03
Monopoly	0.86	-34.04	90.35	0.83	-34.33	89.11
Plastic	0.94	-25.85	29.51	0.92	-26.54	29.49
Rocks1	0.92	-21.67	27.82	0.88	-22.76	31.11
Rocks2	0.93	-21.22	33.51	0.89	-22.32	35.77
Woods1	0.92	-27.89	40.22	0.90	-27.76	42.10
Woods2	0.94	-25.05	28.45	0.92	-25.05	29.70

Fig. 2. Despite a few artefacts – mainly due to failures of the stereo matching – the results are convincing and visually plausible.

5 Conclusions

We presented a pipeline for uncalibrated view-synthesis of novel images. Our method is fully automatic and geometrically sound. We tested it on the conversion to 3DS, where the method produces promising results.

Future work will address the limitations of this work. Hole filling in the virtual image needs to be improved; where no information is available in the source images inpainting techniques (e.g. [5]) should be adopted. The probabilistic splatting step could also be improved by the working on superpixels (e.g.[1]). Heuristics to mitigate the effects of matching failures will be also investigated.



misprint

Fig. 3. Examples of the 3DS output. From left to right: reference image (left eye), virtual image (right eye), red-cyan anaglyph.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Trans. on Patt. Analysis and Machine Intell.* 34(11), 2274 – 2282 (2012)
2. Avidan, S., Shashua, A.: Novel view synthesis by cascading trilinear tensors. *IEEE Trans Vis. and Comp. Graph.* 4(4), 293–306 (1998)
3. Cheng, C.C., Li, C.T., Huang, P.S., Lin, T.K., Tsai, Y.M., Chen, L.G.: A block-based 2D-to-3D conversion system with bilateral filter. In: *Int. Conf. on Consumer Electronics*. pp. 1–2 (2009)
4. Choi, B.T., Lee, S.H., Ko, S.J.: New frame rate up-conversion using bi-directional motion estimation. *IEEE Trans. on Consumer Electronics* 46(3), 603–609 (2000)
5. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. on Image Proc.* 13(9), 1200–1212 (2004)
6. Fragneto, P., Fusiello, A., Magri, L., Rossi, B., Ruffini, M.: Uncalibrated view synthesis with homography interpolation. In: *2nd Joint 3DIM/3DPVT Conf.* pp. 270–277 (2012)
7. Fusiello, A.: Specifying virtual cameras in uncalibrated view synthesis. *IEEE Trans. on Circuits and Systems for Video Technology* 17(5), 604–611 (2007)

8. Fusiello, A., Irsara, L.: An uncalibrated view-synthesis pipeline. In: Proc. Int. Conf. on Image Analysis and Proc. pp. 609–614 (2007)
9. Fusiello, A., Irsara, L.: Quasi-euclidean epipolar rectification of uncalibrated images. *Machine Vis. and Appl.* 22(4), 663 – 670 (2011)
10. Gigengack, F., Jiang, X.: Improved uncalibrated view synthesis by extended positioning of virtual cameras and image quality optimization. In: Asian Conf. Comp. Vis., pp. 438–447 (2010)
11. Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A.: Robust statistics: the approach based on influence functions, vol. 114. John Wiley & Sons (2011)
12. Hirschmuller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: IEEE Conf. on Comp. Vis. and Patt. Rec. pp. 1–8 (2007)
13. Irani, M., Anandan, P.: Parallax geometry of pairs of points for 3d scene analysis. In: Europ. Conf. Comp. Vis., pp. 17–30 (1996)
14. Jung, Y.J., Baik, A., Kim, J., Park, D.: A novel 2D-to-3D conversion technique based on relative height-depth cue. In: IS&T/SPIE Electronic Imaging. pp. 72371U–72371U (2009)
15. Kolmogorov, V., Monasse, P., Tan, P.: Kolmogorov and Zabih's Graph Cuts Stereo Matching Algorithm. *Image Processing On Line* 4, 220–251 (2014)
16. Laveau, S., Faugeras, O.: 3-d scene representation as a collection of images and fundamental matrices. In: Proc. Int. Conf. Patt. Rec. vol. 1, pp. 689–691 (1994)
17. Lee, S.H., Kwon, O., Park, R.H.: Weighted-adaptive motion-compensated frame rate up-conversion. *IEEE Trans. on Consumer Electronics* 49(3), 485–492 (2003)
18. Malapelle, F., Fusiello, A., Rossi, B., Piccinelli, E., Fragneto, P.: Uncalibrated dynamic stereo using parallax. In: Proc. Int. Symp. on Image and Signal Proc. and Analysis (2013)
19. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Patt. Analysis and Machine Intell.* 12(7), 629–639 (1990)
20. Ramachandran, G., Rupp, M.: Multiview synthesis from stereo views. In: Int. Workshop. on Systems, Signals and Image Proc. pp. 341–345 (2012)
21. Sawhney, H.S.: 3D geometry from planar parallax. In: IEEE Conf. on Comp. Vis. and Patt. Rec. pp. 929–934 (1994)
22. Sashua, A., Navab, N.: Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Trans. on Patt. Analysis and Machine Intell.* 18(9), 873–883 (1996)
23. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Proc.* 13(4), 600–612 (2004)
24. Zhang, G., Hua, W., Qin, X., Wong, T.T., Bao, H.: Stereoscopic video synthesis from a monocular video. *IEEE Trans. on Vis. and Comp. Graph.* 13(4), 686–696 (2007)
25. Zhang, G., Jia, J., Wong, T.T., Bao, H.: Consistent depth maps recovery from a video sequence. *IEEE Trans. on Patt. Analysis and Machine Intell.* 31(6), 974–988 (2009)
26. Zhang, L., Vázquez, C., Knorr, S.: 3D-TV content creation: automatic 2D-to-3D video conversion. *IEEE Trans. on Broadcasting* 57(2), 372–383 (2011)
27. Zhu, C., Zhao, Y., Yu, L., Tanimoto, M.: 3D-TV System with Depth-image-based Rendering. Springer (2014)