Olsson, R. , Adhikarla, V. K. , Schwarz, S. & Sjöström, M. (2012). Converting conventional stereo pairs to multi-view sequences using morphing. In *Proceedings of the SPIE, vol 8288 : Conference on Stereoscopic Displays and Applications XXIII, Burlingame, CA, USA, 22 - 26 January 2012*.

http://dx.doi.org/10.1117/12.909253

Converting conventional stereo pairs to multi-view sequences using morphing

Roger Olsson*[(1)], Vamsi Kiran Adhikarla[(2)], Sebastian Schwarz[(1)] and Mårten Sjöström[(1)]

[(1)] Dept. of Information Technology and Media, Mid Sweden University, Sundsvall, Sweden
[(2)] School of Engineering, Blekinge Institute of Technology, Karlskrona, Sweden

ABSTRACT

Autostereoscopic multi view displays require multiple views of a scene to provide motion parallax. When an observer changes viewing angle different stereoscopic pairs are perceived. This allows new perspectives of the scene to be seen giving a more realistic 3D experience. However, capturing arbitrary number of views is at best cumbersome, and in some occasions impossible. Conventional stereo video (CSV) operates on two video signals captured using two cameras at two different perspectives. Generation and transmission of two views is more feasible than that of multiple views. It would be more efficient if multiple views required by an autostereoscopic display can be synthesized from these sparse set of views. This paper addresses the conversion of stereoscopic video to multiview video using the video effect morphing. Different morphing algorithms are implemented and evaluated. Contrary to traditional conversion methods, these algorithms disregard the physical depth explicitly and instead generate intermediate views using sparse sets of correspondence features and image morphing. A novel morphing algorithm is also presented that uses scale invariant feature transform (SIFT) and segmentation to construct robust correspondences features and qualitative intermediate views. All algorithms are evaluated on a subjective and objective basis and the comparison results are presented.

Keywords: 3D, stereo to multiview conversion, view synthesis, warping, field morphing

1. INTRODUCTION

The use of conventional stereo video (CSV) has in the last years produced a large amount of content to provide a 3D experience at the cinema, on TV, and in home entertainment systems such as video game consoles. CSV consists of a left and right view pair that is presented to the viewer's left and right eye respectively. This enables the fundamental depth cue binocular parallax, which greatly contributes to the experience of 3D. [1] A more realistic 3D experience also requires another depth cue: motion parallax, or look-around capability. Motion parallax relies on the availability of a *sequence* of views rather than just a single pair. The viewer then perceives different view pairs depending on his/her position relative to the display. Hence, converting from the single view pair of CSV, into a view sequence consisting of an arbitrary set of views in multi view video (MVV), is a necessity for both providing a more life-like 3D experience as well as enabling the presentation of CSV content on current and future autostereoscopic multi view displays. Many current CSV-to-MVV conversion methods use depth image based rendering (DIBR) for synthesizing intermediate views. [2] DIBR is also a vital component of the 3D video compression standard that is currently under way within MPEG 3DV. Given that the source signal is CSV the required depth is inferred using knowledge about the geometry of the stereo camera setup and the disparity map (D), which is calculated from the left and right view. Unfortunately, deriving the disparity map D is an ill-posed problem that is difficult to solve algorithmically with sufficient quality. Moreover, morphing, contrary to DIBR, does not rely on explicit pixel dense knowledge about scene depth, nor camera parameters in order to produce intermediate views. Instead a set of feature pairs with accompanying left to right (or vice versa) correspondence mappings are used together with an image warping method to generate novel views. Steps have been taken recently to investigate this alternative to DIBR.[3] This paper evaluates the approach of converting from CSV to MVV and the work is presented as follows. Section 2 describes the way morphing may be used for synthesizing novel views from CSV, and a set of design choices that was considered in this work. A description of the experimental design used to evaluate these design choices is presented in Section 3 and the evaluation results are shown in Section 4. Section 5 concludes the paper and elaborates on future work.

