Additional Material (electronic only)

This additional material contains a presentation of additional capabilities of the system, a discussion of performance and temporal coherence as well as other limitations. We conclude with a set of additional results.

1. Additional Capabilities

Our pipeline supports compositing of CG objects with the synthesized scenes, image manipulation and interactive editing.

1.1. Compositing and Image Manipulation

Our approach supports several ways of enriching and manipulating the modeled scene. In particular, 3D CG elements can be integrated with the synthesized scene using the extended depth and the same camera as the proxy image at any given frame. They can then cast shadows into the scene by simple modulative shadow mapping using the extended depth map (e.g., car example Fig. 9). The layered representation of the scene also allows for image manipulation (e.g., changing the sky color) or other compositing effects such as fog.

1.2. Interactive Editing

We adapt the rendering pipeline to support editing of the proxy with interactive updates of the synthesized image. A naive implementation would resynthesize a new image after each edit, resulting in incoherence with the image prior to the edit. We introduce a progressive guide update that yields coherent image updates. Each modification of the proxy is reprojected in the first camera frame. Modified regions are flagged for guide resynthesis. We initialize flagged areas with the new labels and perform correction passes. The proxy depth is extended to match the new detailed silhouettes (§4.3 of the main paper). Pixel colors are then resynthesized in areas where the synthesized guide has been modified. We show interactive scene editing in the accompanying video.

2. Comparisons

We next present three types of comparisons. The first is an informal comparison to a trained modeller, the second is a comparison to the original image analogies approach, and the third a comparison with the CG2Real pipeline.

2.1. Informal Comparison with a trained modeller

We showed a professionally-trained modeller the mountain image (Fig. 9, second row, top-left), and an animation result

Figure 1: *Scene modeled in Maya in 2h30 (left) and by our method in a few minutes (right). Maya rendering suffers from distorted textures and smooth geometry.*

from our pipeline (see video). We asked her to reproduce the animation with as high quality as possible in 2.5 hours. We created a one-frame-per-minute time-lapse of her work (see video). The first hour was largely spent on creating the geometry, while the second hour was spent mainly on creating textures in Photoshop and editing UV's. The end result lacks the rich detail of our silhouettes and contains texture resolution and distortion artifacts. In Fig. 1 we show a frame of the resulting animation (left), and our result (right). An untrained user of a modelling program would likely be incapable of producing such a result even given much more time. In comparison, our image-based shading method only requires a very approximate proxy geometry for a desired scene, once the input photo has been segmented.

Figure 2: *Image Analogies results for three test scenes.*

2.2. Comparison to Image Analogies

Our method is inspired by the texture-by-numbers approach of the seminal Image Analogies work [HJO*01]. However, the third stage of our synthesis differs from this previous work in a number of significant respects. This includes the use of Chamfer distance to respect the discrete nature of texture labels, the introduction of a pyramid based on a voting scheme, and patch-based initialization to achieve fast high-quality synthesis results.

To evaluate these differences, we compare to the original image analogies method using the proxy guide as input and the original program made available by the authors of the original paper. Fig. 2 shows the results for a subset of our examples. While Image Analogies sometimes obtain reasonable results, some cases exhibit strong artifacts. We system-

Figure 3: *First 4 frames of a camera motion sequence. Top row: Without our temporal coherence scheme, frames are fully resynthesized, exhibiting strong incoherence accross frames. Notice the changes in the appearance of the mountain and the grass on the left of each frame. Bottom row: Our temporal coherence scheme ensures a certain temporal continuity. The effect can be seen in conjunction with the accompanying video.*

atically experimented with the κ parameter in Image Analogies in order to ensure that this was not the source of the visible artifacts. Our own method is subject to limitations also shared by Image Analogies, discussed in Sec. 3.3.

2.3. Comparison to CG2Real

Our work shares with *CG2Real* [PFA*09] the idea of using input photographs as a way to enhance modeled CG scenes. Although CG2Real works with image processing operations, thus not requiring 3D geometry, our proposed method overcomes several limitations of CG2Real.

CG2Real allows the use of multiple photographs as input. However, only textures, tone and color are transferred: silhouettes remain mostly unchanged thus lacking the rich details encountered in natural scenes if they are not exhaustively modeled by the artist. In constrast, our proposed method synthesizes silhouettes from a crude approximation – the proxy – sketched by a casual user.

Also, the metric used in CG2Real only matches colors within a region of a segmented image, without a notion of a real discrete identifier as in our method: this may lead to artifacts when regions of similar color represent semantically distinct elements.

In terms of performance, [PFA*09] requires *less than a minute* on a 600x400 image, while our method allows for near-interactive previsualization in the order of one second per frame on more than 4 times bigger images (1024x1024). The computation time for our approach is thus more than two orders of magnitude faster than that of CG2Real.

