Depth Synthesis and Local Warps for Plausible Image-based Navigation

Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, George Drettakis

Depth Synthesis and Local Warps for Plausible Image-based Navigation

GAURAV CHAURASIA and SYLVAIN DUCHENE
REVES/INRIA Sophia Antipolis
and
OLGA SORKINE-HORNUNG
ETH Zurich
and
GEORGE DRETTAKIS
REVES/INRIA Sophia Antipolis

Modern camera calibration and multiview stereo techniques enable users to smoothly navigate between different views of a scene captured using standard cameras. The underlying automatic 3D reconstruction methods work well for buildings and regular structures but often fail on vegetation, vehicles and other complex geometry present in everyday urban scenes. Consequently, missing depth information makes image-based rendering (IBR) for such scenes very challenging. Our goal is to provide *plausible* free-viewpoint navigation for such datasets. To do this, we introduce a new IBR algorithm that is robust to missing or unreliable geometry, providing plausible novel views even in regions quite far from the input camera positions. We first oversegment the input images, creating superpixels of homogeneous color content which often tends to preserve depth discontinuities. We then introduce a *depth-synthesis* approach for poorly reconstructed regions based on a graph structure on the oversegmentation and appropriate traversal of the graph. The superpixels augmented with synthesized depth allow us to define a local shape-preserving warp which compensates for inaccurate depth. Our rendering algorithm blends the warped images, and generates plausible image-based novel views for our challenging target scenes. Our results demonstrate novel view synthesis in real time for multiple challenging scenes with significant depth complexity, providing a convincing immersive navigation experience.

This work was partially funded by the EU IP project VERVE (www.verveconsortium.org); additional funding was provided by Autodesk, Adobe (research and software donations) and NVIDIA (professor partnership program).
Authors' addresses: G. Chaurasia, S. Duchene REVES/INRIA Sophia Antipolis; O. Sorkine-Hornung ETH Zurich; G. Drettakis (george.drettakis@inria.fr).

DOI 10.1145/2487228.2487238
http://doi.acm.org/10.1145/2487228.2487238

Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation—*Viewing algorithms*

Additional Key Words and Phrases: Image-based rendering, image warp, superpixels, variational warp, wide baseline, multiview stereo

1. INTRODUCTION

Recent advances in automatic camera calibration [Snavely et al. 2006] and multiview stereo [Goesele et al. 2007; Furukawa and Ponce 2009] have resulted in several novel applications. As an example, using a small set of photographs of a given scene captured with standard cameras from several viewpoints, methods such as [Eisemann et al. 2008; Goesele et al. 2010] enable smooth transitions between the different views. The underlying 3D reconstruction methods work remarkably well for buildings and regular structures; however, in everyday scenes containing vegetation and other complex geometry, there are always regions which do not have reliable or dense 3D information. Image-based rendering (IBR) for such scenes is thus very challenging.

Our goal is to provide *plausible* free-viewpoint navigation for such casually captured multiview datasets, which contain poorly and/or sparsely reconstructed regions when using state-of-the-art multiview stereo (e.g., [Goesele et al. 2007; Furukawa and Ponce 2009]). To do this, we introduce a new image-based rendering algorithm that is robust to missing and/or unreliable geometry and which provides *plausible* novel views, even in regions quite far from the input camera positions. We introduce a *depth-synthesis* approach for poorly reconstructed regions and a *local* shape-preserving warp to achieve this goal.

Recent IBR approaches try to compensate for poor 3D information. Methods which depend upon a 3D geometric model or proxy, result in significant ghosting or misalignment artifacts, even when compensating for poor 3D with optical flow [Eisemann et al. 2008]. Non-photorealistic rendering (NPR) styles [Goesele et al. 2010] are very effective for view transitions in photo collections, providing a satisfactory solution for different lighting conditions or dynamic scene content, such as moving people. Despite these advantages such methods fall short of our goal of providing plausible novel views for free-viewpoint navigation. The silhouette-based warp of [Chaurasia et al. 2011] can address our target scenes; however it requires manual pre-processing of the input images, limiting its applicability.

The main challenge in achieving our goals is treating poorly reconstructed depth in many regions of each image and the lack of precise silhouettes for the datasets we consider.

Our main contributions are (i) a depth-synthesis algorithm which provides depth samples in regions with poor depth, and (ii) a *local* shape-preserving warp and rendering algorithm that provides plausible novel views. We build our solution on an oversegmentation [Achanta et al. 2012] of input images. Superpixels provide homogeneous image regions and preserve depth discontinuities: the motivation for this choice is that superpixels allow our algorithm to both identify regions requiring depth synthesis and to find appropriate depth exemplars. In addition, superpixels correspond to homogeneous image content, and thus maximize the benefit of our local shape-preserving warp during rendering. Our synthesized depth is not necessarily photoconsistent: however, it is *plausible* and thanks to the shape-preserving warps, our method also produces *plausible* novel views, even when the user is far from the input cameras.

We have applied our approach to 12 different scenes (see Figure 7), including one from Microsoft Photosynth, two from [Pollefeys et al. 2008] and three from [Chaurasia et al. 2011]. In the accompanying video and supplemental material, we demonstrate recorded interactive navigation sessions for all scenes, which show that our method provides plausible novel view synthesis in real time, resulting in very convincing immersive navigation.

2. PREVIOUS WORK

Image-based rendering. Since the early work on plenoptic modeling [McMillan and Bishop 1995], many image-based rendering algorithms have been developed, such as light fields [Levoy and Hanrahan 1996] and unstructured lumigraphs [Buehler et al. 2001] among many others. Other interesting applications have resulted from this work, e.g., camera stabilization [Liu et al. 2009], video enhancement [Gupta et al. 2009] or commercial products like Google Street View.

Image interpolation approaches, e.g., [Mahajan et al. 2009; Lipski et al. 2010; Stich et al. 2011] have recently received attention, with impressive results. We do not discuss them further, since we concentrate on wide-baseline input datasets and free-viewpoint navigation. Most wide-baseline IBR techniques, e.g., [Debevec et al. 1996; Buehler et al. 2001; Eisemann et al. 2008] use geometric proxies to re-project input images to novel views. Regions with poor reconstruction result in a poor proxy, and significant visual artifacts in rendering. Ambient Point Clouds [Goesele et al. 2010] use a non-photorealistic rendering style in poorly reconstructed regions, and are restricted to view interpolation. In contrast, our depth-synthesis approach, coupled with our local shape preserving warp provides plausible free-viewpoint navigation.

Recent approaches have used variational warps guided by sparse multiview stereo point clouds to warp images to novel views. Liu et al. [2009] used 3D points to warp video frames to novel camera positions for video stabilization. Chaurasia et al. [2011] use a similar approach for wide-baseline IBR and handle occlusions by incorporating hand-marked silhouettes in the variational warp. Manual silhouette annotation is a major drawback of this approach; however the method demonstrates that shape-preserving warps can produce plausible novel views. In contrast our approach is completely automatic and the *local* nature of our warp improves quality. We compare to [Eisemann et al. 2008], [Goesele et al. 2010] and [Chaurasia et al. 2011] in Sec. 6.2.

A significant part of IBR research has concentrated on more restricted and controlled settings than ours, typically involving stereo rigs or other specialized (mostly) indoor capture setups. As a result, these methods do not have the kind of poorly reconstructed regions we encounter in our datasets. Co-segmentation techniques like [Zitnick et al. 2005; Bleyer et al. 2011] require dense capture while [Kowdle et al. 2012] handle a single object-of-interest only. These have not been shown on wide-baseline multiview unordered photo collections with multiple foreground objects, which we focus on here. Over-segmentation has been used to enforce silhouettes in depth maps [Zitnick and Kang 2007]. Other applications of oversegmentation include view-interpolation [Stich et al. 2011], depth estimation [Cigla et al. 2007], improving depth of man-made structures [Mičušík and Košecká 2010] etc. We use superpixels for depth-synthesis, local warping and adaptive blending. In contrast to previous methods which assume good depth, superpixels allow us to delineate regions with unreliable or poor depth, helping our depth synthesis. The coherent image regions and silhouettes provided by superpixels also help our local warp.

