A Bayesian Approach for Selective Image-Based Rendering using Superpixels

Rodrigo Ortiz-Cayon, Abdelaziz Djelouah, George Drettakis
Inria
{rodrigo.ortiz-cayon, abdelaziz.djelouah, george.drettakis}@inria.fr

Abstract

Image-Based Rendering (IBR) algorithms generate high quality photo-realistic imagery without the burden of detailed modeling and expensive realistic rendering. Recent methods have different strengths and weaknesses, depending on 3D reconstruction quality and scene content. Each algorithm operates with a set of hypotheses about the scene and the novel views, resulting in different quality/speed trade-offs in different image regions. We present a principled approach to select the algorithm with the best quality/speed trade-off in each region. To do this, we propose a Bayesian approach, modeling the rendering quality, the rendering process and the validity of the assumptions of each algorithm. We then choose the algorithm to use with Maximum a Posteriori estimation. We demonstrate the utility of our approach on recent IBR algorithms which use oversegmentation and are based on planar reprojection and shape-preserving warps respectively. Our algorithm selects the best rendering algorithm for each superpixel in a preprocessing step; at runtime our selective IBR uses this choice to achieve significant speedup at equivalent or better quality compared to previous algorithms.

1. Introduction

Image-based Rendering (IBR) is a powerful alternative to tedious modeling and expensive photo-realistic rendering in computer graphics. Automatic 3D reconstruction methods [16, 11] have encouraged the development of numerous new IBR algorithms [28, 9, 14, 5, 18, 20], which build on and improve the original methods where geometry was either not used [19] or provided (semi-) manually [8, 17, 4]. Recent IBR algorithms often treat specific cases very well, e.g. the floating textures algorithm [9] reduces ghosting, shape-preserving warps [5] allow plausible rendering of badly-reconstructed regions (low texture, vegetation) and gradient domain rendering [18] treats reflections. These methods typically sacrifice performance for quality to treat hard cases; in well reconstructed regions, simpler and faster methods [4] perform very well.

We see that each IBR algorithm has different quality/speed trade-offs, depending on the specific scene and cases it treats, and that no single algorithm is better than all others for all cases. In addition, each method has different parameters which directly affect rendering quality. Modeling such complex rendering processes to improve novel view synthesis is hard, due to the complexity of the solutions and the data, which are often uncertain (e.g., 3D reconstructions, camera calibration). We introduce a general Bayesian approach that models different IBR algorithms but also the possibility to *choose* between them. Bayesian methods have been used in IBR to improve image quality for specific algorithms [10, 21]. Our approach is complementary to these and allows the combined use of several different IBR algorithms by choosing between them in a local manner, i.e., at the level of image regions. We first present this approach in general terms which can be used in the context of several different algorithms. Our Bayesian methodology provides an intuitive description of the problem and takes the full set of complex factors into account. This formulation expresses the likelihood of a choice of rendering method by taking into account the rendering quality and the priors given the assumptions about the scene. We solve a Maximum a Posteriori (MAP) estimation to choose the rendering process at the granularity we target.

To demonstrate the utility of our framework, we apply our general approach to the class of IBR algorithms based on oversegmentation (superpixels). These achieve high rendered image quality by preserving silhouettes. In this algorithmic class, we use the algorithm of Zitnick et al. [28] as a baseline. It uses fronto-parallel depth to render superpixels and is thus fast. We also consider the recent algorithm of of Chaurasia et al. [5], which uses a shape-preserving warp to regularize rendering of superpixels in hard cases. This approach has been demonstrated to be superior in quality to previous methods especially for free-viewpoint navigation [5], but involves an expensive warping step during rendering. We also include an intermediate approach, which uses planar estimation of superpixels (similar to that of [3]) rendered with a method akin to the Unstructured Lumigraph [4].