Finally and more importantly, CG2Real does not allow for temporally coherent motion, thus precluding use for walkthroughs and animation.

3. Discussion of Temporal Coherence, Performance and Other Limitations

In this section we discuss some limitations of temporal coherence and further discuss issues of performance.

3.1. Temporal coherence

As we can see in Fig. 3, our temporal coherence approach (§. 6) significantly improves the quality of the result for a moving camera compared to resynthesis at every frame. This is particularly clear in the accompanying video.

Similarly, our treatment of distortion also improves the final result, since after some movement of the camera the distortion artifacts of the texture become very visible. This can be seen both in Fig. 4 and in the video.

3.2. Performance

Our implementation is far from being optimized; for example our Poisson synthesis step takes about 300ms on the GPU whereas other implementations achieve compute times of 50ms per frame [MP08]. We could use nearest neighbor solutions on the GPU [GDB08] to further accelerate search computations. We also pay a heavy price for CPU/GPU transfers. A carefully crafted GPU-only implementation should greatly accelerate computation. The

cost of dilations, currently implemented as shaders with a large number of texture lookups is very high. Again, well-designed GPU implementations, taking into account memory access patterns etc. should result in significant acceleration.

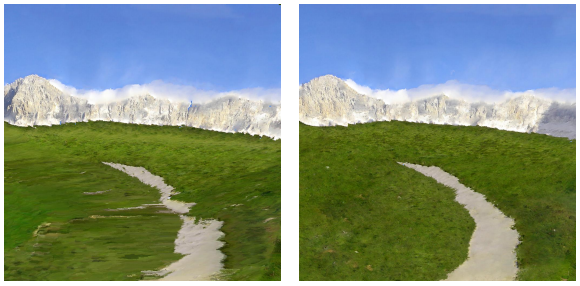

Figure 4: *After several frames with consecutive reprojections, strong distortion artefacts appear (left). Our distortion handling resynthesize pixels with high distortion thus improving quality (right).*

3.3. Other Limitations

The detailed boundaries synthesized in the first frame are only partially reused in subsequent frames: no new details are added to disoccluded regions. For large camera rotations around the proxy geometry, boundary detail is lost. This is shown in Fig. 5 where details of an island are lost when we turn the camera, thus revealing the spherical shape of the proxy geometry used for the island. In addition, the perspective should roughly match between the input photograph and the 3D scene during the camera motion. We believe that a more involved anisometric texture synthesis should overcome this issue.

3.4. Additional Results

We show a set of additional results in what follows; in particular we discuss results of Non-photorealistic rendering and dynamic lighting as well as an additional 11 images on which we have tested our approach (see Fig. 7 and 8).

Non-Photorealistic Rendering (NPR): We are not restricted to photographs and can produce temporally coherent NPR renderings based on a single input painting. We demonstrate this in the accompanying video and in Fig. 6.

Dynamic Lighting: A single input photograph only samples a single lighting condition. However, it would be convenient to interactively explore alternate lighting conditions. This is supported by using a time lapse series of input photographs. Patches used for the initial-synthesis lighting condition are reused for other time-of-day conditions. The Poisson synthesis is performed on the whole image to remove artifacts such as moving shadows. Note that the initial lighting condition should be chosen with some care: using a nighttime condition will not create an informative patch assignment. We show an example of such a time-lapse synthesis in the accompanying video.

Figure 5: *Failure: synthesized boundaries do not get updated when moving the camera. This can lead to undesirable results for camera rotation if synthesized silhouettes are very different from the smooth silhouettes as in this example. Here the spherical shape of the island becomes evident on the second image as the camera turns around the island to the left.*