(*) Corresponding author. E-mail: Roger.Olsson@miun.se, Telephone: + (46) 60 14 86 98

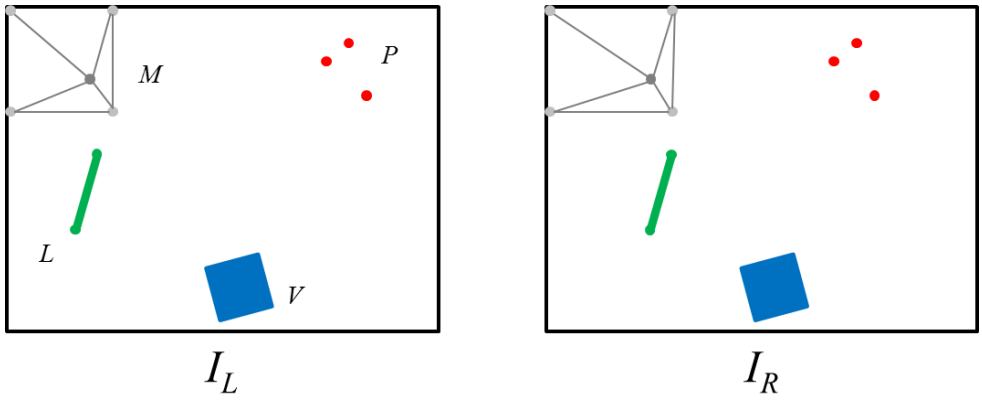


Figure 1. Left and right image I_L and I_R with examples of the feature types point P (red), line segment L (green), mesh M (gray), and feature vector V (blue). Note the features' changes in horizontal position between I_L and I_R.

2. VIEW SYNTHESIS USING MORPHING

Morphing is the combination of image warping and color interpolation. Image warping aims to align common features in two separate images using local 2D geometrical transformation functions, and color interpolation blends the two warped images together into an intermediate image.[4] Hence, a set of design choices must be made when adopting morphing as a way for view synthesis. What are the features that should be used? What metric should determine the feature correspondences between the two images? How spatially dense should the feature set be? What method should be used to produce a pixel dense warping function from the sparse feature set? In the following subsections we will present how we addressed these questions in this work.

2.1 Feature definition

Various geometrical forms can be considered describing common features in the two images, eg. points, line segments, splines, polygons, meshes.[4] Using higher order geometry allows for better approximations of the salient features of the images. However, higher order geometry often leads to a more complex subsequent image warping step. In this work we focus on four feature types:

- Point. $P = (u, v)$, where u and v are image coordinates.
- Line segment. $L = [P, Q] = [(u_P, v_P), (u_Q, v_Q)]$
- Mesh. $M = [P_1, P_2, P_3, \ldots]$
- Features vector. $V = [v_1, v_2, v_3, \ldots]$

Figure 1 illustrates the above feature types, where the V is presented as an image patch with location, direction and size. We denote the set of common features $\mathbb{F} = \mathbb{F}_L \cup \mathbb{F}_R$, where $\mathbb{F}_L$ and $\mathbb{F}_R$ are the features in the left image I_L and right image I_R respectively.

2.2 Correspondence matching

In order to match common features between two images, a source and destination set must be defined. In this work we set the source to be $\mathbb{F}_{\mathbb{L}}$ and the destination is then $\mathbb{F}_{\mathbb{R}}$. We investigate three different approaches to populate the source set of features:

- Local variance
  A sliding window W is used to calculate the local variance of I_L, and each resulting maxima are selected as a feature point $P \in \mathbb{F}_L$.

- SIFT
  The Scale Invariant Feature Transform transforms the image I_L into a number of feature vectors describing inter alia spatial location and dominant direction.[5] Each SIFT feature vector $V \in \mathbb{F}_L$ is invariant to translation, rotation, scale and partially invariant or robust to other changes or distortions.

