Eurographics Symposium on Geometry Processing 2015
Mirela Ben-Chen and Ligang Liu
(Guest Editors)

Volume 34 (2015), Number 5

An Image Degradation Model for Depth-augmented Image Editing

James W. Hennessey Niloy J. Mitra

University College London

Figure 1: *A single RGB-D image (left) is used to create a novel view for a parallax photograph revealing occluded regions (right) with a corresponding degradation model (inset: red and blue respectively indicate higher and lower degradation).*

Abstract

Images remain the most popular medium to capture our surroundings. Although significant advances have been made in developing image editing tools, the key challenge is to intelligently account for missing depth information. The growing popularity of depth images offers a new avenue to revisit image editing tasks. In this work, we investigate how even coarse depth information can be exploited to address some of the fundamental challenges in image editing namely producing correct perspective, handling occlusion, and obtaining segmentation. To this end, we propose a novel image degradation model *that predicts how well an image edit can be performed in presence of coarse depth information. Technically, we create proxy geometry to summarize available depth information, and use it to predict occlusions and ordering between image patches, complete occluded regions, and anticipate image-level changes under camera movement. We evaluate the proposed image degradation model in the context of parallax photography from single depth images.*

1. Introduction

Images remain the most dominant and ubiquitous of visual mediums. A vast selection of tools exists supporting various image editing and manipulation tasks. Typically, users are interested in manipulating scene content or camera pose in order to mimic being in the original 3D scene. However, the lack of actual geometry and depth information makes such edits theoretically impossible to perform correctly.

Beyond full scale 3D acquisition, one can alternatively capture a lightfield of the scene to accurately support many advanced manipulations (e.g., change in camera pose, simulate depth of field effects, etc.). This, however, comes at the cost of specialized and costly imaging setup. In this paper, we show that even very coarse and incomplete depth information can vastly simplify many image processing tasks. This is particularly relevant given the growing ubiquity of depth sensors that capture high resolution RGB informa-

tion with loosely synchronized noisy depth information. We demonstrate such depth information can be used to plausibly handle occlusion, perspective, and completion effects — all from single view inputs. Figure 1 shows an example.

More interestingly, we propose an *image degradation model* that predicts the success likelihood of a proposed manipulation. The motivation behind the degradation model comes from two observations: (i) image completion algorithms are limited and can leave undesirable artefacts and (ii) by introducing depth information to an image it enables the control of occlusions by changing camera viewpoint. The objective of the image degradation model is to identify poorly completed regions and prevent them from being revealed. Technically, we use the rough depth information to help create a planar proxy based abstraction of the input image. We propose an iterative algorithm to segment the input, while estimating the corresponding planar proxies to act as billboards for the respective segments. The information is then used to create a layered set of planar proxies with infilled and clipped textures per layer. Finally, we estimate a degradation score for new camera poses to predict the plausibility of the synthesized image composition.

We evaluate our framework in the context of creating parallax videos from single images. This application demonstrates a number of the challenges including segmentation, image completion, perspective, occlusion and depth ordering in one use-case. We evaluate the proposed method on a variety of scenarios with both planar and non-planar objects, with and without texture.

2. Related Work

Traditional image editing. Given the popularity and ubiquity of images, significant research has been devoted over the last decades in developing image editing algorithms. Many of them are commonly available as standard options in image editing packages like Gimp, Photoshop, etc. The central challenge is to plausibly account for the lack of depth in input images. This makes it difficult to correctly handle perspective and/or occlusion effects. Even the most advanced methods like PatchMatch [BSFG09] fail when scenes are cluttered or a texture changes with the perspective of the object. In the context of segmentation, the GrabCut segmentation algorithm [RKB04] cannot satisfactorily segment objects from different depth with the same appearance. We improve the results of these algorithms by the use of geometric planar primitives.

Depth-aware image editing. Many depth-aware solutions have been proposed to tackle specific use cases and problems. For example RepFinder [CZM*10] use repeated objects in a scene to assist with image completion and depth ordering, while, Caroll et al. [CAA10] use vanishing points for artistically changing image perspective. We approach the problem with noisy depth and want to deal with these challenges for more general scenes.