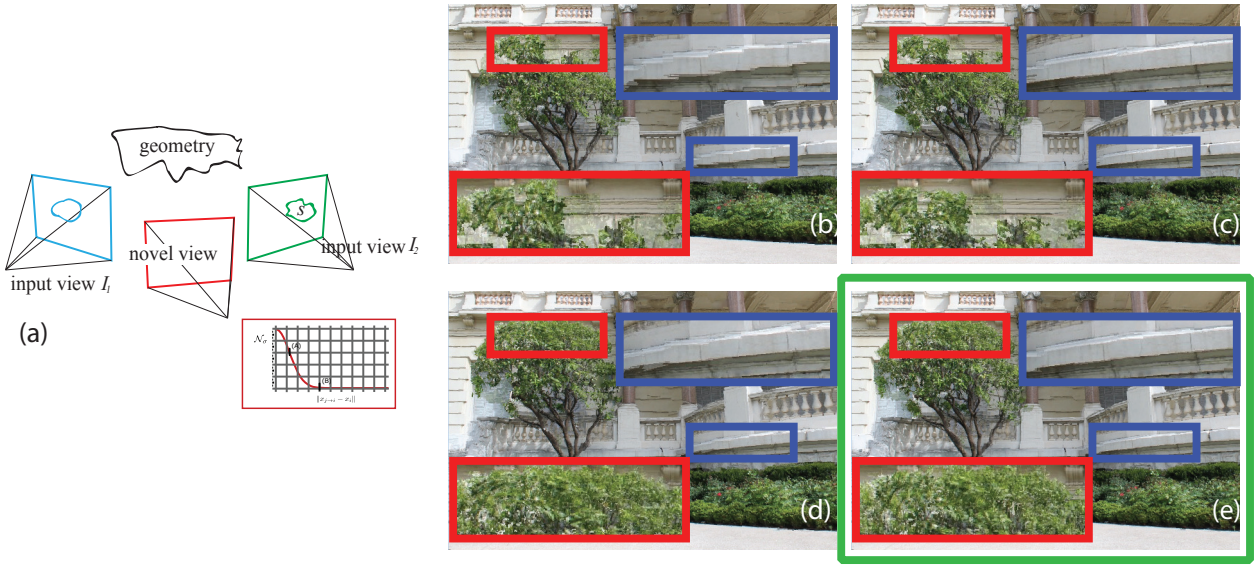

Figure 1. We propose a Bayesian formulation (a) to model rendering quality for different Image-Based Rendering (IBR) algorithms, and a Maximum a Posteriori estimation to select the algorithm producing the highest probability result for a given image region. We apply our algorithm to three IBR methods which use oversegmented input images, each having different speed/quality tradeoffs. In (b), we use planes fronto-parallel to the input view which fail for trees and slanted planes. Using local plane estimation (c) the result is improved, especially for slanted planes (blue box). Using the shape preserving warp (d) of [5], better results are achieved for the tree (red box), but the quality of the slanted planes is worse. Our algorithm (e) makes the correct choice locally, giving the best solution in each case.

In a preprocessing step, we estimate probability densities independently at each superpixel to allow real-time selective rendering at runtime. The probability density function representing rendering quality is expressed using both geometric and photometric errors in re-rendering existing input views. In preprocessing, the MAP estimation on the three possible rendering processes assigns the best choice to each superpixel based on our Bayesian formulation, and our selective IBR algorithm efficiently generates high-quality novel viewpoints in real-time accordingly.
Our main contributions are:

- A new Bayesian formulation to model the choice and quality of rendering algorithms for IBR.
- A selective IBR algorithm for oversegmentation-based methods that chooses the rendering method most suited for a given superpixel in a preprocessing step, allowing high-quality real-time rendering.

Our implementation shows that the selective rendering algorithm is much faster with equivalent or even better quality than the best of the three approaches taken separately.

2. Previous Work

Structure-from-Motion (SfM) [24] and multi-view stereo (MVS) methods [15, 11] are now sufficiently powerful to allow casual capture of scenes with a small number of cameras. These methods result in a typically sparse and approximate 3D reconstruction in the form of a point cloud and/or a mesh. SfM and MVS are preprocessing steps for many image based rendering methods that have to overcome errors and uncertainties in the reconstruction when generating images for novel viewpoints.

2.1. Image-Based Rendering

The first image-based rendering solutions used image interpolation [19] or depth/geometry [8, 17] to synthesize novel views, even in the context of unstructured capture [4]. In the context of controlled multi-camera setups, oversegmentation has been used to achieve high-quality rendering [28]. More recently, optical flow has been used to improve free-viewpoint image quality [9] or to perform video manipulation in the Videomesh system [7]. Epipolar constraints have also been used [14] to render unreconstructed regions. Parallax Photography [27] uses a carefully built mesh on the input images and soft visibility to improve new view rendering with hole filling.

The above methods depend on relatively good quality depth. When it is not available in all image regions, oversegmentation can be used together with depth synthesis and shape-preserving warp to allow navigation in regions far from the input cameras [5]. In this case, image warping requires solving a small system of linear equations for each superpixel. A hybrid rendering approach [20], computes expensive dense matches to perform hybrid depth reconstruction and finally renders by warping a dense grid of pixels. Specific reconstruction approaches [22] and a gradient-

based method [18] have been proposed to improve quality, in the hard case of reflective surfaces. The latter however incurs the additional expense of integrating the gradients in the last step to obtain the final image.

All methods try to overcome the limitations imposed by inaccurate and incomplete reconstruction. In urban scenes, many man-made structures exist and thus many image regions (superpixels) can be well approximated by local planes [23, 13]. In recent work, local planes have been assigned to superpixels from sparse SfM point clouds [3] with the idea of creating a light-weight approximate representation of 3D scenes for large-scale reconstructions.

In contrast to the above, we robustly identify regions that can be well rendered using planar approximations or image warps respectively, allowing the development of our selective IBR method which is more efficient and has equivalent or better quality than previous solutions.