3D reconstruction and depth propagation. Multi-view stereo [Goesele et al. 2007; Furukawa and Ponce 2009]) can reconstruct reasonable point clouds for many scenes. We target scenes which are captured with a simple digital camera, in a casual manner, rather than methods which require specific camera rigs (e.g., the 8-camera setup in [Zitnick et al. 2004]) suitable for the use of stereo algorithms. Please refer to [Seitz et al. 2006] for an excellent overview.

Modern multiview stereo [Furukawa and Ponce 2009; Pollefeys et al. 2008], together with recent improvements (e.g., [Gallup et al. 2010]) provide the best results for the scenes we consider. Furukawa et al. [2009] typically reconstruct 100k-200k pixels from 5-6 megapixel images i.e., around 2% in our tests. Moreover, the distribution of depth samples is highly irregular, sparse and/or erroneous near silhouettes. Reconstructed 3D points or depth maps can then be merged using surface reconstruction [Kazhdan et al. 2006; Fuhrmann and Goesele 2011] and used as "proxies" for IBR.

The above methods typically rely on optimizing photo-consistency which becomes challenging for texture-poor surfaces, complex (dis)occlusions (e.g. leaves), non-lambertian surfaces etc. They give excellent results on closed objects [Sinha et al. 2007], but irregular objects such as trees are often poorly reconstructed or even completely missed (see the examples in [Gallup et al. 2010], which are similar to our target scenes). Objects of this kind do, however, appear frequently in everyday (sub)urban scenes. By synthesizing depth in such poorly reconstructed regions, we enable plausible interactive image-based navigation.

Dense depth maps can be generated by propagating depth samples to unreconstructed pixels of the image. [Hawe et al. 2011] show that dense disparity maps can be reconstructed from a sparse sub-sampling given high density of depth samples near silhouettes. [Yang et al. 2007; Dolson et al. 2010] create pixel-dense disparity maps from the dense and regularly spaced disparity samples provided by range scans, but the method is not appropriate in our setting. Within the context of multiview stereo, piecewise-planar reconstruction has been presented in [Sinha et al. 2009; Gallup et al. 2010]. Similarly, [Furukawa et al. 2009] fit planes to the 3D point cloud to generate complete depth maps, giving impressive results for structured and planar regions like façades. However, rich (sub)urban scenes often deviate from planar priors because of the presence of vegetation, cars, etc. making these methods less effective for our scenes.

Goesele et al. [2010] use image-based depth interpolation without photo-consistency, which can lead to over-smooth depth maps and silhouette flattening. They use a graph-cut to retain interpolated depth only in regions with a high density of depth samples, while significant regions with sparse or irregular depth are discarded and instead rendered in an NPR style. Our depth synthesis allows plausible IBR even for such problematic regions.

3. OVERVIEW AND MOTIVATION

Our approach has two main steps: a *depth-synthesis* pre-processing step and a *local* shape preserving warp, followed by a three-pass rendering algorithm. Our input is a set of images of a scene, taken from multiple viewpoints. We first extract camera matrices using Bundler [Snavely et al. 2006] and use multiview stereo [Furukawa and Ponce 2009] to reconstruct a 3D point cloud of the scene. We project these 3D points into the images, providing a set of projected depth samples in each image. We then oversegment [Achanta et al. 2012] all the input images creating superpixels that denote regions of homogeneous color content and preserve depth discontinuities. We assume the best reconstruction and segmentation techniques are used. Our approach is independent of the choice of reconstruction and segmentation approaches.

Depth-synthesis. The key motivation for this step is that even after using the best reconstruction, there can be significant regions with no depth. Instead of discarding such regions, we synthesize *plausible* depth suitable for IBR walkthroughs, which is not necessarily photoconsistent. The oversegmentation and projected depth allow us to identify poorly reconstructed superpixels in each image. Depth-synthesis fills in poorly reconstructed superpixels using depth from "similar" superpixels of the image; we *do not* augment the 3D reconstruction. We create a graph structure with superpixels as nodes and define a careful traversal of the graph which allows us to identify best matching superpixels in terms of color and spatial proximity. We keep the three best matching superpixels and interpolate the depth from these superpixels to add a small set of new depth values into the original poorly reconstructed superpixel. These best matches are generally not immediate spatial neighbors; our depth synthesis thus performs *non-local* interpolation which maintains depth discontinuities provided by the superpixel representation.

Local Shape-Preserving Warp and Rendering. Superpixels now contain reconstructed depth from multiview stereo or *plausible* synthesized depth. The depth samples may not be photoconsistent; reprojecting them will lead to visible artifacts in rendering. To allow plausible novel views, we perform a *local* shape-preserving warp on each superpixel individually, in contrast to previous methods [Liu et al. 2009; Chaurasia et al. 2011] which warp the entire image. Superpixels correspond to well-defined regions of homogeneous color content, and thus give good results with our local shape-preserving warp.

Rendering is achieved with a three-pass blending algorithm. We first select four input cameras closest to the novel camera, and warp these images to the target view. The four warped images are then blended, with weights specified by camera orientation but also the reliability of depth information in each warped superpixel. Finally, we fill holes with Poisson blending [Pérez et al. 2003].

We present an extensive set of example scenes, all containing challenging regions which state-of-the-art multiview stereo reconstructs poorly. Our algorithm allows plausible navigation for all these scenes. We also compare to the three most relevant recent IBR algorithms [Eisemann et al. 2008; Goesele et al. 2010; Chaurasia et al. 2011]. Our approach diminishes many of the artifacts of these methods, and provides very convincing IBR navigation experiences, as can be seen in the accompanying videos.

4. DEPTH SYNTHESIS ALGORITHM

Our input is a set of images of a given scene, taken from different viewpoints. After 3D reconstruction, we use [Achanta et al. 2012] to oversegment each input image, an efficient algorithm that gives superpixels of approximately equal size and with regular shapes (see Figure 1(b)), unlike [Felzenszwalb and Huttenlocher 2004] which gives superpixels of highly irregular shapes and sizes due to lack to compactness constraints.

We denote the set of all superpixels in an image by $\mathcal{S} = \{S_i\}_{i\in\{0\ldots n-1\}}$. We project the reconstructed 3D points into the image, such that the depth at pixel $\mathbf{x}$ is denoted by $D[\mathbf{x}]$ (shown in Figure 1(c)). The set of depth samples inside each superpixel is thus $\mathcal{D}[S_i] = \{\mathbf{x} \in S_i \mid D[\mathbf{x}] > 0\}$. We distinguish two classes of superpixels: those containing less than 0.5% reconstructed pixels, which we call *target superpixels* (shown in green in Figure 1(d)) and all others which we consider to have reliable depth.

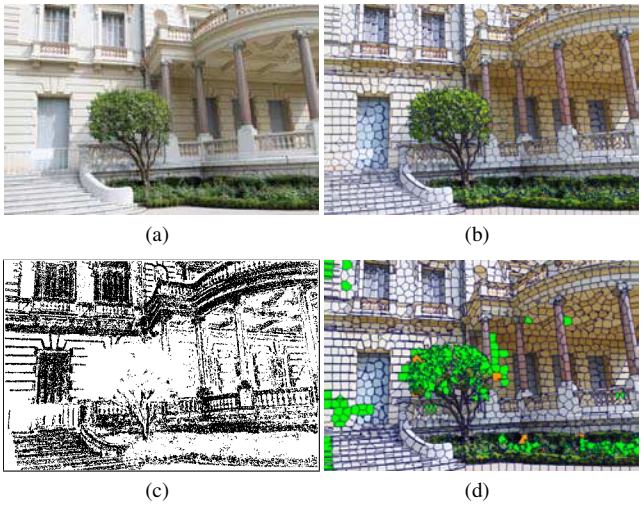


Fig. 1. (a) Input image, (b) superpixel oversegmentation, (c) projected depth samples, and (d) *target superpixels* marked in green. The superpixels marked in orange could not be assigned depth reliably by our depth synthesis step (Sec. 4.1.2). These are marked as holes.

4.1 Approach

Our goal is to synthesize plausible depth for a sufficient number of points in each target superpixel. We do this by identifying a set of *source superpixels*, which are spatially close and should ideally belong to the same object in the scene as that of the target superpixel. In addition, our goal is to have a fully automatic algorithm which requires no scene dependent parameter tuning.