A recent approach for editing man-made objects in 3D is to allow the user to create 3D proxies for objects in the scene. One example approximates objects with cuboids [ZCC*12] and another generalized cylinders [CZS*13]. The results for both are impressive but require substantial and specialized user-interaction [WCM15]. Our approach also uses geometric primitives for editing objects in 3D, however, we demonstrate that it is not necessary to accurately parameterize the objects for a range of interactions.

Our application of parallax photography shares motivation with viewpoint changes from a single image demonstrated by [OCDD01] . Their application again has significant interaction to assign a depth to each segment. Tour into the picture [HAA97] allow users to make animations from a single image by changing viewpoints, but do not complete occluded regions or achieve a parallax effect.

Rendering 3D Models in Images. Recent applications combine images with 3D models with impressive results in either editing the scene [KSES14] or realistically compositing objects in the scene [KSH*14]. These require 3D models, that match objects in the scene, to be available for edits to be made. We aim to use the depth information as go between the image and the 3D model. RGB-D images have become increasingly popular and many methods have been proposed to address the common challenges of segmentation and depth map completion [EPD12, LRL14, SMZ*14].

Our application of parallax photography can be compared with [ZCA*09] who create Parallax Photographs from LightField images. Their results are impressive but as a Lightfield is sampling the ray space the input is a lot richer (and heavy-weight). Instead RGB-D images are much easier to acquire. However, the simplicity introduces a number of new problems such as segmentation, occlusion and image completion.

3. Overview

Our application takes as input a single registered RGB-D image captured using a camera and calibrated consumer depth sensor. The RGB-D inputs have high density RGB measurements, but poor and incomplete depth information (see Figure 3). The goal is to utilize the available information to create an *image degradation* model to predict how successful typical image manipulations will be. In other words, the degradation model characterizes edits as simple, or difficult and likely to show artifacts.

In order to build such a model, we analyze an input RGB-D image to create an intermediate representation. Specifically, we segment the input (either automatically or semi-automatically), billboard-approximate them using planar proxies, obtain the relative ordering of the respective planes, and infill the occluded regions for each segment. We then build an image degradation model that captures the plausibility of the infilled pixels. The effectiveness of the