2.2. Bayesian Methods for IBR

Previous work on Bayesian methods in IBR focus on how to estimate the final pixel color as opposed to our approach which estimates the best choice of rendering method as a preprocess. The work on image-based priors [10] posed IBR in a Bayesian setting, where novel view synthesis is described as finding the most likely new view given the input images. Using Bayes rule, the problem becomes the estimation of the photoconsistency likelihood and a texture prior. These are estimated using depth and color comparisons and by learning the texture prior from a dictionary of patches created from the input images. Follow-up work has investigated ways to accelerate this computation [25]. For the specific case of the Unstructured Lumigraph, recent work [21] systematically models depth uncertainty and sensor noise, resulting in the first Bayesian formulation of the blending heuristics originally presented by Buehler *et al.* [4].

In these previous Bayesian methods, costly optimizations are performed for each pixel in each novel viewpoint, excluding their use for real-time rendering. Our analysis shares some common tools with these approaches but our goal is real-time IBR. Our Bayesian approach (Sec. 3) includes terms describing rendering quality as well as the choice of real-time capable rendering methods and their parameters. We will concentrate here on modeling the *choice* of the rendering methods and their quality while assuming that method parameters are fixed. Our optimization is lightweight since it operates on superpixels and is estimated once in a pre-processing step. This allows us to introduce an efficient real-time IBR algorithm. In addition to speed, our algorithm inherits the quality of the methods we use.

3. Bayesian Formulation

We next introduce our Bayesian approach to model IBR and the choice between different rendering algorithms. Our final goal is to compute the best quality image, which we model as being *the most likely image* in a probabilistic sense [2]. The preprocessing step we describe next will assign a rendering algorithm to each superpixel of each image. At runtime each superpixel will be rendered using the chosen rendering algorithm.

3.1. A Bayesian Approach to IBR

We define a probabilistic model of the rendering function that generates novel images I. The rendering function is very general and corresponds to the set of three rendering methods we consider (i.e., [4, 29, 5]). These rendering methods are respectively characterized by the sets of parameters ξ_1, ξ_2 and ξ_3. These parameters represent all the necessary information needed by the rendering function to estimate images for new viewpoints. We define the label l_s^i that identifies which of the three rendering method is used for each superpixel s of the input views i.

Noting $\xi = \{\xi_1, \xi_2, \xi_3\}$ the set of all rendering parameters and L the vector of all labels l_i^s, we define the probability distribution $p(\xi, L|I)$ which expresses the likelihood of a choice L of rendering method with parameters ξ given images I. To estimate this distribution, we use a generative model [2] as we will explicitly model inputs (input images) and outputs (renderings). The model describes the method of rendering new viewpoints as follows:

$$p(\xi, L|I) = \frac{p(I|\xi, L)p(\xi)p(L)}{p(I)} \tag{1}$$

The denominator $p(I)$ is a normalization factor and since we will be maximizing likelihood, we can ignore it, leading to the simpler expression:

$$p(\xi, L|I) = p(I|\xi, L)p(\xi)p(L) \tag{2}$$

We define this model for rendering methods based on superpixels [4, 28, 5], but it can be applied to any rendering method. The selection can be defined for different parts of the process, e.g., image regions in input images, input or output camera positions, specific pixels, etc.

The general relation described by this generative model is illustrated in Fig. 2(a), along with the specific case of selective rendering (Fig. 2(b)).

Rendering quality. We model rendering quality with the term $p(I|\xi, L)$, which expresses the likelihood to generate images I given vectors of labels L and parameters ξ. High probability $p(I|\xi, L)$ means that the image I is close to the result obtained with the rendering method selected by the state variable L. In Sec. 5, we use this rendering quality to choose the rendering method for each superpixel of the input images.

Priors on rendering parameters. We consider the rendering methods used by the rendering function as black

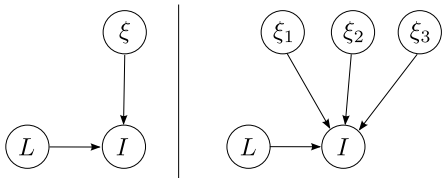

Figure 2. Probabilistic graphical models for selective IBR: (a) In the general model, the rendering of image I is estimated according to label L that indicates which rendering method to use with parameters ξ. (b) Selective rendering uses a set of 3 rendering methods. Each rendering method is described by its parameters ξ_1, ξ_2 and ξ_3. These parameters can be for example the number of superpixels. For superpixels, L is a vector of labels specifying which rendering method to use for each one of them.

boxes. The prior $p(\xi)$ is thus considered uniform and we do not need to further develop the list of parameters of each method. Our assumption is that, independently, each algorithm is close to optimal and our objective is to find the best way of combining them. However, if the goal is also to improve the rendering algorithm, such a PDF can be used to favor a certain set of parameters.

Prior on the choice of rendering method. The probability $p(L)$ is a prior on the choice of rendering algorithm. It does not depend on the resulting images but only on the selected method. We can use it for example to favor a specific rendering method when we expect it to perform better in a given context.

We now have a probabilistic framework for image based rendering that models the relation between different algorithms and the final rendered image. Instead of estimating the most probable image (which is time consuming [21]), this probabilistic model can be used to choose a real-time rendering method and its parameters in a pre-process.