There are several ways to achieve this goal; two seemingly straightforward approaches include object classification and interpolation/upsampling of existing depth. Object classification approaches [Andreetto et al. 2008] give remarkable results on some classes of objects, such as man-made structures, animals, humans, etc. However, for cluttered scenes such as ours, which often include vegetation, results can be less reliable. In addition, our experiments with e.g., [Andreetto et al. 2008] indicate very high computation times. Please refer to [Chaurasia et al. 2011] for experiments with state of the art segmentation algorithms.

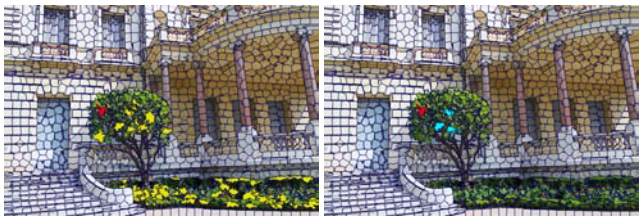
Interpolation techniques have been used for regions with sufficient depth density (e.g., [Goesele et al. 2010]). For the regions with very sparse depth, these techniques result in silhouette flattening and over smooth depth maps which diminish parallax effects during rendering.

We propose an efficient and robust approach which combines image-content similarity and spatial proximity in the choice of source superpixels employed to synthesize depth. The irregular shape of superpixel boundaries requires definition of appropriate distance metrics and search strategies both for image content and for spatial proximity. We use histogram comparison to identify superpixels with similar image content and a graph traversal approach to provide a robust and parameter-free algorithm. Depth values within target superpixels are synthesized using an interpolation approach based on the distribution of depths in the source superpixels.

4.1.1 *Computing similar superpixels.* We first compute a set of "most similar" superpixels for each target superpixel. Among many similarity metrics for measuring the affinity of irregular image regions, Grundmann et al. [2010] have successfully used χ^2 distance between LAB histograms of superpixels in order to measure color similarity. Other metrics like sum of squared differences (SSD) are less suitable for irregular shapes and sizes of superpixels. Measuring average color of a superpixel performed worse than LAB histogram distance. Inspired by the approach of Grundmann et. al [2010], we convert the image into LAB space and create separate histograms for each superpixel with 20 bins in each of L, A and B axes. We concatenate the histograms to give a 60D descriptor $\mathcal{H}_{Lab}[S_i]$ for each superpixel $S_i \in \mathcal{S}$. We compute the nearest neighbors of each target superpixel from all superpixels already containing depth samples using the histogram descriptors space with χ^2 distance metric. This gives a set of "most similar" superpixels $\mathcal{N}[S_i]$. We keep the 40 most similar superpixels, shown in yellow in Figure 2(a) for the target superpixel shown in red. We assume that any significant object would be around 5% of image area, equivalent to 40-60 superpixels. We experimented successfully with 40-80 most similar superpixels; higher numbers needlessly increased computation.

4.1.2 *Shortest walk algorithm.* These neighboring superpixels can belong to very different objects or far off regions of the same object in rich urban scenes. This can occur because of texture-less architecture, stochastic texture (e.g., trees, hedges) or texture repetition (e.g., windows) as shown in Figure 2(a). We refine $\mathcal{N}[S_i]$ by selecting the spatially closest superpixels. However, the irregular and highly non-convex shapes of superpixels make Euclidean distance between superpixels very ambiguous. Moreover, the size of the spatial neighborhood is also ambiguous because of the varying sizes of superpixels.

We resolve the above ambiguity using a graph traversal algorithm. We create a 2D superpixel graph by adding edges between any two superpixels which share a common boundary. We compute the path between *target superpixel* S_i^T and each *source superpixel* $S_j \in \mathcal{N}[S_i^T]$ which involves the *least change in color*. We measure the change in color between two superpixels by the χ^2 distance between their LAB histograms described above. This path is computed by minimizing the path cost C over all possible paths from S_i^T to S_j.



(a) (b)

Fig. 2. (a) Target superpixel (red) and the set of similar neighbors (yellow) in a color-content sense, (b) the shortest walk algorithm selects 3 best matches (cyan).

$$C(S_i^T \xrightarrow{\gamma} S_j) = \sum_{t=1}^{|\gamma|-1} d(\mathcal{H}_{Lab}[\gamma(t)],\ \mathcal{H}_{Lab}[\gamma(t+1)]) \quad (1)$$

$$\tilde{C}(S_i^T \rightarrow S_j) = \min_{\gamma \in \Gamma[S_i^T \rightarrow S_j]} C(S_i^T \xrightarrow{\gamma} S_j) \quad (2)$$

where $\Gamma[S_i^T \rightarrow S_j]$ is the set of all paths from target superpixel S_i^T to S_j, γ is one such path of length $|\gamma|$ such that $\gamma(0) = S_i^T$ and $\gamma(|\gamma|) = S_j$, $C(S_i \xrightarrow{\gamma} S_j)$ is the cost of path γ, and $d(\cdot,\cdot)$ is the χ^2 distance between histograms. We implement the above using the Dijkstra shortest path algorithm where the edge weight between two superpixels is the χ^2 LAB histogram distance.

We compute $\tilde{C}(S_i^T \rightarrow S_j)$ for all $S_j \in \mathcal{N}[S_i^T]$ and choose a set of three superpixels $\tilde{\mathcal{N}}[S_i^T]$ with the smallest path costs. We then plot the histogram of depth samples contained in $\cup S_k \in \tilde{\mathcal{N}}[S_i^T]$. A single strong peak in the depth histogram or two contiguous peaks (see Figure 3(a),(c)) indicate that all $S_k \in \tilde{\mathcal{N}}[S_i^T]$ are at similar depths and can be reached from S_i^T without crossing color discontinuities, which means that the superpixels are likely to belong to the same object. We obtained similar results for 3-6 superpixels with smallest paths costs; numbers higher than 6 often gave multiple peaks in the depth histogram e.g. Figure 3(d). If the final depth histogram has more than two peaks or split peaks (see Figure 3(d)), then the superpixels selected by our shortest walk algorithm most likely belong to different scene objects. We ignore such superpixels for the moment. We use an iterative approach: superpixels filled in a previous iteration are used to add depth to remaining superpixels in the next iteration. The algorithm stops when no more superpixels can be assigned depth samples. If no pixels of a particular scene object were originally reconstructed, the superpixels of such an object will find source superpixels from other objects and the final depth histogram is most likely to remain unreliable. We discard superpixels with multiple split peaks and mark them as holes (see Figure 1(d)).

Note that we could incorporate spatial distance and LAB histogram distance in a single metric by weighing them appropriately, but this would involve tuning the weights carefully for each dataset depending on image content, object shapes, etc.

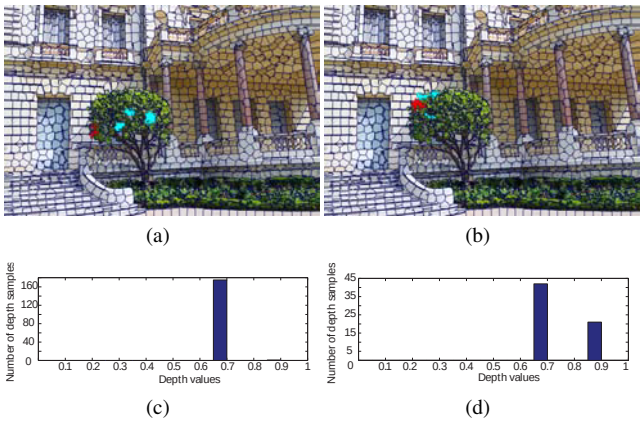


Fig. 3. Top: target superpixel in yellow and the *source superpixels* $\tilde{\mathcal{N}}[S_i^T]$ in blue. Bottom: corresponding depth histograms of $\tilde{\mathcal{N}}[S_i^T]$. Depth histogram for the first has a single peak indicating reliable depth. Split peaks in the second indicate that *source superpixels* have depth from a different scene objects. This is true for the source superpixels at the tree silhouette which contains 3D points from the wall behind the tree (see Figure 4(left)).