- Canny-based line segmentation
  Canny edge detection results in a binary image where pixels that are determined to be part of an edge is set to 1. The edge points are then linked into a list of line segments,[6] where each line segment is selected as a feature line $L \in \mathbb{F}_L$.

A correspondence matching function $c : \mathbb{F}_L \rightarrow \mathbb{F}_R$ must be determined after the source set has been defined. How c is constructed can be largely divided into two main approaches:

I) Construct the destination set $\mathbb{F}_R$ in the same way as for the source set $\mathbb{F}_L$ and use a quality metric Q to determine the correspondences,

II) Copy $\mathbb{F}_L$ into $\mathbb{F}_R$ and adjust feature parameters using some transformation T.

An advantage of approach I) is that the sets $\mathbb{F}_L$ and $\mathbb{F}_R$ have equal properties in terms of feature type characteristics. A disadvantage is that I) might result in a 1-to-many and/or many-to-1 mapping instead of the desirable 1-to-1 relationship between $\mathbb{F}_L$ and $\mathbb{F}_R$. Hence, additional processing of c might be required to achieve this. In approach II) the property of 1-to-1 mapping is a natural consequence of transforming the source set. For the feature types P and M we use block based normalized cross-correlation (NCC) as Q, which determines the correspondences. Under the assumption of a perfectly calibrated parallel stereo camera setup the correspondence mapping is a strictly horizontal translation d. For each feature, located at $I_L\,(u, v)$, d is then calculated as:

$$\arg\max_d r(d) := \frac{1}{\sigma^2_{I'_L}\sigma^2_{I'_R}} \sum_{s,t \in B} \left(I_L(u+s, v+t) - \bar{I}'_L\right)\left(I_R(u+s+d, v+t) - \bar{I}'_R\right) \tag{1}$$

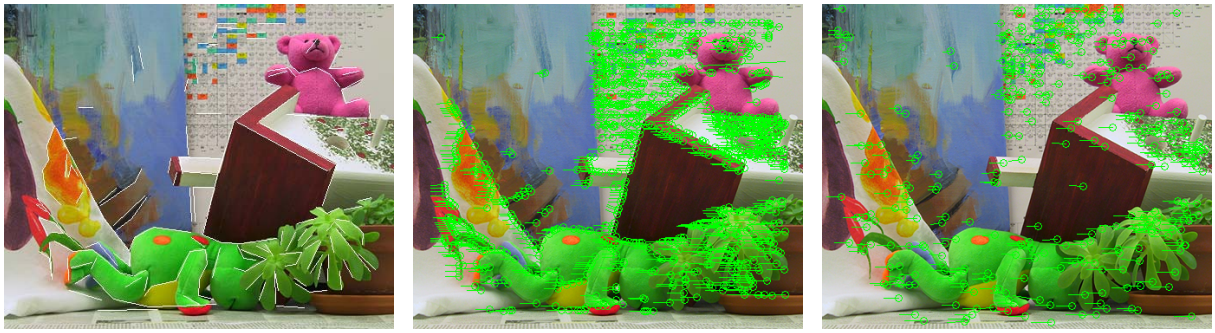
For the feature vector V stemming from SIFT, the quality metric used is the minimum Euclidian distance between each $V \in \mathbb{F}_L$ and all feature vectors in $\mathbb{F}_R$. When using line segments L, approach I) is not as straightforwardly pursued as when the feature type is P or M. Although the main difference between $\mathbb{F}_L$ and $\mathbb{F}_R$ is expected to be horizontal translation a common line pair might differ slightly in length and slope. This gives an increased dimensionality of the search space within which Q must be evaluated for correspondence matching. We address this complexity problem by adopting approach II). Each $L \in \mathbb{F}_L$ is divided into its start and end point (P_1, P_2) and the corresponding point pair is searched for in I_R using Eq. (1). In order to remedy incorrect mismatches each resulting $L \in \mathbb{F}_R$ is compared to its corresponding $L \in \mathbb{F}_L$. Any significant deviation in slant disqualifies the line pair, which is then discarded from $\mathbb{F}$.