3.2. Rendering selection as MAP estimation

Using the proposed generative model, we can express the selection of the rendering method L^* as a MAP estimation:

$$L^* = \underset{L}{\operatorname{argmax}}\, p(I|\xi, L)p(\xi)p(L) \tag{3}$$

Quality measures on the rendered images to estimate are used to select the rendering method L. These rendering methods are treated as black boxes and we do not impose any prior on their parameters, so $p(\xi)$ can be ignored in further development:

$$L^* = \underset{L}{\operatorname{argmax}}\, p(I|\xi, L)p(L). \tag{4}$$

To solve the above equation, we would ideally need to evaluate Eq. 4 over a large number of images I. Unfortunately, this is impossible since these images are not available. As an approximation, we can evaluate the density $p(I_i|\xi, L)p(L)$ for each input image I_i. We do this by rendering all *other* input images $\{I_1, \ldots, I_{i-1}, I_{i+1}, \ldots, I_n\}$ into the viewpoint of I_i, and evaluating how well the synthesized image matches the ground truth input I_i. This MAP estimation can thus be performed as a pre-process.

4. Superpixel IBR Algorithms

As explained previously, IBR methods based on oversegmentation achieve high quality by preserving silhouettes while maintaining real-time performance; they are thus suited to our objectives. In preprocessing, we use the proposed Bayesian model to choose the best method for each superpixel, allowing fast and high quality rendering at runtime. In what follows we assume that the input is a set of images from different viewpoints, processed by SfM and MVS. We thus assume that a set of reconstructed points $\mathcal{X}^s$ is assigned to each superpixel s of the oversegmentation.

4.1. Planar superpixels for IBR

The baseline algorithm we consider is that of Zitnick et al. [29] which oversegments the input images, and uses fronto-parallel depth for subsequent rendering. In our case for each superpixel s we use the median depth of the 3D points $\mathcal{X}^s$. We extend the original method by using depth values hallucinated by propagation from similar superpixels in the image [5] when a superpixel does not contain reconstructed geometry. Such superpixels are assumed to be fronto-parallel to the corresponding input camera. We call this algorithm fronto-parallel planar (FPLAN).

We also propose an intermediate algorithm that enhances this approach with a local plane estimation, similar in spirit to Bodis-Szomoru et al. [3]. We use RANSAC on the reconstructed points $\mathcal{X}^s$ to estimate the planes after some initial filtering of outliers as described in supplemental material. If a superpixel s is well approximated by a plane, we define a planar quadrilateral bounding the superpixel and transform the superpixel to the new view using a homography. We assume for now that the quadrilateral is a good approximation of the geometry corresponding to the superpixel since our probabilistic model will identify other cases as discussed below. Note that the actual rendering uses s as a mask and only renders pixels of the rendered quadrilateral which correspond to the region of s [5]. This algorithm can be seen as a combination of [29] and the Unstructured Lumigraph [4]. We call this algorithm planar reprojection (PLAN).

4.2. Superpixel warp

The highest quality oversegmentation-based IBR method we consider is shape-preserving warp [5]. For each input view, the shape-preserving warp algorithm (SWARP) takes as input the set of superpixels and the corresponding reconstructed 3D points. Rendering proceeds by building a small mesh of triangles over each superpixel s and

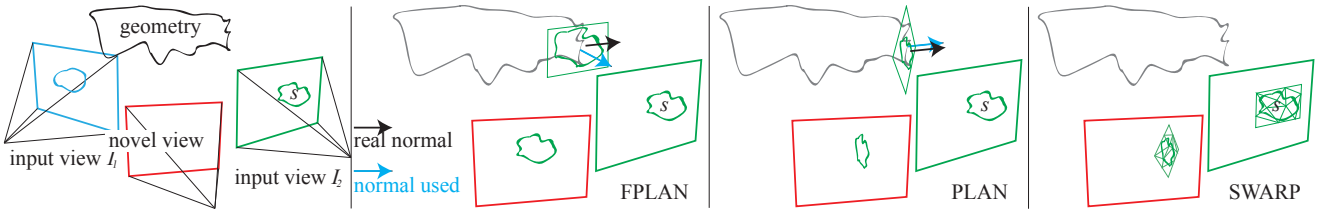

Figure 3. (a) The geometry, input cameras, and oversegmented images. (b) FPLAN: a frontoparallel plane is assigned to superpixel s. (c) PLAN: a plane is estimated for s. (d) SWARP: a shape preserving warp is applied to s in image space.

performing an image-space warp of the mesh into the novel view, using shape-preserving constraints and 3D reconstruction. This algorithm, though computationally expensive, handles poorly reconstructed regions and allows free-viewpoint navigation far from the input viewpoints.

If the reconstruction of the model corresponding to 3D space covered by the superpixel s is of high quality and if s is (almost) planar, the warp is wasteful since it will give essentially the same result as direct reprojection. However, when the quality of the reconstruction is uncertain or unknown, the shape-preserving constraints will dominate and provide a plausible solution in many cases.

