# VIEWPOINT INTERPOLATION: DIRECT AND VARIATIONAL METHODS

*Sergi Pujades* *Frédéric Devernay*

Inria - PRIMA Team,
Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France,
CNRS, LIG, F-38000 Grenoble, France.

## ABSTRACT

We address the topic of novel view synthesis from a stereoscopic pair of images. The techniques have mainly 3 stages: the reconstruction of correspondences between the views, the estimation of the blending factor of each view for the final view, and the rendering. The state of the art has mainly focused on the correspondence topic, but little work addresses the question of which blending factors are best. The rendering methods can be classified into "*direct*" methods, defining the final image as a function of the original images, and "*variational*" methods, where the synthesized image is expressed as the solution minimising an energy. In this paper, we experiment different combinations of the blending factors and the rendering method, in order to demonstrate the effect of these two factors on the final image quality.

***Index Terms—*** Viewpoint interpolation, image-based rendering, blending factors, variational method.

## 1. INTRODUCTION

Novel view synthesis from a stereoscopic pair of images has been extensively studied, for example for content creation for glasses-free 3D displays from binocular stereoscopic content. Those techniques proceed in general in 3 stages: the estimation of correspondences between the novel and the reference views, the estimation of the contribution (or weight) of each view in the final view, and the rendering method. The stereo correspondence estimation problem has been largely explored [**?**], but few works have formally studied which is the "*correct*" leverage between the contributions of each view. We will call this leverage the "*blending factors*" of each view. The most common blending factors considered in the literature are the local deformation of the image introduced by the change in the view point, and the distance between the new view and the reference ones. Yet, several questions arise: How is each blending factor choice formally supported? How does the synthesized view quality depend on these blending factors? Ultimately, what blending factors should be preferred?

This work was done within the Action 3DS project, founded by the French *Caisse des dépôts et consignations*

In the view interpolation domain, methods can be clustered in 2 groups. The "*direct*" methods and the "*variational*" methods. Most state of the art methods are direct: the color of a pixel in the final image is given as a function of the colors of the corresponding pixels in the reference images. In the variational methods, the final image minimises an energy corresponding to a maximum a posteriori, often derived from a generative model. Using the Bayesian formalism, the blending factors between the views can be formally derived. However, those optimization techniques require heavier computations than the direct ones. We would like to know if the results obtained with these methods compensate for their computational complexity. In this article we propose a study to analyse the impact of the blending factors and the used method in the final result. We evaluate those methods on Lambertian and non-Lambertian scenes in order to see, in which case, which choice is better.

## 2. PRIOR WORK

Viewpoint interpolation methods belong to the largely studied field of image-based rendering [1]. *Unstructured Lumigraph* [2] introduces the desirable properties an ideal image rendering method should have: "*use of geometric proxies*", "*unstructured input*", "*epipolar consistency*", "*minimal angular deviation*", "*continuity*", "*resolution sensitivity*", "*equivalent ray consistency*", and finally "*real-time*". Moreover they state the weight of each reference image when rendering the final view. Authors also present a direct method taking into account all this desirable properties. The "*minimal angular deviation*" is measured with the angles between the optical rays of the rendered and reference images. The "*resolution sensitivity*" is enforced by computing an approximation of the Jacobian of the planar homography between the rendered and the reference image. The balance between this factors is adjusted depending on the scene. Precisely, "*resolution sensitivity*" has in most experiments a tiny weight compared to the "*minimal angular deviation*". We would like to put forward the influence of those weights in the final result.

Methods addressing the problem of view interpolation from a stereoscopic pair of images use two blending factors: most of them consider the normalised distance $\alpha$ between

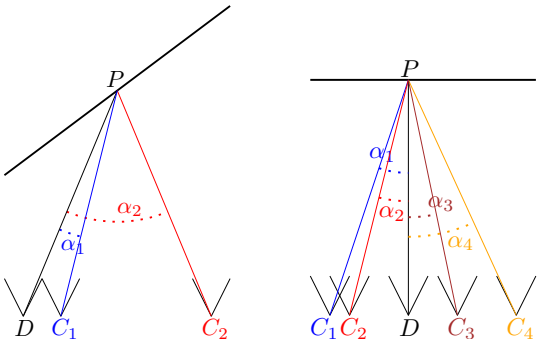


**Fig. 1**. Rendering view $D$ from $C_i$ using [7]. Left: resolution sensitivity will prefer $C_2$ to $C_1$, even-though angular distance would prefer $C_1$ over $C_2$. Right: flat scene parallel to views: all views will have the same blending factor, even-though their angular distance is different for each of them.

the new image and the reference image. The two associated blending factors are $\alpha$ and $(1 - \alpha)$ [3, 4, 5]. Other methods measure the deformation of the image [6], using the Jacobian of the planar homography between the rendered image and the reference image.