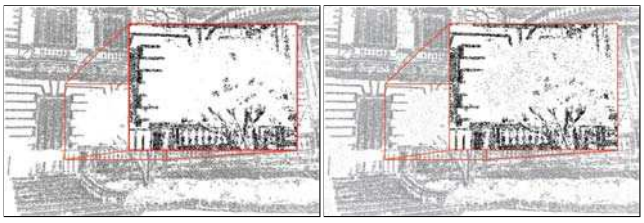


Fig. 4. Our depth synthesis adds samples with plausible depth (right) values to poorly reconstructed regions shown in the left figure (and Figure 1(c)).

4.1.3 *Interpolating depth samples.* We now interpolate depth samples from the *source superpixels* $\tilde{\mathcal{N}}[S_i^T]$. We create the combined histogram of depth samples from all source superpixels. We then create the joint probability distribution of depth samples by normalizing the histogram bin size by the total area under the histogram. This gives the approximate probability density function (PDF) of depth samples. Using the PDF as interpolation weights automatically attenuates the effect of noisy depth samples. We interpolate the inverse of depth values, as depth is inversely proportional to disparity [Goesele et al. 2010]. The final inverse depth at pixel $\mathbf{x}$ of S_i^T is given by

$$\frac{1}{D[\mathbf{x}]} = \frac{\sum_{S_k \in \tilde{\mathcal{N}}[S_i^T]} \left(\sum_{\mathbf{y} \in \mathcal{D}[S_k]} P(D[\mathbf{y}]) \|\mathbf{x}-\mathbf{y}\|^{-2} \cdot D^{-1}[\mathbf{y}] \right)}{\sum_{S_k \in \tilde{\mathcal{N}}[S_i^T]} \left(\sum_{\mathbf{y} \in \mathcal{D}[S_k]} P(D[\mathbf{y}]) \|\mathbf{x}-\mathbf{y}\|^{-2} \right)} \quad (3)$$

We add 10-15 depth samples at random pixels in S_i^T. The result for the example in Figure 1(c) is shown in Figure 4. We got similar results for 5-50 depth samples; higher numbers increased the size of the warp optimization.

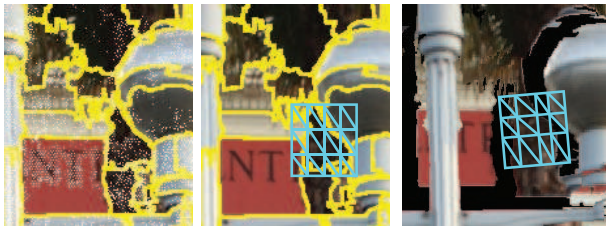


Fig. 5. Left: Superpixel segmentation showing superpixels at multiple depths as well as depth samples contained inside each superpixel (shown as white dots). Middle: The regular grid which is used as warp mesh, overlaid over each superpixel. Right: Warped superpixels and grid for a novel view. Warping each superpixels independently preserves all silhouettes. Note how background superpixels slide under foreground.

Furukawa et al. [2009] do not reconstruct sky regions. We identify such regions using the approach described in the appendix and assign them 99th percentile depth of the image before applying the above depth synthesis. This is an optional step required if there are significant sky regions.

5. LOCAL WARPING OF SUPERPIXELS WITH DEPTH SAMPLES

Depth samples from multiview stereo can be noisy, especially near silhouettes. In addition, our synthesized depth is only *plausible* rather than photo-consistent or accurate. Consequently, direct reprojection of superpixels using these depth samples, e.g., using the Video Mesh data structure [Chen et al. 2011], will result in disturbing artifacts. We demonstrate these problems in the Sec. 6.2.

To alleviate these problems, we adopt a variational warp approach to regularize the depth samples. In contrast to previous methods [Liu et al. 2009; Chaurasia et al. 2011], we do not warp the entire image, but perform an individual local warp for each superpixel, which allows much more freedom to navigate in the scene and reduces some artifacts (see Figure 10 and 11).

5.1 Shape-preserving warp

At each frame, we warp each superpixel of each image *individually* to the novel view, represented by its projection matrix C_N. Our warp satisfies two energy terms in a least-squares sense: a *reprojection energy* at each depth sample that is reprojected into the novel view, and a *shape-preserving energy* or regularization term for each warp mesh triangle that preserves the shape of the superpixel during the warp.

We create an axis-aligned bounding box for each superpixel and overlay a regular grid which serves as the warp mesh (see Figure 5, middle). Each grid triangle contains zero or more depth samples. The unknowns in the warp optimization are the warp grid vertex positions $\tilde{\mathbf{v}}$. Our variational warp energy is similar to [Liu et al. 2009; Chaurasia et al. 2011], but each superpixel is warped separately rather than warping the entire image, making it a *local* warp.

Re-projection energy. For each depth sample $D[\mathbf{x}]$, we locate the triangle T of the warp mesh that contains it. Denote the vertices of T by $(\mathbf{v}_{j_1}, \mathbf{v}_{j_2}, \mathbf{v}_{j_3})$ and let the barycentric coordinates of the

location of the depth sample at pixel $\mathbf{x}$ in triangle T be $(\alpha_j, \beta_j, \gamma_j)$:

$$\mathbf{x} = \alpha_j \mathbf{v}_{j_1} + \beta_j \mathbf{v}_{j_2} + \gamma_j \mathbf{v}_{j_3}. \quad (4)$$

The reprojection energy measures the distance between the warped position of the depth sample and the reprojected location using the novel view matrix C_N:

$$E_p[\mathbf{x}] = \left\| \alpha_j \tilde{\mathbf{v}}_{j_1} + \beta_j \tilde{\mathbf{v}}_{j_2} + \gamma_j \tilde{\mathbf{v}}_{j_3} - C_N \left(C_{I_i}^{-1}(D[\mathbf{x}]) \right) \right\|^2, \quad (5)$$

where $C_{I_i}^{-1}$ is the back-projection operator of image I_i.

Shape-preserving energy. For each triangle of the warp mesh with vertices $(\mathbf{v}_{m_1}, \mathbf{v}_{m_2}, \mathbf{v}_{m_3})$, this energy term measures its shape distortion after the warp. Ideally the triangle only undergoes a similarity transformation, resulting in a null energy value. The similarity energy is obtained by expressing one vertex of the triangle as a linear combination of the other two:

$$\begin{aligned} a &= (\mathbf{v}_{m_3} - \mathbf{v}_{m_1})^T (\mathbf{v}_{m_2} - \mathbf{v}_{m_1}) / \|\mathbf{v}_{m_2} - \mathbf{v}_{m_1}\| \quad (6) \\ b &= (\mathbf{v}_{m_3} - \mathbf{v}_{m_1})^T R_{90} (\mathbf{v}_{m_2} - \mathbf{v}_{m_1}) / \|\mathbf{v}_{m_2} - \mathbf{v}_{m_1}\| \\ E_s[T] &= \|\tilde{\mathbf{v}}_{m_3} - (\tilde{\mathbf{v}}_{m_2} + a\,(\tilde{\mathbf{v}}_{m_1} - \tilde{\mathbf{v}}_{m_2}) + b\, R_{90}(\tilde{\mathbf{v}}_{m_1} - \tilde{\mathbf{v}}_{m_2}))\|^2, \end{aligned}$$

where R_{90} is 90° rotation. Please refer to [Liu et al. 2009; Chaurasia et al. 2011] for the derivation of this energy term. The overall energy function for the superpixel warp is given by

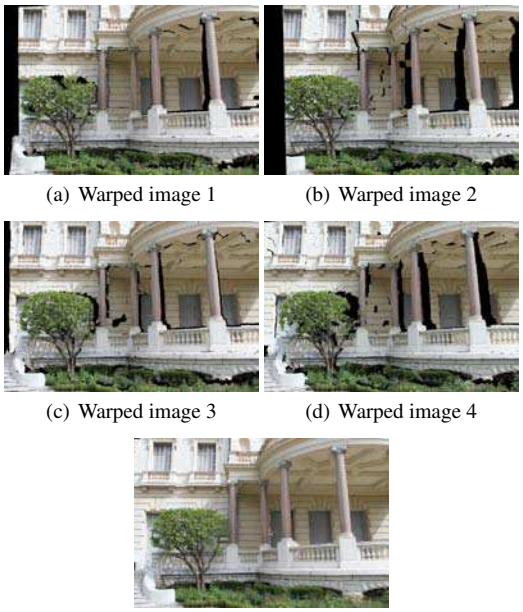
$$E_w[S_k] = \sum_{\mathbf{x} \in \mathcal{D}(S_k)} E_p[\mathbf{x}] + \sum_T E_s[T]. \quad (7)$$

We minimize $E_w[S_k]$ for each superpixel by building a sparse linear system and solving it using CHOLMOD [Chen et al. 2008] on the CPU. We solve thousands of small independent local warps in parallel, which is faster than a single global warp as in [Liu et al. 2009; Chaurasia et al. 2011]. We compare to [Chaurasia et al. 2011] in Sec. 6.2 and also discuss the effect of the shape-preserving warp as compared to methods which reproject depth samples directly (e.g., [Chen et al. 2011]).