5. MAP Estimation for Rendering Selection

We now have three rendering algorithms based on image over-segmentation, PLAN, FPLAN, and SWARP. In this section we show how the probabilistic framework presented in Section 3 can be used to select which rendering method should be considered for a given superpixel.

5.1. MAP selection at superpixel level

With the rendering methods precisely defined we can adapt the general formulation (Eq. 4) to our specific scenario. As already mentioned, the observations for the MAP estimation are the input images $\{I_1, I_2, \ldots, I_n\}$ and we can rewrite Eq. 4 as:

$$L^* = \underset{L}{\operatorname{argmax}} \prod_{i=1}^{n} p(I_i|\xi, L)p(L) \tag{5}$$

To find L^* the MAP estimate of the labels L, we need to evaluate $p(I_i|\xi, L)$. In this MAP estimation, we generate the image corresponding to the viewpoint of I_i using the images I_j ($j \neq i$). To do this, the images I_j are transformed to the viewpoint of I_i using depth and/or shape-preserving warps and blended together. If we note $\mathcal{R}_{j\to i}$ the *rendering* obtained by transforming I_j, then creating the approximation $\tilde{I}_i$ to the input image I_i can be expressed as:

$$\tilde{I}_i = \sum_{j\neq i} \alpha_j \mathcal{R}_{j\to i} \qquad \text{with} \sum_{j\neq i} \alpha_j = 1 \tag{6}$$

So $p(I_i|\xi, L)$, which models the error in rendering, can be expressed as a function of the distance between I_i and the transformed images $\mathcal{R}_{j\to i}$. With the assumption that improving any of these intermediate images improves the final blended image, we can write:

$$p(I_i|R, L) \propto \prod_{j\neq i} p(I_i|\mathcal{R}_{j\to i}, L_j) \tag{7}$$

The rendering methods reason on superpixels and novel viewpoints are generated by independently estimating superpixel transformations and blending them. Thereby, the choice of rendering algorithm must be made for each superpixel. The selection variable L_j is defined as the vector of labels l_j^s selecting the rendering method to use with each superpixel s from image I_j. We can now expand the expression for the MAP estimation:

$$L^* = \underset{L}{\operatorname{argmax}} \prod_{i=1}^{n} \prod_{j\neq i} p(I_i|\mathcal{R}_{j\to i}, L_j)p(L_j) \tag{8}$$

In our case the possible values for l_j^s are $\{PLAN, FPLAN, SWARP\}$. Assuming that rendering is independent between the superpixels, the MAP estimation becomes:

$$L^* = \underset{L}{\operatorname{argmax}} \prod_{i=1}^{n} \prod_{j\neq i} \prod_{s} p(I_i|\mathcal{R}^s_{j\to i}, l_j^s)p(l_j^s) \tag{9}$$

Maximizing the previous probability can be done independently for each superpixel and the MAP equation for each superpixel label is thus:

$$l_j^{s,*} = \underset{l_j^s}{\operatorname{argmax}} \prod_{i\neq j} p(I_i|\mathcal{R}^s_{j\to i}, l_j^s)p(l_j^s) \tag{10}$$

This equation allows the selection of the rendering algorithm. Note that starting from the general Eq. 4 and by leveraging rendering algorithm properties, we derive a model that expresses the same ideas at the level of superpixels. In this case, $p(I_i|\mathcal{R}^s_{j\to i}, l_j^s)$ expresses the quality of rendering superpixel I_j^s in the different view i using the rendering algorithm l_j^s. The probability $p(l_j^s)$ is the prior on the choice of rendering algorithm l_j^s and is considered uniform over all the labels. In the following, we show how using only rendering quality for superpixels we are able to perform algorithm selection for rendering.

5.2. MAP selection using rendering quality

We model the probability distribution $p(I_i|\mathcal{R}^s_{j\to i}, l^s_j)$ as a function of the distance between the transformed image $\mathcal{R}^s_{j\to i}$ and the observed input I_i image, using two distributions:

$$p(I_i|\mathcal{R}^s_{j\to i}, l^s_j) = p_{\text{geom}}(I_i|\mathcal{R}^s_{j\to i}, l^s_j) p_{\text{pho}}(I_i|\mathcal{R}^s_{j\to i}, l^s_j). \tag{11}$$

The first term corresponds to the geometric rendering quality. It expresses how well the 3D structure of the scene is preserved under the rendering transformation. The second distribution is based on appearance and will be referred to as photometric rendering quality. It models the error between the rendered image and the observation in terms of color differences. We also use occlusion information from MVS reconstruction estimating rendering quality only in viewpoints where the superpixel is visible.

Geometric rendering quality. To render the image at the view of input image I_i, the superpixel s will undergo a transformation corresponding to a warp (for $l^s_j = $ *SWARP*) or a plane projection (for $l^s_j =$ *PLAN* or *FPLAN*). One way to measure the error in this transformation from a geometric point of view is to use reconstructed 3D points present in the superpixel.