2.3 Warping

The above step results in feature correspondences between the left and right image that are sparse in relation to the pixel lattice, as the examples in Figure 2 shows. Figure 2 (a) only presents $\mathbb{F}_L$, in order not to clutter the image I_L, whereas Figures 2 (b) and (c) also show $\mathbb{F}_R$ as overlaid circles connected with $\mathbb{F}_L$ using horizontal lines. However, in order to sufficiently align the two images so that the subsequent blending does not introduce ghosting, an accurate pixel-dense correspondence must exist. A warping function performs the necessary transition from sparse feature-correspondence to dense pixel-correspondence, which may be of arbitrary complexity.[7] In this work we have evaluated thin plate spline interpolation, mesh morphing,[4] and field morphing.

Thin plate spline interpolation is the two-dimensional equivalent to one-dimensional cubic spline interpolation.[8] The set of splines are centered on the sparse feature correspondences c that, when weighted together, produces an interpolation surface from which a pixel-dense warping function can be derived. Compared to the other warping methods, spline interpolation relies more on the availability of a large number of feature correspondences in order to produce intermediate images of high quality.

In mesh morphing the defined feature points are considered nodes in a mesh M. Restricting the mesh to be made from triangles allows Delaunay triangulation to be applied to the feature points, which results in triangle meshes similar



(a) (b) (c)

Figure 2. Example of (a) feature set $\mathbb{F}_L$, derived using Canny based line segmentation; and (b) sparse feature correspondences c, based on local variance and NCC; and (c) c based on SIFT and minimized Euclidian distances

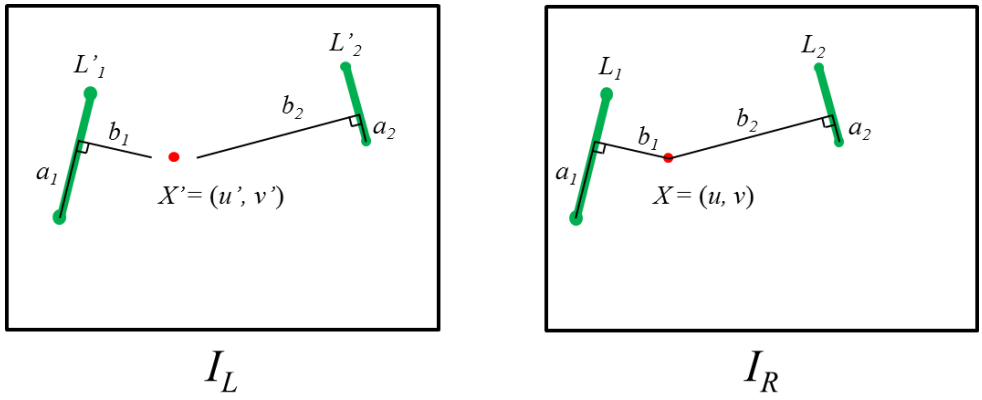


Figure 3. Field morphing where two correspondence line pairs (L'_1, L_1) and (L'_2, L_2) contribute to the correspondence for the pixel pair (X', X). Note that the calculated position of X' is the average of the two positions resulting from each line pair.

to those shown in Figure 1. The benefit of a mesh is that interpolation is performed within each triangle. This increases the quality of the resulting warping function as interpolated values are determined locally and not influenced from distant feature points.