5.2 Rendering

Rendering is achieved in three passes. In the first pass, we select and warp the four closest input cameras. Next, we blend the resulting warped superpixel images to synthesize the novel view. A final hole-filling pass completes the rendering algorithm.

Pass 1: Camera selection and warping. For each novel view, we select the four input cameras closest to the novel camera position based on camera orientation. We warp the superpixels of each of these images as described previously and render the warped superpixels of each image in a separate floating point render target with depth test enabled. We reproject the median depth of a superpixel[1] into the novel view and use it for the depth test. The warp mesh of each superpixel is rendered with an alpha matte defined by the outline of the superpixel. We use a "soft alpha matte" by rendering an additional 4 pixel wide zone outside the superpixel boundary if the neighboring superpixel's median depth is almost the same as the current superpixel. This fills in small cracks between warped superpixels, if any. We store the reprojected median depth and the superpixel ID of each warped superpixel in an additional render target while warping. These are used in the next pass to compute blending weights. This gives us four warped images where occluded background superpixels slide under foreground superpixels and disocclusions create holes in the warped images (see Figure 6).



(a) Warped image 1 (b) Warped image 2

(c) Warped image 3 (d) Warped image 4

(e) Blended image

Fig. 6. Warped superpixel images and final result after blending.

Pass 2: Blending. We render a screen-size quad into the frame buffer and blend the colors from the four warped images to get the final result. When shading a pixel in the fragment shader, we assign an appropriate weight for each of the four warped images. A number of different blending strategies have been proposed for composing novel views. View angle penalties have been used in [Buehler et al. 2001] to compute smoothly varying blending weights. Chaurasia et al. [2011] scale the weight of the highest weighted image by an additional factor to minimize blending, which has been demonstrated to be a perceptually objectionable ghosting artifact [Vangorp et al. 2011]. We first compute the angle penalties [Buehler et al. 2001] at each pixel and then discard the two lowest weight candidates to avoid excessive blending.

We use an adaptive blending scheme by creating a superpixel correspondence graph across images. We add a correspondence edge between two superpixels from different images if they share 3D reconstructed points. If the pixels to be blended have a correspondence edge, we use the weights computed above. If superpixels do not have such a correspondence edge and one superpixel contains "true" depth samples obtained from PMVS while the other contains depth samples added by our synthesis, we increase the weight of the former by a factor of 2.0. In all other cases of non-corresponding superpixels, we reduce ghosting artifacts by using the heuristic that it is better to display incorrect parallax on background regions; background parallax errors being less noticeable than those in the foreground. We therefore increase the blending weight of the pixel with the higher depth value by a factor of 2.0; tests showed that this value provides satisfactory results on our datasets. Values higher than 4 effectively disable blending.

[1] computed as median of all depth samples contained within the superpixel.

Table I. Depth synthesis running times

Scene	1	2	3	4	5	6	7	8	9	10	11	12
Images	27	30	28	12	25	30	20	10	24	25	35	36
DS	46	66	75	51	126	136	41	23	57	50	152	120

Number of images used and depth synthesis times in minutes. 1: Museum1, 2: Museum2, 3: University, 4: Yellowhouse_12, 5: ChapelHill1, 6: ChapelHill2, 7: Aquarium_20, 8: Street_10, 9: VictorHugo1, 10: VictorHugo2, 11: Commerce, 12 School.

Pass 3: Hole filling. Moving the novel view significantly away from input cameras creates large disoccluded regions which are not captured by any of the input images. Such regions appear as holes; we solve the Poisson equation [Pérez et al. 2003] with zero gradient values to create blurred color in such holes (see Figure13(c)).

6. RESULTS AND COMPARISONS

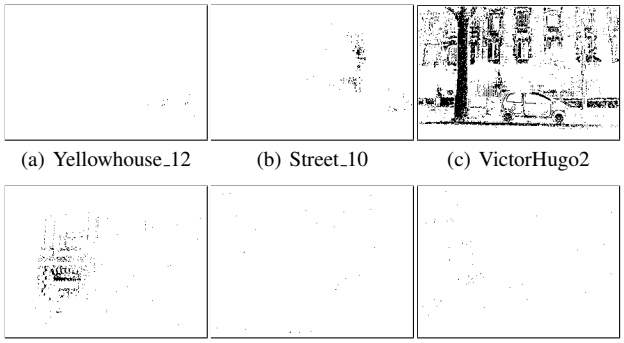
We present results and comparisons, which are best appreciated by watching the accompanying video and supplemental material.

6.1 Results

We have tested our approach on a wide variety of datasets, including scenes captured by ourselves and by others. We downloaded School[2] from Microsoft Photosynth. ChapelHill1 and ChapelHill2 are from the street-side capture in [Pollefeys et al. 2008]; we subsampled the video stream to simulate a sparse casual photo capture. Aquarium_20, Street_10 and Yellowhouse_12 are taken from [Chaurasia et al. 2011] which assumes manual silhouette marking and thus includes challenging reflective surfaces (car windows). We have additionally captured six new scenes: Museum1, Museum2, University, VictorHugo1, VictorHugo2 and Commerce. We show synthesized views for viewpoints which are quite far from input cameras in Figure 7. We list the number of images and running times for depth synthesis for all the datasets in Table I. Only 10 to 35 images are required for all our scenes. Depth synthesis running times are reported for an unoptimized MATLAB implementation which could be accelerated by an order of magnitude by running multiple images of the dataset in parallel on separate cores. Multiview stereo including Bundler [Snavely et al. 2006] and PMVS [Furukawa and Ponce 2009] took between 30-60 minutes for all our datasets depending upon the number of images. We modified the oversegmentation source code of [Achanta et al. 2012] to segment multiple images in parallel which gave running times of 1-3 minutes for all the images in any our datasets.

Rendering is real-time with an average frame rate of 53 FPS and 50 FPS at 800×600 and 1280×800 resolutions respectively on a 12-core Intel Xeon X5650 2.67Ghz CPU with NVIDIA Quadro 6000 GPU running Fedora 16. We achieve 23 FPS and 13 FPS respectively on a laptop with a dual-core Intel 2640M 2.80GHz CPU and NVIDIA GTX 525M GPU running Fedora 16.

Our algorithm works well on a variety of different scenes, which all include challenging cases of poorly reconstructed vegetation and other foreground objects (e.g. cars). As shown in Figure 8, such regions get very few depth samples from multiview stereo. Piecewise-planar techniques like [Sinha et al. 2009] tend to ignore these depth samples while finding dominant planes in the scene, while [Goesele et al. 2010] use "ambient point clouds" to produce an NPR effect. In contrast, our depth synthesis facilitates plausible rendering using just these few points. More often than not, urban or suburban scenes do contain trees, vegetation and cars; our method thus represents a significant step in making IBR algorithms practical.



(a) Yellowhouse_12 (b) Street_10 (c) VictorHugo2

(d) Aquarium_20 (e) ChapelHill1 (f) ChapelHill2

Fig. 8. Original reconstructed points for one of the images from some of our datasets. Though architecture is well reconstructed, regions with vegetation or other foreground objects are very poorly reconstructed. Our approach is capable of generating plausible renderings even for such regions.