We define $\mathcal{X}^s_j$ as the set of the 3D reconstructed points X that project in the superpixel s in view j. We denote x_j the 2D position of the projection of X in view j. As previously described, the superpixel s undergoes a transformation to the viewpoint of an input camera. The points x_j will follow the same transformation and their new position is noted $x_{j\to i}$. If the transformation is well estimated, then $x_{j\to i}$ and x_i (the projection of X in view i) should coincide. To define the geometric term, we use a Gaussian distribution defined on the distance between $x_{j\to i}$ and x_i:

$$p_{\text{geom}}(I_i|\mathcal{R}^s_{j\to i}, l^s_j) = \prod_{X\in\mathcal{X}^s_j} \mathcal{N}_\sigma\left(\frac{\|x_{j\to i} - x_i\|}{|\mathcal{X}^s_j|}\right) \tag{12}$$

If there are no reconstructed points, it is impossible to estimate a plane and so p_{geom} is set to zero for *PLAN*. For *FPLAN* and *SWARP* depth will be propagated from neighbors. The choice between these two labels will only depend on p_{pho}.

Photometric rendering quality. The objective is to estimate the rendering quality in terms of appearance. We denote $s_{j\to i}$ the result of transforming the superpixel s to the image plane of camera C_i. To measure the rendering quality in terms of appearance, we use the mean squared distance (MSE) between the pixel colors of I^s_j and $I_i^{s_{j\to i}}$. If the transformation is well estimated, the distance should be small. To define the photometric term, we use a Gaussian distribution defined on the mean squared distances between I^s_j and $I_i^{s_{j\to i}}$:

$$p_{\text{photo}}(I_i|\mathcal{R}^s_{j\to i}, l^s_j) = \mathcal{N}_{\sigma_2}(\text{MSE}(I^s_j, I_i^{s_{j\to i}})) \tag{13}$$

We note that other error measures could be considered but this was sufficient in our case.

Final labeling. To obtain a fast rendering algorithm we need to favor plane projection methods (*PLAN* and *FPLAN*) when they result in similar quality to the warp based approach. To this end we use a smaller value for σ_2 in the case of *PLAN* and *FPLAN* labels. Thanks to this, when both planar and warp based methods achieve good results, the planar rendering will be favored, resulting in important speedup. We can now compute Eq. 10 for each superpixel of each image, for each of *PLAN, FPLAN*, and *SWARP*. We discuss below how we use this estimation in a preprocessing step for our selective rendering algorithm.

6. A Selective IBR Algorithm

The input to our approach is a set of images of a given scene, which have been processed by automatic calibration (e.g., VisualSFM [26]) and MVS reconstruction (e.g., [12]). These two steps provide camera calibration parameters, and a 3D point cloud of the scene, which can be sparse and inaccurate in regions with low texture or stochastic (e.g., vegetation) or reflective (e.g., cars) content.

Preprocessing. For each image, we first run the superpixel oversegmentation of [1] and the depth synthesis as described in [5]. We then perform the plane estimation for each superpixel, as described in Sec. 4.1. In a preprocessing step, we perform the MAP estimation on this data, following Eq. 5. This is done for each superpixel of each input image and each rendering algorithm, i.e. $L = \{$*PLAN,FPLAN,SWARP*$\}$. A rendering algorithm is then chosen for each superpixel in this preprocess.

Rendering. Similarly to Chaurasia et al. [5], the four spatially closest views to the novel view are chosen and each superpixel of these views is projected into the novel view. In contrast to previous methods, each superpixel is projected into the novel view using the choice of rendering algorithm l^s_j, as computed in the pre-processing step.

The projection operation for *FPLAN,PLAN* uses standard OpenGL polygon rendering in the GPU, and is much cheaper than the superpixel warp. We measured a factor of approximately 3 times speedup, depending on the number of MVS points in each superpixel which add constraints to the warp. Speedup depends on the percentage of superpixels using the *SWARP*, as shown in the results.

Implementation Details The preprocessing step and rendering were implemented in C++ with OpenGL/GLSL shaders. For *SWARP* a triangle mesh covering superpixels

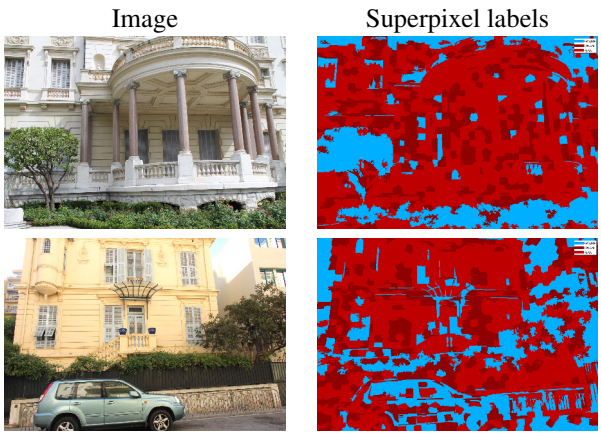

Figure 4. Selection of the rendering algorithm at the superpixel level. Superpixels in dark and medium red are rendered using planes with respectively the *FPLAN* and *PLAN* algorithm. When the *SWARP* algorithm is selected the superpixels are in blue. This last label is mainly used in regions with poor or non existing 3D information such as leaves and specular car windows.