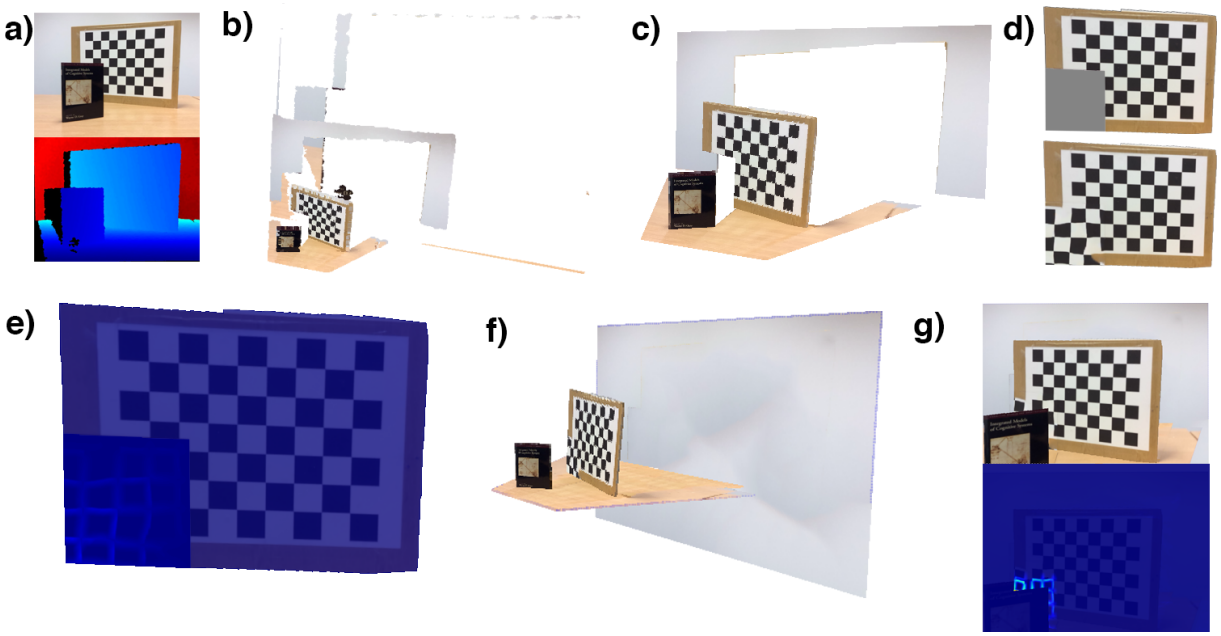

Figure 2: *Method overview: (a) Input RGB + depth image; (b) incomplete point cloud with noisy data (note the misalignment of RGB and depth); (c) segmentation, primitive fitting, and depth completion; (d) occlusions identified and infilled using the primitives (shown for one segment); (e) degradation model built for infilled pixels; (f) decomposed and completed layered scene; and, (e) new view synthesized from user defined camera pose is flagged by the degradation model as undesirable.*

proposed image degradation model is evaluated by using it to find suitable camera paths to create aesthetically pleasing parallax photographs from single images.

Figure 3: *Example RGB-D input: Point cloud rendered from original camera pose (left) showing large regions of missing depth information labeled in red. When rendered from a different camera pose (right) the point cloud reveals many points have mislabeled depth values. The coarseness of the data emphasises the need for decomposing the scene into geometric proxies.*

Our pipeline has three stages: (i) *Scene decomposition and completion* wherein we propose an iterative approach for image segmentation, depth map completion, and planar proxy fitting. These primitives are then used to determine occlusions and improve image completion. (ii) *An image degradation model* is then created consisting of a degradation score for each of the occluded pixels in each segment, completed in the previous step, representing the plausibility of the completed pixel. The degradation model consists of a spatial term and texture term. The intuitive idea behind these terms is that pixels close to known pixel values and in a low texture region should receive a low degradation score, versus those far from known pixels with a large amount of texture variations. Finally, we use the decomposed and completed scene with the computed degradation model in the (iii) *camera path generation* for a parallax photograph. Starting from user specified camera key frames we utilize the degradation model to find a good camera path between them.

Contributions. Our key contribution is an approach to use coarse depth to simplify image manipulation tasks. Central to this is a novel image degradation model that captures the quality of synthetic regions of images. Additionally we propose a method for creating proxy geometry to summarize coarse depth information and exploit these proxies when dealing with common challenges such as segmentation, occlusions, and perspective changes.

4. Method

All of our RGB-D images are captured using an Apple iPad and Occipital StructureSensor. The StructureSensor has a range of 0.4m to 3.5m+ with precision 0.5mm at 40cm, 30mm at 3m. In practice the upper bound is further but precision degrades, which, our pipeline mitigates. We use Occipital's calibration app to register the color and depth channels.

The RGB-D images are input into our system (see Fig-

ure 2 for pipeline). We convert the data to a point cloud and estimate pointwise normals using local PCA fitting. Note that the data is largely incomplete (marked in black in the depth channel) and also there is misalignment between color and depth channels as seen in the point cloud. The scene is then segmented using color and depth information (automatically or with user guidance), abstracted with planar primitives, and completed using guidance from the obtained primitives.

4.1. Scene Decomposition and Completion

The main goal of this step is to simultaneously performs segmentation, planar primitive fitting and pixel assignment. We couple these three steps in an iterative approach that reassigns pixels to primitive to improve the segmentation and obtain improved primitive fits. Figure 4 shows an example.

Scene decomposition. First, we segment the RGB image into SLIC SuperPixels [ASS*12] (Figure 4c) and fit planar primitives to the different segments. We encode the fitted primitives in the normal-intercept form as $\mathbf{n}\cdot\mathbf{p}+d=0$. We cluster the superpixel plane primitives using k-means in the $\mathbb{R}^4$ space. Empirically, we found $k=20$ provided good results (Figure 4d). We again fit planar proxies to the superpixel clusters to obtain a rough initial segmentation. Essentially, the clustering step links superpixels sharing similar fitted planes. This allows non-locally linking superpixels, for example walls are identified to be coming from the same plane even under occlusion.

An alpha-expansion graph cut [BK01] is used to improve upon the initial superpixel segmentation. The graph cut allows $(k+1)$ possible labels that a superpixel could be assigned, representing the current segments and their primitives from the superpixel clustering, and an additional possible assignment of a plane at large distance away. We use a unary cost for each label that encourage the average distance between the label's plane primitive and all points in a superpixel to be small and the average angle between point normals and plane normal to be similarly small.

$$E_u(i) := \frac{1}{N}\sum_{i=1}^{N}(|\mathbf{p}_i\cdot\mathbf{n}_{prim}+d_{prim}|)+\lambda\exp(-|\mathbf{n}_{sp}\cdot\mathbf{n}_{prim}|).$$

In our tests we set $\lambda=1000$. Note that for pixels with no assigned depth value (i.e., missing depth) we skip the unary term. If a whole superpixel has no depth data then it is given a uniform cost for all primitives.