6.2 Comparisons

There exists a vast literature on image-based rendering techniques. However, only a few recent solutions target the type of datasets we focus on, i.e., scenes captured with a simple digital camera, in which large regions are very poorly reconstructed.

Overall IBR comparison. To evaluate our overall IBR result, we compare our method to three recent approaches. We compare to Floating Textures [Eisemann et al. 2008] using the author's implementation. This approach also requires a 3D model or "proxy" of the scene, which we create using [Kazhdan et al. 2006] from the reconstructed point cloud. We use our own implementation for Ambient Point Clouds [Goesele et al. 2010] and the author's implementation for Silhouette-aware Warping [Chaurasia et al. 2011]. To validate our implementation of [Goesele et al. 2010], we provide a rendering of the Hanau dataset in the supplemental material which shows that our implementation closely resembles the original method. We also implemented the rendering method of [Chen et al. 2011], which is an alternative warp approach based on reprojection, allowing a comparison to our shape-preserving warp.

In Figure 9, we compare our view interpolation results for Yellowhouse_12 and Museum1 datasets. Floating textures [Eisemann et al. 2008] have ghosting artifacts because poor or wrong 3D geometry leads to texture misalignment which are too big to compensate by optical flow. [Goesele et al. 2010] use a NPR effect by smearing an ambient point cloud for all poorly reconstructed regions which leads to disturbing artifacts if such regions lie on important scene objects, e.g., cars, trees etc. Our depth synthesis allows plausible novel views even for such regions. Despite the manual silhouette marking, [Chaurasia et al. 2011] gives distortions in several regions which is even more pronounced if the novel camera

[2]http://photosynth.net/view.aspx?cid=aaeb8ecf-cfef-4c03-be42-bc1ae2f896c0

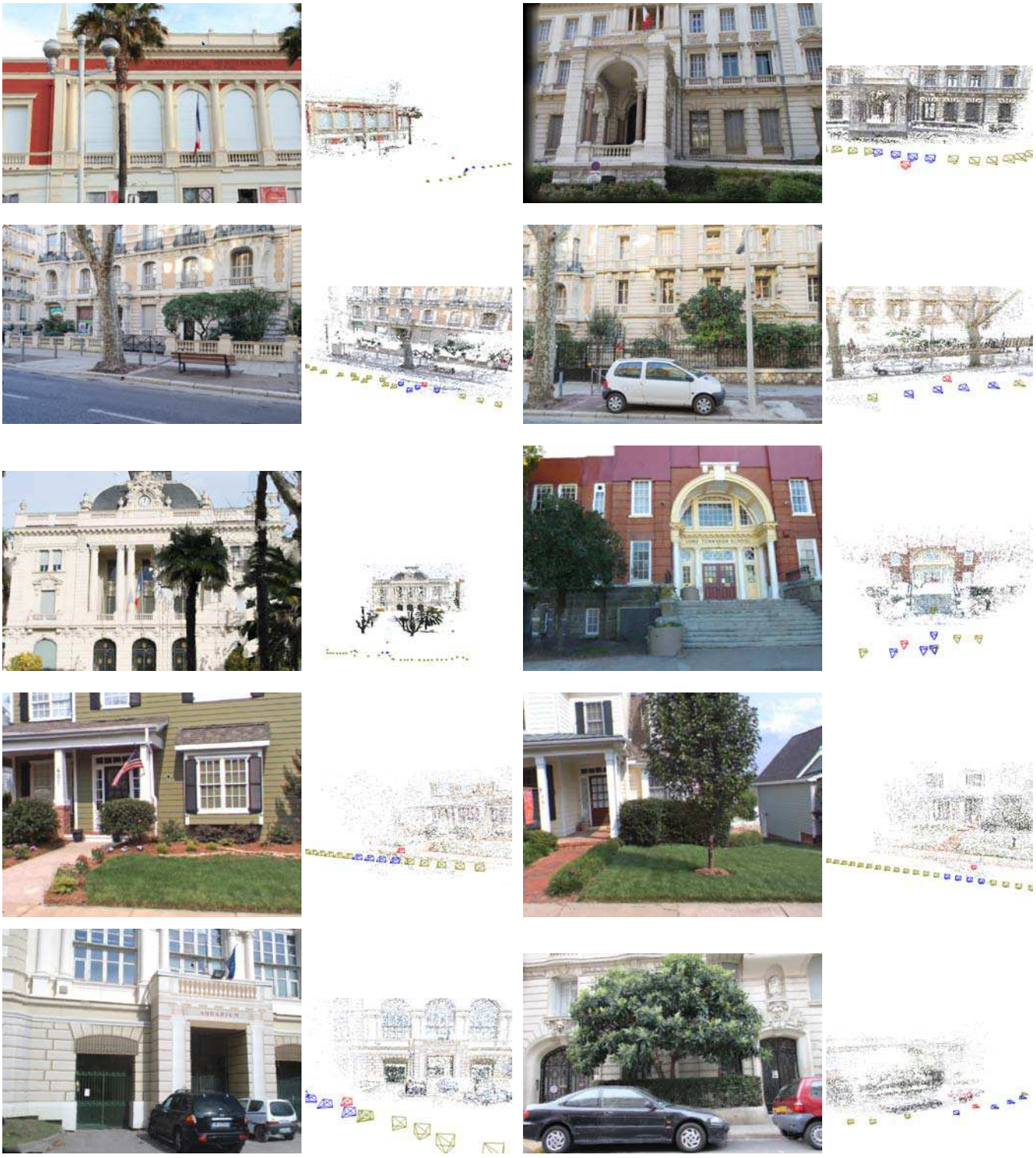


Fig. 7. A single frame and corresponding top view of the scene for all the datasets. In scanline order, University, Museum2, VictorHugo1, VictorHugo2, Commerce (our capture); School (Photosynth); ChapelHill1, ChapelHill2 ([Pollefeys et al. 2008]); Aquarium_20, Street_10 ([Chaurasia et al. 2011]) datasets. The top view shows the input cameras in yellow, novel camera in red and the 4 images selected for generating the novel view in blue. Please see video and supplemental material for all complete recorded sequences.

(a) Our result (b) [Eisemann et al. 2008] (c) [Goesele et al. 2010] (d) [Chaurasia et al. 2011]

Fig. 9. View interpolation comparison for the Yellowhouse_12 and Museum1 datasets. [Eisemann et al. 2008] depends on a 3D model and thus shows significant ghosting. In regions with very poor depth (see Figure 8), our method is able to create plausible results while [Goesele et al. 2010] creates a smeared point cloud. [Chaurasia et al. 2011] gives results similar to ours after 1.5 hours of manual intervention to mark accurate silhouettes and add/correct depth samples, however some distortions are still visible which become much more pronounced away from view-interpolation path (see Figure 10).

(a) Our result (b) [Eisemann et al. 2008] (c) [Chaurasia et al. 2011] (d) Novel camera position

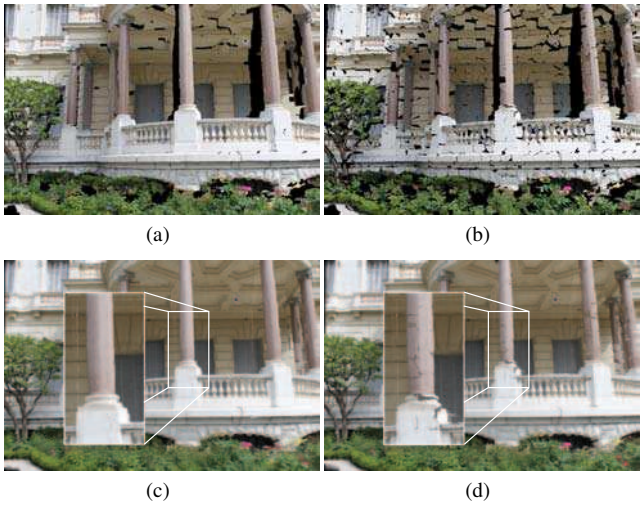
Fig. 10. Free-viewpoint navigation comparison for the Yellowhouse_12 and Museum1 datasets. Our method produces plausible results even for viewpoints quite far from the input images. In contrast, the artifacts of [Eisemann et al. 2008] are clearly visible. The distortions incurred by the global warp of [Chaurasia et al. 2011] are even more pronounced, despite 1.5 hours of manual intervention.

is moved away from the view interpolation path, as shown in Figure 10. We do not include [Goesele et al. 2010] in free-viewpoint IBR comparison because it is designed only for view interpolation.