Contrary to the two previous warping methods, field based morphing relies on line pairs for correspondence rather than point features.[9] Figure 3 shows a field morphing example where two correspondence line pairs are used to determine the pixel correspondence of X and X'. When a single line pair is used the line pair jointly defines a coordinate system transformation, which is applied to all pixels in I_L and I_R. For each pixel position X in the destination I_R, its distance to line L (in terms of a and b) is calculated. The corresponding pixel position X' in I_L is then found using the same distances a and b relative to L'. For more line pairs than one, each transformation contributes to an weighted average that determines the correspondence between X and X'. Pixels nearby a specific line are influenced more by that specific line than by others. The interested reader is referred to the paper by Beier and Neely for further details about field morphing. [9]

After the pixel-dense correspondence mapping, or warping function w, is determined, the color values of an intermediate image I_n are calculated using linear interpolation:

$$I_n = \frac{N-n}{N} I_L^n + \frac{n}{N} I_R^n, \tag{2}$$

where $N-2$ is the total number of intermediate images between I_L and I_R. I_L^n and I_R^n are the intermediate left and right warped images respectively, where a fraction of the translation determined by the warping function w is used. Extending

the morphing process to consider extrapolated images, to the left of I_L or to the right of I_R, is straightforward although it may require additional processing to handle e.g. image border constraints.

3. EXPERIMENT DESIGN

Section 2 presented a set of tools contributing to view synthesis using morphing. These tools were evaluated for

1. intermediate view quality using PSNR as a function of image complexity, disparity variation and maximum disparity
2. computational complexity in terms of execution time as a function of feature set size and image resolution

Execution time was measured on an Intel Pentium Dual-Core 2.1GHz processor with 4GB RAM running Windows 7 64bit and Matlab R2009b.

A set of multiview image sequences were selected for evaluation based on their individual number of unique objects in depth and total disparity range; features that were determined using disparity histogram. Four image sets were used to evaluate view quality: *Teddy* (450x375), *Venus* (434x383), *Tsubuka*, *Art* (1390x1110) and *Penguin* (640x360); [10–12] each containing at least three images used as left I_L, right I_R, and middle I_M. The latter located halfway between I_L and I_R. In *Teddy*, there are a larger number of objects distributed at different disparities or depth levels within the scene, but with a lower spread in disparity of a particular object. For *Venus* on the other hand the number of objects are less but the spread in disparity of an object is larger. In *Penguin* there is only one foreground object in addition to a background wall and the maximum disparity is large when compared to the other images. In all PSNR calculations I_M was used as reference to a generated intermediate image I_n with $n = \frac{N}{2}$. When studying PSNR as a function of maximum disparity, four different stereo pairs were selected from the image sets, with increasing inter-camera distance and thereby disparity.