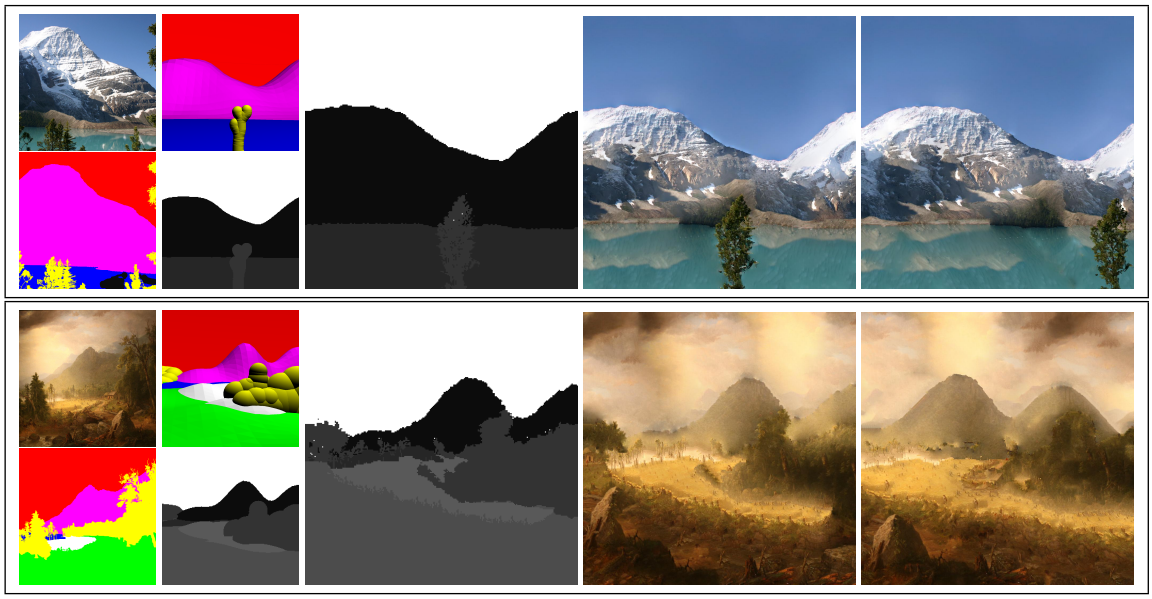

Figure 6: *More results, including NPR. Left, clockwise: Input photo, 3D proxy, corresponding IDs, photo segmentation. Then, from left to right: Synthesized IDs; Corresponding result from our pipeline; Another view with temporal coherence.*