The results for Museum1 dataset for [Chaurasia et al. 2011] in Figure 9 and 10 required 1.5 hours of manual intervention because a large number of silhouettes had to be marked and depth samples had to be added in large regions such as trees. Even then, the results show a lot of distortion because the global warp diffuses distortions due to the slightest depth gradients over the whole image, which become particularly severe when moving away from the view interpolation path (see Figure 10). Adding too many intersecting silhouettes into the Conformal Delaunay triangulation of [Chaurasia et al. 2011] leads to numerical issues. In contrast, our method scales to scenes with arbitrary number of silhouettes. Also, the global warp disintegrates when any depth sample of the input image lies behind the novel camera because such a depth sample behind cannot be projected into the novel camera (see Figure 11). Our local warp simply ignores the superpixels which contain such depth samples, while the rest of the image is warped normally. This makes our approach suitable for potential immersive applications.

Comparison with Video Mesh. The warp described in Video Mesh [Chen et al. 2011] triangulates and reprojects depth samples directly into the novel view. Inaccurate or outlier depth values can cause the depth sample to be reprojected at incorrect pixel coordinates, causing objectionable artifacts, most noticeable in the form of cracks. Our warp regularizes the effect of noisy depth values and outliers with the shape preserving constraint (see Sec. 5). As a consequence, our results have far fewer cracks (see Figure 12).

Fig. 11. The global warp of [Chaurasia et al. 2011] (left) disintegrates if any depth samples is *behind* the novel camera as shown in top view (right). This prevents the user from walking "into" the scene. Our local warp does not suffer from this limitation (middle).



(a) (b) (c) (d)

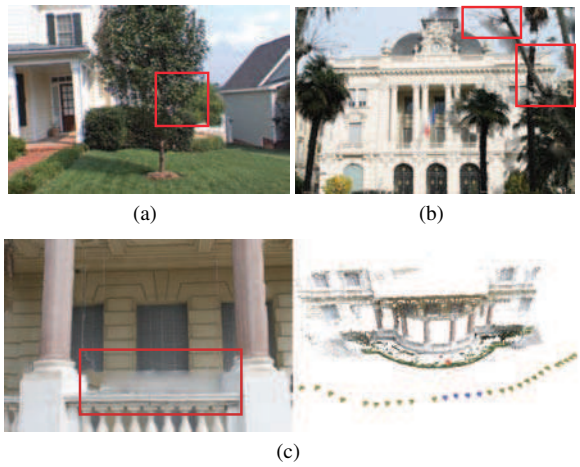
Fig. 12. (a) Superpixels warped using our approach, (b) superpixels warped using our implementation of Video Mesh [Chen et al. 2011], (c) final result generated from our warped superpixels in (a), (d) final result generated from Video Mesh style warping in (b).

6.3 Limitations

We have demonstrated our method on a large and diverse set of very challenging scenes. Evidently, the method does have limitations. The first limitation comes from depth synthesis: if the target superpixel corresponds to an object at a depth which does not exist elsewhere in the image, incorrect depth may be assigned. This is shown in Figure 13(a), where the background tree is not reconstructed at all and ends up being assigned depth from the foreground tree. The confounding factors are that the trees are spatial neighbors and have extremely similar color/texture to the extent that the boundary between the trees is barely discernible to the human eye. Depth synthesis does not handle completely unreconstructed regions dynamic content e.g., people. Our approach is limited by the capabilities of the oversegmentation: very thin structures cannot be captured (see Figure 13(b)). Finally, our hole filling approach is very basic. We resort to blurring in holes caused by disocclusions if we move far from the input views and visualize regions of the scene not captured in the input images. We discuss possible solutions to these limitations in Sec. 7.

7. DISCUSSION, FUTURE WORK AND CONCLUSIONS

We have presented a new approach to provide plausible image-based rendering for navigation in casually captured multiview datasets which have poorly reconstructed regions. Such regions are



(a) (b) (c)

Fig. 13. (a) Incorrect depth assignment on the unreconstructed background tree which is barely distinguishable from the foreground tree, (b) very thin structures cannot be properly represented by superpixel and result in rendering artifacts, and (c) hole filling in disoccluded regions not captured in input images results in blurring.

due to hard-to-reconstruct objects such as vegetation or complex foreground geometry, which occur very frequently in real-world scenes. We have used [Furukawa and Ponce 2009] for reconstruction; we believe that such scenes will prove to be challenging for any multiview stereo algorithm.

We present a depth-synthesis algorithm using a graph structure built on an oversegmentation of the input images. This step provides a *plausible* solution for regions with very sparse 3D reconstruction while other existing approaches [Sinha et al. 2009; Goesele et al. 2010] ignore such sparse depth. We then use the oversegmentation to develop a localized shape-preserving warp and rendering algorithm. This approach has a very low run-time overhead, and our three-pass GPU implementation allows interactive display rates. We demonstrate our approach on 12 different datasets, including one from Microsoft Photosynth, and others from [Pollefeys et al. 2008] and [Chaurasia et al. 2011], apart from our own datasets. We also present comparisons with the three most recent IBR algorithms which can treat datasets with similar properties [Eisemann et al. 2008; Goesele et al. 2010; Chaurasia et al. 2011]. Our method avoids many of the visual artifacts in these previous methods, and has significant advantages such as free-viewpoint navigation (compared to [Goesele et al. 2010]) and the fact that it requires no manual intervention (compared to [Chaurasia et al. 2011]).

We also discussed the limitations of our method (Sec. 6.3), which leads naturally to directions for future work. In particular, we will investigate ways to provide structure-preserving hole-filling when moving too far from the input viewpoints. Inpainting [Criminisi et al. 2003], combined with recent acceleration techniques e.g., PatchMatch [Barnes et al. 2009], could provide a basis for such a solution. However, achieving acceptable levels of quality and speed requires significant algorithmic advances, which could be based on the availability of depth and silhouette information provided by our approach. A second direction involves a way to combine piecewise planar-reconstruction [Gallup et al. 2010] with our depth synthesis algorithm; this would involve rethinking how we combine oversegmentation with synthesis. The treatment of reflections and

transparency is still challenging in our approach. Recent advances [Sinha et al. 2012] provide a promising direction.

Finally, we believe that our approach is a significant step towards plausible free-viewpoint image-based navigation from internet photo collections. This is why we have focused on images captured casually using consumer cameras instead of assuming studio capture or stereo setups.

APPENDIX

We discuss the details of depth synthesis for images which have significant sky regions, specifically the University and ChapelHill2 datasets. Our depth synthesis approach can synthesize depth values on objects which have *some* though sparse depth samples. Large regions of sky typically have no depth samples at all. We identify such sky regions in the image using a graph-cut. We assume that the images are captured upright and sky pixels are close to the top border. We create a graph with all the pixels of the image as nodes and add edges between adjacent pixels. The label costs for the graph cut are given in the following table. We keep a very high penalty

Pixel	Label 0 cost	Label 1 cost
Pixels along top border contained in superpixels with no depth samples	0	10^6
All other pixels contained in a superpixel with no depth samples	1	0
All other pixels	10^6	0

of 10^6 for having neighboring pixels with different labels, except at superpixel boundaries where we relax it to 100. After computing the graph cut using [Kolmogorov and Zabih 2004], we mark the pixels labeled 0 as sky and assign them 99th percentile depth of the image. Note that [Hoiem et al. 2007] may be used to identify sky regions; we resort to this approach because it is sufficient and much faster.

ACKNOWLEDGMENTS

We thank Martin Eisemann for providing us with the source code of [Eisemann et al. 2008], Michael Goesele for helping us implement their approach [Goesele et al. 2010] and Sylvain Paris for reviewing an earlier draft on the paper.