4. RESULTS

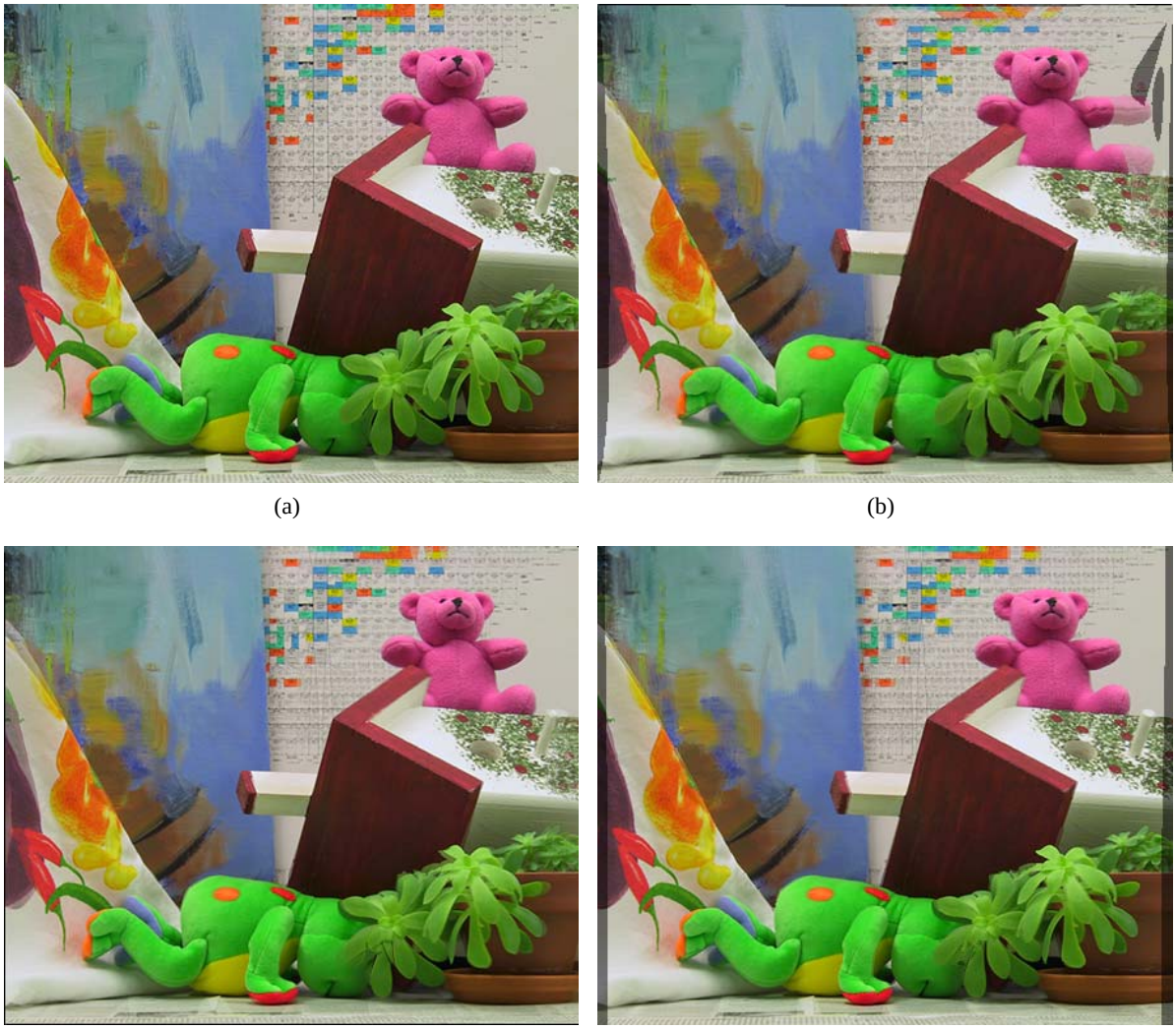
4.1 Generated intermediate images

Figure 4 shows typical examples of intermediate images generated by the evaluated methods spline interpolation using NCC points, mesh morphing using canny-based line segment points, and field morphing using canny-based line segments. When studying I_n in Figure 4(b), and the utilized c of Figure 2(b), the ghosting seen to the right of the pink bear is due to mismatches when performing NCC. Reducing the variance threshold increases the number of correspondences, potentially increasing the intermediate image quality. However, this also potentially increase the number of erroneous correspondence matches. Using canny-based line segment points as mesh nodes reduces the risk of assigning non-correct correspondence since points are restricted to lie on edges in the image. However, ghosting might still appear like at the top left intersection of the white and purple cloth. In this case due to lack of correspondence features, as can be seen in the top left part of Figure 2(a), were no lines have been assigned to that specific edge. This is less of a problem for field morphing, as can be seen in Figure 4(d), even though both the mesh and field morphing results are using the same canny edge detector output. The reason being that field morphing utilize the edges as line segments and calculate the warping function globally, contrary to the point based local interpolation performed by mesh morphing. Remember that pixels close to, or at, a line, will be preserved in field morphing.

4.2 Quality as a function of disparity

In order to evaluate the quality as a function of disparity, pixels belonging to a specific disparity range must be extracted and evaluated separately. This is done using the available ground truth disparity information D present in the example image sets. The evaluated approaches are applied to *Art* and the disparity is sampled at four ranges ($70<d<90$, $130<d<140$,$170<d<180$, $d>213$). The resulting PSNR is presented in the middle of each disparity interval in Figure 5(a). Field morphing with a single line placed at the foreground, using SIFT correspondences, results in superior quality at foreground disparities. As expected this comes at the expense of worse quality at middle- and background parts. Spline interpolation and mesh morphing performs similarly although spline results in better foreground quality, which is caused by NCC providing better correspondences for *Art* than Canny.