While addressing the super-resolution problem, Wanner and Goldluecke [7] propose a very general variational method to generate images at new viewpoints. They present a generative model describing the image formation process and establishing the energy corresponding to its maximum a posteriori, using the Bayesian formalism. This formalisation brings them to derive the blending factor of the images to be given by the determinant of the Jacobian of the transformation between the final image and the reference one. While these blending factors take into account the "*resolution sensitivity*", the "*minimal angular deviation*" is overlooked. It doesn't appear in the equations and Fig. 1 illustrates two configurations showing the contradictions with [2].

Alternative methods are proposed by [8, 9] based on the *deformable meshes* from Gal et al.[10]. Their main hypothesis is that artifacts introduced by mesh deformations are visually more acceptable than those produced by image blending. However, in case of important deformations of the images, it would be interesting to merge two images generated with these techniques. Unfortunately this question is avoided in their works.

So, in the literature different blending factors are used, but to our knowledge, there is no study directly comparing their performance.

## 3. VIEWPOINT INTERPOLATION METHODS

All methods will have the same input: a rectified pair of stereoscopic images, 2 disparity maps with the correspondences between the views, and a value $\alpha \in [0, 1]$ corresponding to the position between the reference views to be rendered, being 0 the left image position, and 1 the right image position.

### 3.1. Direct Methods

The direct methods first compute the inverse transformations going from the final image to each reference image using the disparity maps and the $\alpha$ value. Pixels without a correspondence are labeled as invalid. This typically happens at disocclusions. The final image is generated using the color of the corresponding pixels in the reference images, and using a linear interpolation if coordinates have floating point precision.

We have chosen to study 4 different weights for the blending factors. First we consider the "*classic*" weight $((1 - \alpha), \alpha)$ taking into account the "*minimal angular deviation*". The second weight also fulfills this property. It is $((1 - \alpha)^2, \alpha^2)$. The third chosen weight ignores the "*minimal angular deviation*" and assigns to each image the same weight. This is done independently of the $\alpha$ value and the local deformation of the image transformation. The fourth weight is proportional to the local deformation of the transformation of the image, as described in [7], $|\det D\tau|^{-1}$. In our case $\tau$ is the transformation given by the disparity and we compute its deformation with finite differences. In all 4 cases weights are normalized so that their sum is 1. If one of the two corresponding pixels is marked as invalid we assign a 0 weight to it, and a 1 weight to the other pixel. If both corresponding pixels are invalid, we mark the pixel as invalid. If one or both of the corresponding pixels are valid, we assign to the final pixel the weighted sum of the values. Notice that at this stage some pixels of the rendered image are labeled as invalid. We explain how we handle those cases in section 3.3.