The pairwise cost for the graph encourages the neighbouring primitives to have similar color and depth as:

$$E_p(i,j) := \alpha\exp(-|\mathbf{c}_i-\mathbf{c}_j|)+\beta\exp(-|d_i-d_j|) \qquad (1)$$

where, $\mathbf{c}_x$ and d_x respectively denote the mean color and depth values assigned to the current segmentation primitives and normalized between 0 and 1. We used $\alpha=1000$ and $\beta=200$ in our tests. Again, we exclude the depth term here if one of the pixels in a superpixels have no associated depth value. Finally, we refit planar segments to the updated segmentation results. The SuperPixel level graph cut can be seen in Figure 4e.

We then iteratively refine the segmentation and depth map but now working at the pixel level. Each of the three iterations consists of performing a pixel level alpha expansion, updating the primitives, and updating the depth map. Specifically, the alpha expansion uses the same terms as previously. However, as we are working at the pixel level the point to primitive distance is no longer averaged, nor is the color or depth term in the pairwise cost. We update the depth map by setting each point's position as the ray-plane intersection for the assigned primitive. In the first iteration, we only reassign the pixels with already known depth to correct flying pixels. Subsequently, we visit the remaining pixels to also fill in regions with missing depth.

User assistance. In complex scenes, the above approach can fail to detect small objects, or very similar objects (in depth and color) can be wrongly merged. This is particularly a challenge for mid to far objects, where the corresponding depth precision is particularly poor. In such cases, we allow the user to scribble objects as specific segments. From the scribble marked regions, we compute the region's mean color, $\mathbf{c}_\mu$, point normal, $\mathbf{n}_\mu$, and depth value, d_μ. Using region growing, we append neighboring candidate pixels p to the current region if the following conditions hold:

$$|\mathbf{c}_\mu-\mathbf{c}_p|<\lambda_c \text{ AND } d_\mu-d_p<\lambda_d \text{ AND } \mathbf{n}_\mu\cdot\mathbf{n}_p<\lambda_n$$

where, $\lambda_c=65$, $\lambda_d=0.3(d_{max}-d_{min})$ and $\lambda_n=25^\circ$. This rough segmentation is then used instead of the output of the SuperPixel level alpha-expansion, and we continue with the iterative segmentation, primitive fitting and depth map completion at the pixel level as previously described.

Billboarding. Note that we do not require the segmented regions to be planar. While it is possible to work directly with the 3D pointcloud segments, we demonstrate that *billboarding* the pointset is a much simpler and sufficient for many of the target applications (cf., [MSM11]). This drastically simplifies subsequent processing steps while we can still plausibly handle perspective and occlusion effects. However, some segments are not well approximated by a plane and in some cases result in a plane with poor orientation. Hence, we identify the non-planer segments based on the corresponding fitting residue, and 'billboard' them fronto-parallel. Specifically, we assign the points to a plane with a normal facing the camera. This avoids inaccurate planes being fitted to an object, causing issues later in our pipeline. We found for scenes with a large range setting the residue threshold to 2000 best, scenes with medium range 1000 and small range scenes 300.

Occlusion map. Next, we identify which regions on the primitives are occluded by foreground objects. For each pixel we find the 3D point on each primitive using ray-plane intersection. If the point's depth is greater than the associated