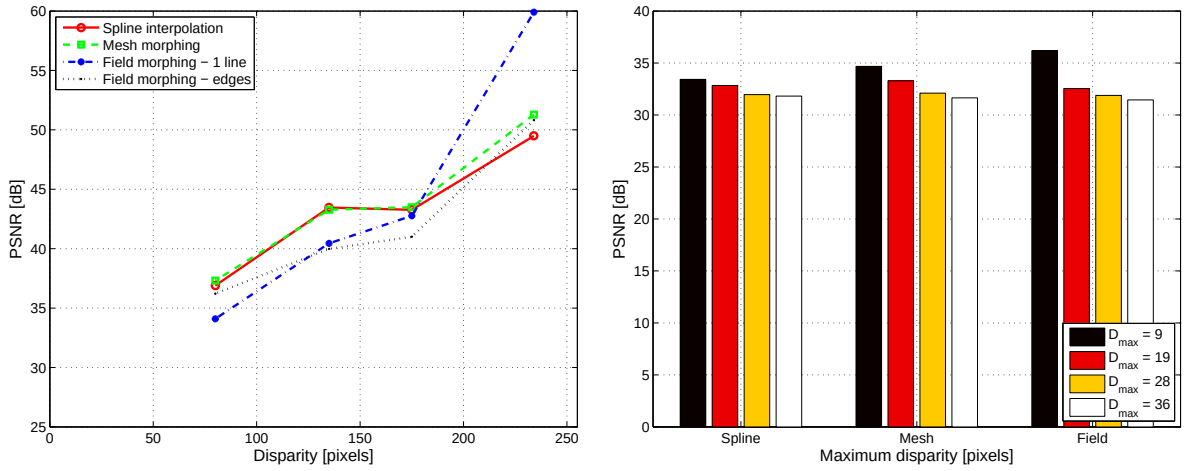
Quality as a function of maximum disparity is shown in Figure 5(b) where it is seen that field morphing is better to use for stereo images with lower disparity ranges. The difference between the methods are not big though. They all have more difficulties in producing high quality images when the inter-camera distance, and the resulting disparity, increases.



(a) (b)

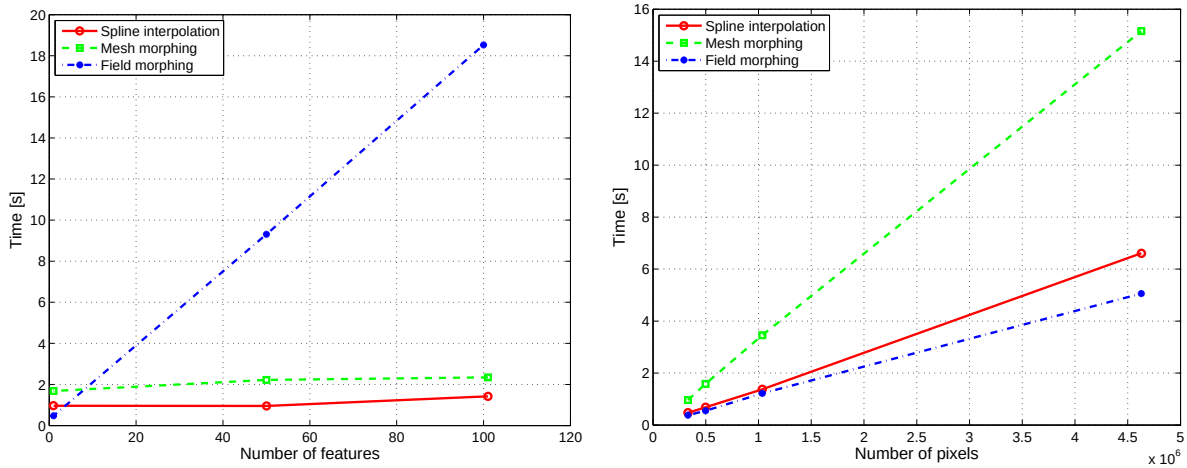
(c) (d)

Figure 4. Morph results compared to (a) original middle image I_M and intermediate images I_n with $n = \frac{N}{2}$ for (b) spline interpolation, (c) mesh morphing, and (d) field morphing.