REFERENCES

ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P., AND SÜSSTRUNK, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. PAMI 34,* 11, 2274 –2282.

ANDREETTO, M., ZELNIK-MANOR, L., AND PERONA, P. 2008. Unsupervised learning of categorical segments in image collections. In *CVPR Workshops*.

BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph. 28,* 3, 24:1–24:11.

BLEYER, M., ROTHER, C., KOHLI, P., SCHARSTEIN, D., AND SINHA, S. 2011. Object stereo joint stereo matching and object segmentation. In *Proc. CVPR*. 3081–3088.

BUEHLER, C., BOSSE, M., MCMILLAN, L., GORTLER, S., AND COHEN, M. 2001. Unstructured lumigraph rendering. In *SIGGRAPH*. 425–432.

CHAURASIA, G., SORKINE, O., AND DRETTAKIS, G. 2011. Silhouette-aware warping for image-based rendering. *Comput. Graph. Forum (Proc. EGSR) 30,* 4, 1223–1232.

CHEN, J., PARIS, S., WANG, J., MATUSIK, W., COHEN, M., AND DURAND, F. 2011. The video mesh: A data structure for image-based three-dimensional video editing. In *Proc. ICCP*.

CHEN, Y., DAVIS, T. A., HAGER, W. W., AND RAJAMANICKAM, S. 2008. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw. 35,* 3, 22:1–22:14.

CIGLA, C., ZABULIS, X., AND ALATAN, A. 2007. Region-based dense depth extraction from multi-view video. In *Proc. ICIP*.

CRIMINISI, A., PÉREZ, P., AND TOYAMA, K. 2003. Object removal by exemplar-based inpainting. In *Proc. CVPR*. 721–728.

DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proc. SIGGRAPH*. 11–20.

DOLSON, J., BAEK, J., PLAGEMANN, C., AND THRUN, S. 2010. Upsampling range data in dynamic environments. In *Proc. CVPR*. 1141–1148.

EISEMANN, M., DECKER, B. D., MAGNOR, M., BEKAERT, P., DE AGUIAR, E., AHMED, N., THEOBALT, C., AND SELLENT, A. 2008. Floating textures. *Comput. Graph. Forum 27,* 2, 409–418.

FELZENSZWALB, P. F. AND HUTTENLOCHER, D. P. 2004. Efficient graph-based image segmentation. *Int. J. Comput. Vision 59*, 167–181.

FUHRMANN, S. AND GOESELE, M. 2011. Fusion of depth maps with multiple scales. In *Proc. SIGGRAPH Asia*. 148:1–148:8.

FURUKAWA, Y., CURLESS, B., SEITZ, S. M., AND SZELISKI, R. 2009. Manhattan-world stereo. In *Proc. CVPR*. 1422–1429.

FURUKAWA, Y. AND PONCE, J. 2009. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. PAMI 32,* 8, 1362–1376.

GALLUP, D., FRAHM, J.-M., AND POLLEFEYS, M. 2010. Piecewise planar and non-planar stereo for urban scene reconstruction. In *CVPR*.

GOESELE, M., ACKERMANN, J., FUHRMANN, S., HAUBOLD, C., AND KLOWSKY, R. 2010. Ambient point clouds for view interpolation. *ACM Trans. Graph. 29*, 95:1–95:6.

GOESELE, M., SNAVELY, N., CURLESS, B., HOPPE, H., AND SEITZ, S. M. 2007. Multi-view stereo for community photo collections. In *ICCV*.

GRUNDMANN, M., KWATRA, V., HAN, M., AND ESSA, I. 2010. Efficient hierarchical graph based video segmentation. In *Proc. CVPR*.

GUPTA, A., BHAT, P., DONTCHEVA, M., CURLESS, B., DEUSSEN, O., AND COHEN, M. 2009. Enhancing and experiencing spacetime resolution with videos and stills. In *Proc. ICCP*.

HAWE, S., KLEINSTEUBER, M., AND DIEPOLD, K. 2011. Dense disparity maps from sparse disparity measurements. In *ICCV*.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2007. Recovering surface layout from an image. *Int. J. Comput. Vision 75,* 1, 151–172.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Proc. SGP*. 61–70.

KOLMOGOROV, V. AND ZABIH, R. 2004. What energy functions can be minimized via graph cuts? *IEEE Trans. PAMI 26,* 2, 147–159.

KOWDLE, A., SINHA, S. N., AND SZELISKI, R. 2012. Multiple view object cosegmentation using appearance and stereo cues. In *ECCV*.

LEVOY, M. AND HANRAHAN, P. 1996. Light field rendering. In *Proc. SIGGRAPH*. 31–42.

LIPSKI, C., LINZ, C., BERGER, K., SELLENT, A., AND MAGNOR, M. 2010. Virtual video camera: Image-based viewpoint navigation through space and time. *Comput. Graph. Forum 29,* 8, 2555–2568.

LIU, F., GLEICHER, M., JIN, H., AND AGARWALA, A. 2009. Content-preserving warps for 3D video stabilization. In *SIGGRAPH*. 44:1–44:9.

MAHAJAN, D., HUANG, F.-C., MATUSIK, W., RAMAMOORTHI, R., AND BELHUMEUR, P. 2009. Moving gradients: A path-based method for plausible image interpolation. *ACM Trans. Graph. 28,* 3.

MCMILLAN, L. AND BISHOP, G. 1995. Plenoptic modeling: an image-based rendering system. In *Proc. SIGGRAPH*. 39–46.

MIČUŠÍK, B. AND KOŠECKÁ, J. 2010. Multi-view superpixel stereo in urban environments. *Int. J. Comput. Vision 89,* 1, 106–119.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. In *Proc. SIGGRAPH*. 313–318.

POLLEFEYS, M., NISTÉR, D., FRAHM, J. M., AKBARZADEH, A., MORDOHAI, P., CLIPP, B., ENGELS, C., GALLUP, D., KIM, S.-J., MERRELL, P., SALMI, C., SINHA, S., TALTON, B., WANG, L., YANG, Q., STEWÉNIUS, H., YANG, R., WELCH, G., AND TOWLES, H. 2008. Detailed real-time urban 3D reconstruction from video. *Int. J. Comput. Vision 78,* 2-3, 143–167.

SEITZ, S. M., CURLESS, B., DIEBEL, J., SCHARSTEIN, D., AND SZELISKI, R. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*. Vol. 1.

SINHA, S. N., KOPF, J., GOESELE, M., SCHARSTEIN, D., AND SZELISKI, R. 2012. Image-based rendering for scenes with reflections. *ACM Trans. Graph. 31,* 4, 100:1–100:10.

SINHA, S. N., MORDOHAI, P., AND POLLEFEYS, M. 2007. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *ICCV*.

SINHA, S. N., STEEDLY, D., AND SZELISKI, R. 2009. Piecewise planar stereo for image-based rendering. In *Proc. ICCV*. 1881–1888.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph. 25,* 3, 835–846.

STICH, T., LINZ, C., WALLRAVEN, C., CUNNINGHAM, D., AND MAGNOR, M. 2011. Perception-motivated interpolation of image sequences. *ACM Trans. Appl. Percept. 8,* 2, 11:1–11:25.

VANGORP, P., CHAURASIA, G., LAFFONT, P.-Y., FLEMING, R. W., AND DRETTAKIS, G. 2011. Perception of visual artifacts in image-based rendering of façades. *Comput. Graph. Forum 30,* 4, 1241–1250.

YANG, Q., YANG, R., DAVIS, J., AND NISTÉR, D. 2007. Spatial-depth super resolution for range images. In *Proc. CVPR*.

ZITNICK, C. L., JOJIC, N., AND KANG, S. B. 2005. Consistent segmentation for optical flow estimation. In *Proc. ICCV*. 1308–1315.

ZITNICK, C. L. AND KANG, S. B. 2007. Stereo for image-based rendering using image over-segmentation. *Int. J. Comput. Vision 75,* 1, 49–65.

ZITNICK, C. L., KANG, S. B., UYTTENDAELE, M., WINDER, S., AND SZELISKI, R. 2004. High-quality video view interpolation using a layered representation. *ACM Trans. Graph. 23,* 3, 600–608.