is warped [5]. For Eq. 12 we use barycentric coordinates of the mesh triangle before the warp to determine their position in the same triangle after the warp. For Eq. 13 every rasterized patch is read back in RGB color space. This requires 2 min to process an image of 1M pixels. Instead of reading back, computing in GPU the values of Eq. 13 would reduce to few seconds.

7. Results and Comparisons

We ran our algorithm on six different scenes, shown in Figs. 4, 5 and 6. There are five scenes taken from previous work [6, 5] (Yellowhouse-12, Museum-27, Street-10, Tree-18, Aquarium-20) and a new scene called House-25 (Fig. 5). The sufix in the name of a dataset indicates the number of images in it.

The main goal of our approach is to choose the most appropriate rendering process according to quality criteria. This limits the usage of expensive computations for image-space warps [5] to regions with poor 3D information and favors planar approximation for the superpixels where its rendering quality is high. In Fig. 4 we can see an illustration of the selected rendering method for superpixels of the three datasets. The planar approximation is mostly used on buildings where 3D reconstruction is most reliable. In more challenging parts of the scene, the image-space warps are more likely to be used. This is the case for leaves where geometry is not necessarily well approximated by a plane. Due to reflections, windows are also not well reconstructed and we notice a higher proportion of superpixels of *SWARP* labels selected (in blue). Table 1 shows the percentages of superpixels classified in *FPLAN,PLAN,SWARP* on average (with standard deviation) for each dataset. Planar approximations are used on average for 78% of superpixels allowing our algorithm to run 2.5 times faster (mean value) than *SWARP*. We ran the House scene test on a 12-core 2.5GHz Dell Z800 (NVIDIA Titan GTX GPU); all others on a 6-core Dell 3.2GHz Z420 (GTX 680). After MVS reconstruction, the whole preprocess takes about 3min/image. Warps are parallelized, explaining the difference in overhead of our approach compared to planar methods in the two configurations. At rendering time, the cost of choosing the four nearest neighbors is negligible.

Scene	*FPLAN*	*PLAN*	*SWARP*
Yellowhouse-12	36.62 ± 5.84	39.45 ± 5.88	23.93 ± 7.87
Street-10	35.30 ± 6.03	$38,47 \pm 5.12$	26.23 ± 6.62
Museum-27	31.52 ± 3.12	55.5 ± 3.53	12.98 ± 1.30
Tree-18	38.67 ± 3.67	30.24 ± 6.23	31.09 ± 5.28
Aquarium-20	34.02 ± 4.94	56.75 ± 5.23	9.23 ± 1.93
House-25	38.87 ± 5.47	37.65 ± 5.01	23.48 ± 5.99

Table 1. Average (standard deviation) percent of the Bayesian preprocessing phase. The percentage of superpixels requiring a warp is low on average.

Scene	Speedup	*Selective*	*SWARP*	*F/PLAN*
Yellowhouse-12	**2.5**	145.7	58.7	346.0
Street-10	**2.5**	158.5	62.5	373.5
Museum-27	**2.9**	158.3	55.3	319.3
Tree-18	**2.2**	136.5	62.5	418.3
Aquarium-20	**3.5**	218.0	62.5	314.3
House-25	**2.4**	97.0	41	102

Table 2. FPS for each algorithm and our selective approach. The speed up factor is relative to the *SWARP* method.

Figure 6. Comparison of the proposed selective method (on the right) and the dense correspondences for rendering [20] (on the left) for a given position on the view interpolation path. We note that the rendering based on dense correspondences has the typical artifcats due to a bad estimation of correspondences (see close ups). In regions with poor 3D information (building of the left) both methods show rendering artifacts.

7.1. Comparisons

The main advantage of our method is speed, since it only uses shape-preserving warps when necessary. Our selective IBR is on averge 2.5 times faster than *SWARP*, reaching 3.5 times for the Aquarium-20 scene (Table 2). We show frames per second (FPS) for each algorithm.

In the following we compare our approach with the two

Figure 5. House, Yellowhouse, Street, Aquarium and Tree scenes for *FPLAN,PLAN,SWARP* and our algorithm, left to right.

recent IBR methods [5, 20]. These approaches have already shown their superiority [5, 20] over methods based on optical flow estimation [9], epipolar constraints [14] or manually defined silhouettes [6].

In figure 6, we show a visual comparison with the dense correspondence approach of Lipski *et al.*[20], notably the rendered image for a given position on the view interpolation path. Overall visual quality is close, although our methods avoids some of the blurring due to correspondence tracking.