**Fig. 2**. Example images from the datasets: "*aloe*" from [14] and "*tarot*", "*bracelet*" and "*amethyst*" from [15].

| TAROT | *view 2* | | *view 3* | | *view 8* | | *view 9* | | *view 14* | | *view 15* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Direct Methods* | | | | | | | | | | | | |
| $\alpha(1-\alpha)$ | **33.65** | **25** | **31.76** | 35 | **30.19** | **42** | 30.84 | **36** | **32.60** | **29** | **33.81** | **24** |
| $\alpha^2(1-\alpha)^2$ | 33.49 | 26 | 31.47 | *38* | 30.21 | **42** | 30.78 | **36** | 32.38 | 31 | 33.58 | 26 |
| Constants | 32.20 | 27 | 31.24 | **34** | 30.16 | **42** | **30.88** | **36** | 31.61 | 33 | 32.64 | 26 |
| Deformation | *31.81* | *29* | *30.94* | 35 | 29.96 | 44 | 30.67 | 37 | 31.42 | 34 | 32.40 | 27 |
| *Variational Methods* | | | | | | | | | | | | |
| $\alpha(1-\alpha)$ | 33.60 | **25** | 31.65 | 35 | 29.95 | *45* | 30.45 | *39* | 31.96 | 33 | 33.10 | 27 |
| Deformation ([7]) | 32.48 | 27 | 31.31 | 35 | *29.78* | *45* | *30.29* | *39* | *31.22* | *36* | *32.33* | *28* |
| BRACELET | *view 2* | | *view 3* | | *view 8* | | *view 9* | | *view 14* | | *view 15* | |
| *Direct Methods* | | | | | | | | | | | | |
| $\alpha(1-\alpha)$ | 36.12 | **14** | 32.87 | 28 | **33.81** | **24** | 33.69 | **24** | 33.71 | **23** | 35.67 | **16** |
| $\alpha^2(1-\alpha)^2$ | 36.06 | 15 | 32.66 | 29 | **33.81** | **24** | **33.71** | **24** | 33.46 | 25 | 35.48 | **16** |
| Constants | 34.20 | 20 | 32.50 | 30 | 33.79 | **24** | 33.67 | 25 | 33.21 | 26 | 34.59 | 19 |
| Deformation | *33.68* | *23* | *32.20* | *32* | 33.46 | 25 | 33.40 | 26 | *32.94* | *28* | *34.25* | *21* |
| *Variational Methods* | | | | | | | | | | | | |
| $\alpha(1-\alpha)$ | **36.27** | **14** | **32.92** | **27** | 33.46 | 26 | 33.30 | *27* | **33.80** | 24 | **35.77** | **16** |
| Deformation ([7]) | 34.52 | 19 | 32.44 | 30 | *33.11* | *27* | *33.15* | *27* | 33.06 | 27 | 34.47 | *20* |

**Table 1**. Numerical results of the comparison between real and rendered images for the datasets "*tarot*" and "*bracelet*". For each view and method we present PSNR in dB (the bigger the better the signal) and DSSIM scaled with $10^{-4}$ (the smaller, the most similar the images are). Views (2, 3, 14 and 15) are close to the reference views. Views (8 and 9) are central viewpoints. The best value for each view is in bold, and the worse in italic.

### 3.2. Variational Methods

Variational methods find the image minimising an energy. Those energies have often two parts: the data term and the regularizer.

$$E(u) = E_{\text{data}}(u) + \lambda E_{\text{prior}}(u). \tag{1}$$

In the Bayesian formalism, the first term can be obtained by modeling the image formation process. A common hypothesis is to consider the sensor noise as a zero mean Gaussian. Its maximum a posteriori energy matches the minimum of a least squares overdetermined system. To write $E_{\text{data}}(u)$ we note the final image as $u$, the reference image $i$ as $v_i$, the definition domain of the images as $\Omega_i$. $m_i$ is a binary operator telling if the pixel is visible in the final image and $\tau_i$ the transformation from the reference image $i$ into the final image. $\omega_i$ is the weight of the pixels of image $i$. Then

$$E_{\text{data}}(u) = \sum_{i=1}^{n} \frac{1}{2} \int_{\Omega_i} \omega_i m_i ((u \circ \tau_i) - v_i)^2. \tag{2}$$

For the variational methods we have chosen to use two different weights. The first is the one proposed by [7] corresponding to the deformation of the transformation of the images:

$$\omega_i = |\det D\tau_i|^{-1} \tag{3}$$

The second weight is the "*classic*" $((1-\alpha),\ \alpha)$, taking into account the "*minimal angular deviation*". Although this second weight is not formally derived from a known generative model, the energy is well defined and can be minimized. Again, all weights are normalized so that their sum is 1.

### 3.3. Prior or regularizer

In the Bayesian formalism a prior is used in order to decide in cases where no information is available for some areas, or when several candidates for a solution are possible. In the direct methods this phenomena also arises if none of the reference images can propose information for the desired area. A "*classical*" prior in computer vision is the total variation [11]: $E_{\text{prior}}(u) = \int_\Gamma |Du|$ . It has the important property to provide a convex energy and to be well suited for the gradient descend minimization techniques. In the direct methods, we fill the remaining invalid pixels with a hole filling technique to propagate valid information from the nearest neighbors. Its behaviour is similar to the role played by the prior [11] during the minimization, so it seems a fair choice in order to provide a reasonable comparison. We highlight that our goal is not to obtain the best possible images, but to provide a fair comparison between "*direct*" and "*variational*" methods.