(a) (b)

Figure 5. Peak Signal to Noise Ratio as a function of (a) disparity variation and (b) disparity range.



(a) number of features (b) image size

Figure 6. Execution time as a function of (a) number of features and (b) image size.

4.3 Execution time

The size of the feature set has a significant impact on the quality of the final morph, but also on execution time. This dependency is shown in Figure 6(a). From the figure, it can be noticed that field morphing suffers from the drawback that execution time increases significantly as the size of the feature set grows. This is because all line segments need to be referenced for every pixel. In case of field morphing with a single line, the execution time is less than that required by spline interpolation and mesh morphing. Dependency of execution time on image size is shown in Figure 6(b). Image size has a significant impact on mesh morphing because of the increase in number of pixels to be handled in a Delaunay triangle.

5. CONCLUSION

In this paper we have investigated the use of morphing for view synthesis when converting conventional video frames to multi view video content. Three morphing methods have been evaluated with respect to image quality and computational complexity. We have concluded that mesh morphing produce high PSNR values but is sensitive to having the mesh nodes positioned at edges in the images. If not, mesh morphing tends to introduce stretched and/or squeezed regions, which

reduce quality. Field morphing with line correspondences generated by Canny also provides high quality results, yet at the expense of long execution times. We have proposed a way to reduce the execution time by transforming found edges, rather than finding new, when producing the required correspondence feature set. Using this field morphing approach with single line gives robust performance and good quality in solving the CSV to MVV conversion problem.

ACKNOWLEDGEMENT

This work has been supported by grant 00156702 of the EU European Regional Development Fund, Mellersta Norrland, Sweden, and by grant 00155148 of Länsstyrelsen Västernorrland, Sweden.

REFERENCES

[1] Siegel, M. and Nagata, S., "Just enough reality: Comfortable 3-D viewing via microstereopsis," *IEEE Transactions on Cicuit and Systems for Video Technology* **10**, 387 – 396 (April 2000).

[2] Smolic, A., Müller, K., Dix, K., Merkle, P., Kauff, P., and Wiegand, T., "Intermediate view interpolation based on multiview video plus depth for advanced 3d video systems," in [*IEEE International Conference on Image Processing*], (2008).

[3] Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., and Gross, M., "Nonlinear disparity mapping for stereoscopic 3d," *ACM Trans. Graph.* **29** (2010).

[4] Wolberg, G., "Recent advances in image morphing," in [*Proc. Computer Graphics International*], (1996).

[5] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* **60**(2), 91–110 (2004).

[6] Kovesi, P. D., "MATLAB and Octave functions for computer vision and image processing." Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[7] Ruprecht, D. and Müller, H., "Image warping with scattered data interpolation methods," *IEEE Computer Graphics and Applications* **15**(2) (1995).

[8] Belongie, S., "Thin plate spline." MathWorld–A Wolfram Web Resource. Created by Eric W. Weisstein. http://mathworld.wolfram.com/ThinPlateSpline.html.

[9] Beier, T. and Neely, S., "Feature-based image metamorphosis," in [*Proceedings of the 19th annual conference on Computer graphics and interactive techniques*], *SIGGRAPH '92*, 35–42, ACM, New York, NY, USA (1992).

[10] Scharstein, D. and Szeliski, R., "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision* **47**, 7 – 42 (2002).

[11] Scharstein, D. and Szeliski, R., "High-accuracy stereo depth maps using structured light," in [*IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*], (2003).

[12] Hirschmüller, H. and Scharstein, D., "Evaluation of cost functions for stereo matching," in [*IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*], (2007).