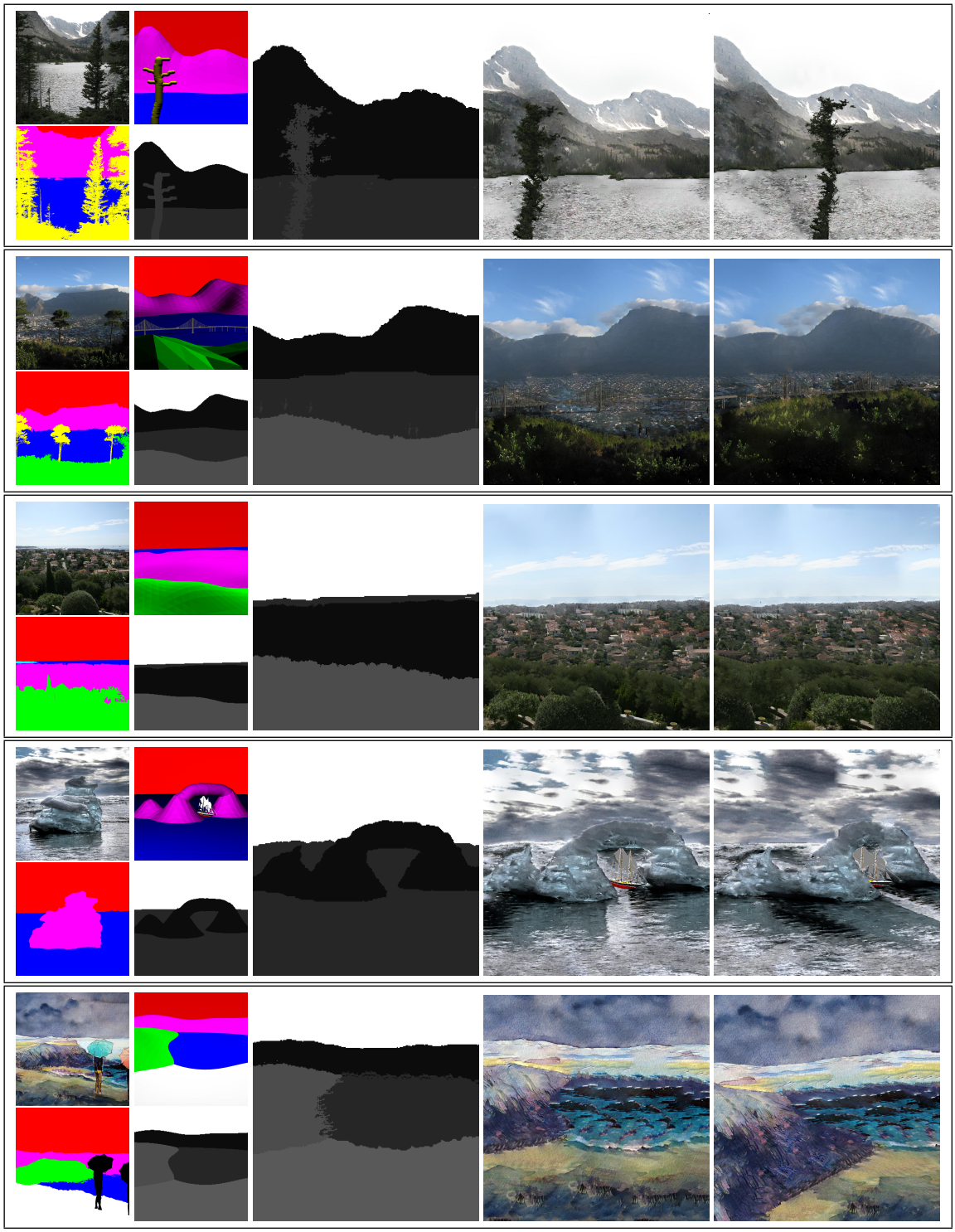

Figure 7: *More results, including integration of 3D objects, and NPR results.*

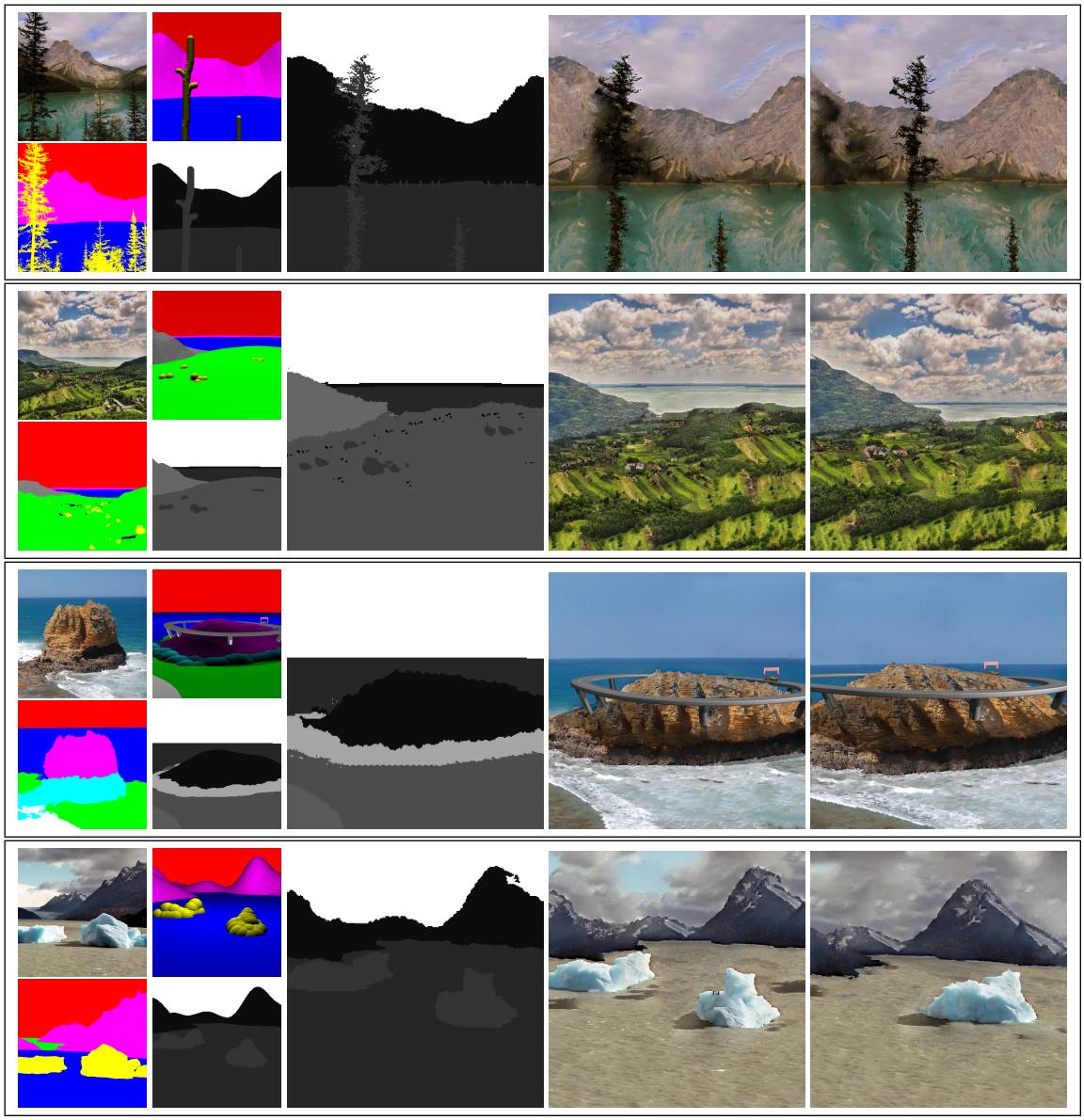

Figure 8: *More results, including integration of 3D objects*