### 3.4. Experiments

In order to compare the different methods and the impact of the blending factors in the final result we have used multiple datasets from the "*Middlebury Stereo Dataset*" [14] and the "*Stanford Lightfield Archive*" [15]. Fig. 2 shows examples of the used images. The first one proposes very lambertian scenes (color does not depend on the viewing angle). Each scene has 6 aligned views. We use the most distant ones (1 and 6) to generate the other 4. The second database provides more challenging scenes, including highly specular reflections, transparencies and inter-reflections. Each dataset

| AMETHYST | *view 2* | | *view 3* | | *view 8* | | *view 9* | | *view 14* | | *view 15* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Direct Methods* | | | | | | | | | | | | |
| $\alpha(1-\alpha)$ | **33.84** | 1247 | **32.67** | 1265 | 30.66 | 1292 | 30.88 | 1299 | 33.60 | 1305 | 34.86 | 1305 |
| $\alpha^2(1-\alpha)^2$ | 33.81 | *1255* | 32.61 | *1282* | **30.69** | 1294 | 30.84 | 1300 | 33.42 | *1325* | 34.79 | *1313* |
| Constants | 32.16 | 1218 | 31.62 | 1239 | 30.63 | 1288 | **30.91** | 1294 | 32.67 | 1273 | 33.13 | 1274 |
| Deformation | *31.56* | 1239 | 31.07 | 1259 | *30.27* | *1307* | *30.54* | *1313* | *32.00* | 1293 | *32.36* | 1296 |
| *Variational Methods* | | | | | | | | | | | | |
| $\alpha(1-\alpha)$ | 32.80 | **1093** | 31.39 | **1116** | 30.30 | **1160** | 30.66 | **1168** | **33.77** | **1142** | **34.98** | **1141** |
| Deformation ([7]) | 31.87 | 1107 | *30.96* | 1126 | *30.27* | 1163 | 30.63 | 1170 | 33.01 | 1150 | 33.68 | 1152 |

**Table 2**. Numerical results for the dataset "*amethyst*". Same measures as in Tab 1 are displayed.

has 256 view per scene arranged in a regular 16x16 grid. We have selected the central line (the 8th) and used its most distant viewpoints (1st and 16th) to generate the other 14. This way we can compare the original images with the generated ones. To do so we have used two state of the art measures. The "*Peak Signal to Noise Ratio*" (PSNR), in dB, the bigger, the better is the signal. And the "*Structural SIMilarity*" (SSIM) developed to measure the visual similarity between two images. We report results with a distance based on SSIM: $\mathrm{DSSIM} = \frac{1-\mathrm{SSIM}}{2}$, having no units. The smaller, the more similar are the images.

For the study we have used standard methods for the disparity maps computation [12] and [13]. Disparity maps obtained with [13] are not dense but we have completed them using the hole filling technique in [12]. The $\lambda$ parameter in eq. 1 was empirically set to $0.15$ for all experiments.

### 3.5. Results

We present the obtained results for the datasets "*tarot*", "*bracelet*" and "*amethyst*", using the disparities computed with SGBM [13]. Results with dataset "*aloe*" of "*Middlebury Dataset*" are very similar to those from "*tarot*" and are not presented. Results obtained with disparities computed with [12] are very close to the presented ones and are skipped.

Input images are coded in sRGB space. It is advised to convert them into RGB-linear when operating with pixel values. We tested using the sRGB and the linear versions and very similar results were obtained. Running times for direct methods are around $\frac{1}{30}$s for 1024x1024 images (real-time). Variational methods were solved using a gpu implementation of FISTA [16, **?**]. Running time was about 1s for 1024x1024 images (not real-time).

Dataset "*tarot*" has a crystal ball with transparencies, violating the lambertian model. However colors on the rest of the scene do not change from one view to the other. Dataset "*bracelet*" has some specualr highlights on the metal, but the color difference between the left and right views is very small. Dataset "*amethyst*" is more complex. It has inter-reflections and highlight effects.