Quality evaluation is subjective, especially for the complex imagery we consider here. From visual inspection of interactive sessions, using different navigation paths for the various datasets, our approach globally outperforms the other superpixel based algorithms. To illustrate this, Fig. 5 shows a selection of challenging viewpoints, off the view-interpolation trajectory. Each time the proposed selective approach results in rendering quality equivalent or better than the three methods taken separately. This is more obvious in the accompanying video, where artifacts (e.g., incorrect plane reconstruction) become particularly visible during camera motion. The choice of *SWARP* for unreconstructed regions results in improved overall visual quality compared to *PLAN,FPLAN* albeit with an increase in computational overhead, depending on the CPU used.

8. Conclusions and Discussions

We proposed a Bayesian formulation to model the choice of the most suitable rendering method for IBR algorithms based in superpixels, using probability distributions to model rendering quality and choice of rendering method. We solve for the most suitable rendering method using MAP estimation, which a rendering method for each superpixel as a preprocess. We use the result to define a selective IBR algorithm combining the benefits of previous algorithms, with a very good overall speed/quality tradeoff. One important strength of our approach is that it identifies regions of the image where using the more expensive IBR approaches (e.g., [5]) is wasteful, and replaces it with a cheaper planar reprojection method of equivalent quality.

We currently use the camera selection and blending of Chaurasia et al. [5]. These can definitely be improved, but both topics are hard problems involving different tradeoffs which we will investigate in future work. A good solution will improve quality of our algorithm significantly.

This work provides a first indication on the utility and power of MAP estimation as a preprocess for real-time IBR. We plan to pursue these ideas further in the more general context taking the prior $p(\xi)$ into account, improving the rendering methods and their parameters. Developing such solutions raises several hard challenges, including a way to estimate quality of IBR in the absence of a reference and preferably online. Another important issue is the balance between preprocessing and runtime: optimization per pixel at rendering time is prohibitively expensive, but some combination of preprocessing and well-designed GPU data structures could result in significant improvements in rendering quality using an extension of our Bayesian approach.

9. Acknowledgments

Research funded by EU FP7 project 611089 CR-PLAY and French ANR project SEMAPOLIS (ANR-13-CORD-0003). Thanks to J. Esnault for software support and G. Chaurasia, G. Brostow and M. Goesele for proofreading.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. PAMI*, 2012.

[2] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[3] A. Bodis-Szomoru, H. Riemenschneider, and L. V. Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *CVPR*, 2014.

[4] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, 2001.

[5] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. on Graphics (TOG)*, 2013.

[6] G. Chaurasia, O. Sorkine, and G. Drettakis. Silhouette-aware warping for image-based rendering. *Comp. Graph. Forum*, 2011.

[7] J. Chen, S. Paris, J. Wang, W. Matusik, M. Cohen, and F. Durand. The video mesh: A data structure for image-based three-dimensional video editing. In *ICCP*, 2011.

[8] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *SIGGRAPH*, 1996.

[9] M. Eisemann, B. D. Decker, M. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating textures. *Comp. Graph. Forum*, 2008.

[10] A. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. *IJCV*, 2005.

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *CVPR*, 2007.

[12] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. PAMI*, 2010.

[13] D. Gallup, J.-M. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *CVPR*, 2010.

[14] M. Goesele, J. Ackermann, S. Fuhrmann, C. Haubold, and R. Klowsky. Ambient point clouds for view interpolation. *ACM Trans. Graph.*, 2010.

[15] M. Goesele, B. Curless, and S. M. Seitz. Multi-view stereo revisited. In *CVPR*, 2006.

[16] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007.

[17] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The Lumigraph. In *SIGGRAPH*, 1996.

[18] J. Kopf, F. Langguth, D. Scharstein, R. Szeliski, and M. Goesele. Image-based rendering in the gradient domain. *ACM Trans. Graph.*, 2013.

[19] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH*, 1996.

[20] C. Lipski, F. Klose, and M. Magnor. Correspondence and depth-image based rendering: a hybrid approach for free-viewpoint video. *IEEE T-CSVT*, 2014.

[21] S. Pujades, F. Devernay, and B. Goldluecke. Bayesian view synthesis and image-based rendering principles. In *CVPR*, 2014.

[22] S. N. Sinha, J. Kopf, M. Goesele, D. Scharstein, and R. Szeliski. Image-based rendering for scenes with reflections. *ACM Trans. Graph.*, 2012.

[23] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, 2009.

[24] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph.*, 2006.

[25] O. Woodford and A. W. Fitzgibbon. Fast image-based rendering using hierarchical image-based priors. In *BMVC*, 2005.

[26] C. Wu. Towards linear-time incremental structure from motion. In *3DV*, 2013.

[27] K. C. Zheng, A. Colburn, A. Agarwala, M. Agrawala, D. Salesin, B. Curless, and M. F. Cohen. Parallax photography: creating 3d cinematic effects from stills. In *Proc. Graph. Interface*, 2009.

[28] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 2007.

[29] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 2004.