In Tab. 1, the difference between the methods for a fixed scene is small. PSNR and DDSIM values are very close. In "*bracelet*", results of view 2 and 15 are slightly better when taking into account the "*minimal angular deviation*" $(\alpha, (1-\alpha))$ and $(\alpha^2\,(1-\alpha)^2)$ both in direct and variational methods. This was expected as those methods are capable of better rendering the highlights. However the improvement dissipates quickly as we render the next views (3 or 14). No significant difference between the direct and variational methods can be reported.

Notice how PSNR values on Tab. 1 ("*tarot*" and "*bracelet*") and Tab. 2 ("*amethyst*") are on the same order of magnitude; but DSSIM values are higher on Tab. 2 than on Tab. 1. We believe that this difference appears due to the fact that the scene is more complex, together with a greater capability of the DSSIM measure to compare the visual quality. Again, for each view, tendencies are similar as in Tab.1. However we report a significant difference between the DDSIM values of direct and variational methods. For complex scenes, variational methods are capable of better reconstructing the structure of the image.

## 4. DISCUSSION AND CONCLUSION

In this paper we have presented a study of the influence of the blending factors and the method in the result of the viewpoint interpolation techniques. We have compared several blending factors and two groups of methods, direct and variational. Conducted experiments show that for lambertian scenes, the choice of the blending factors has insignificant impact. Moreover, results obtained with direct or variational methods are equivalents, so direct methods should be preferred thanks to their simplicity. In the non-lambertian scenes, the choice of the blending factors considering the "*minimal angular deviation*" produce slightly better results with both direct and variational methods, although the improvements are only visible when rendering new images close to a reference view. However, the variational methods are capable of better render the structure of the image, obtaining improved results. In this case variational methods considering "*minimal angular deviation*" should be preferred.

In future work it would be interesting to continue this study for more general configurations: multiple input views and general position of reference and final view.

## 5. REFERENCES


[1] Heung-Yeung Shum, Shing-Chouw Chan, and Sing Bing Kang, *Image-based rendering*, Springer, 2007.

[2] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen, "Unstructured Lumigraph rendering," in *Proc. SIGGRAPH*. ACM, 2001, pp. 425–432.

[3] Joon Hong Park and Hyun Wook Park, "Fast view interpolation of stereo images using image gradient and disparity triangulation," *Signal Processing: Image Communication*, vol. 18, no. 5, pp. 401–416, 2003.

[4] Aljoscha Smolic, Karsten Muller, Kristina Dix, Philipp Merkle, Peter Kauff, and Thomas Wiegand, "Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems," in *Proc. ICIP*. IEEE, 2008, pp. 2448–2451.

[5] Frédéric Devernay and Adrian Ramos Peon, "Novel view synthesis for stereoscopic cinema: detecting and removing artifacts," in *Proceedings of the 1st international workshop on 3D video processing*. ACM, 2010, pp. 25–30.

[6] Maxime Lhuillier and Long Quan, "Image interpolation by joint view triangulation," in *CVPR*. IEEE, 1999, vol. 2.

[7] Sven Wanner and Bastian Goldluecke, "Spatial and angular variational super-resolution of 4D light fields," in *Proc. ECCV*, pp. 608–621. Springer, 2012.

[8] Tao Yan, Rynson WH Lau, Yun Xu, and Liusheng Huang, "Depth mapping for stereoscopic videos," *IJCV*, pp. 1–15, 2013.

[9] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross, "Nonlinear disparity mapping for stereoscopic 3d," *TOG*, vol. 29, no. 4, pp. 75, 2010.

[10] Ran Gal, Olga Sorkine, and Daniel Cohen-Or, "Feature-aware texturing," in *Proceedings of the 17th Eurographics conference on Rendering Techniques*. Eurographics Association, 2006, pp. 297–303.

[11] Leonid I Rudin, Stanley Osher, and Emad Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[12] Mikhail Sizintsev and Richard P Wildes, "Coarse-to-fine stereo vision with accurate 3d boundaries," *Image and Vision Computing*, vol. 28, no. 3, pp. 352–366, 2010.

[13] Heiko Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 2, pp. 807–814.

[14] Heiko Hirschmuller and Daniel Scharstein, "Evaluation of cost functions for stereo matching," in *CVPR*. IEEE, 2007, pp. 1–8.

[15] V. Vaish and A. Adams, "The (New) Stanford Light Field Archive," `http://lightfield.stanford.edu`, 2008.

[16] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIIMS*, vol. 2, no. 1, pp. 183–202, 2009.