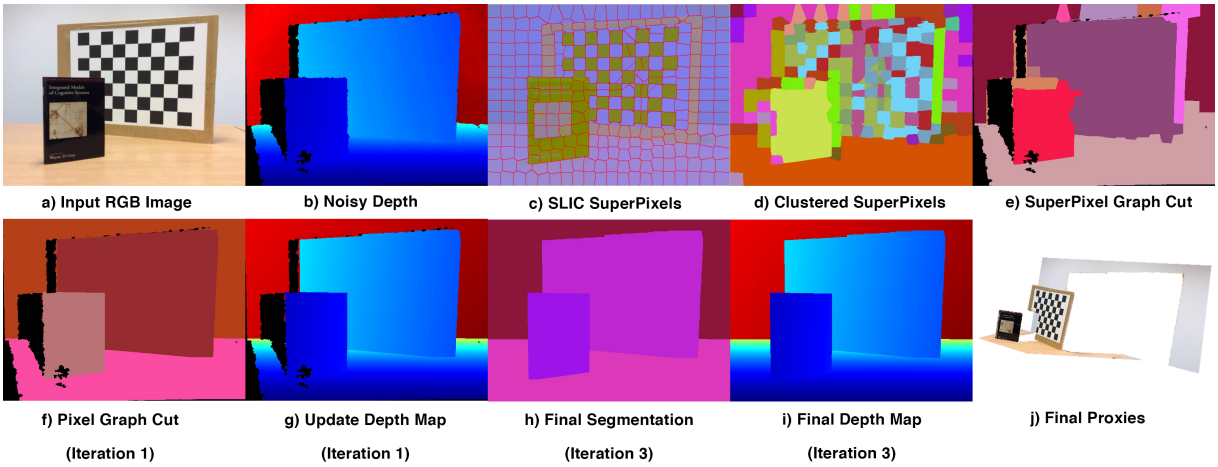

Figure 4: *Scene decomposition: simultaneously performs segmentation, planar primitive fitting, and pixel depth assignment a) Input RGB image b) Input depth map; note incomplete and missing regions c) SLIC Superpixels computed and planer primitives fitting d) Clustering of SuperPixels plane primitives e) Alpha-expansion graph-cut at SuperPixel level f) First iteration of alpha-expansion graph-cut on pixel level g) First iteration of planar primitive fitting and depth map completion h) Final image segmentation i) Final primitive fitting and depth map completion j) Final segmentation and 3D proxies*

value in the completed depth map and the pixel for this segment has no color information (i.e., is not visible in the input image), we mark it as occluded. After searching over all the points in a layer, we test if the marked occluded regions are connected to a region on the primitive that is visible. This removes false positive occlusions when the object has actually ended, but this is not known at the primitive level. We further clip the primitives by extending the visible edges into the occluded regions.The result for each layer are pixels marked as being occluded that need to be infilled.

Fronto-parallel image completion. The final step is to complete the occluded regions. As has been observed by Huang et al. [HKAK14] image completion works better with planer surfaces. As our scene is positioned around the depth sensors optical center, we transform each primitive so it is fronto-parallel with the camera by finding the rotation between the primitive's normal vector and the vector pointing down the negative z-axis. We apply this transformation to the points in 3D, and find the corresponding 2D homography and apply it to the image (see Figure 5). Thus we exploit the planar proxies to obtain fast and light-weight image warping.

To deal with shadows before performing image completion, we grow the depth-occluded pixels slightly to include some visible pixels, removing any shadowing artefacts. We used the Photoshop implementation of PatchMatch [BSFG09] for image completion. We warp the new completed image back to the original pose by applying the inverse rotation.

4.2. Image Degradation Model

We can use the created scene abstraction to propose a simple image degradation model that predicts plausibility of image manipulations. In other words, it provides a confidence score for the quality of the infilled pixels from the previous step and penalize bad ones if they are revealed by proposed image manipulations. The overall degradation of the image is then the sum of pixel-level degradation scores visible in the image. For example, in the case of parallax photography from a single image, the degradation score is zero when the camera is at the origin, and increases as we move further from the original camera pose, however, *not* uniformly in all directions. Hence, the score can inform the user which directions to pursue, and more importantly which ones to avoid.

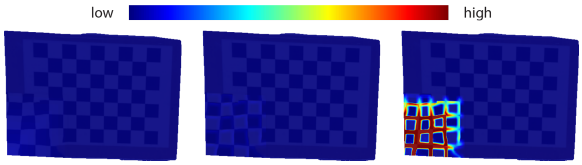

Figure 6: *Degradation Model: The heat-maps visualize the degradation model for one segment. The spatial term (left) gently degrades the further known pixel values. The texture terms (middle) shows greater degradation around the sharp texture boundaries of the checkerboard pattern but low degradation in the centre of squares. The combined final term (right) shows how high textured regions close to known values will receive a moderate score; such regions further from known pixels receive a high score.*

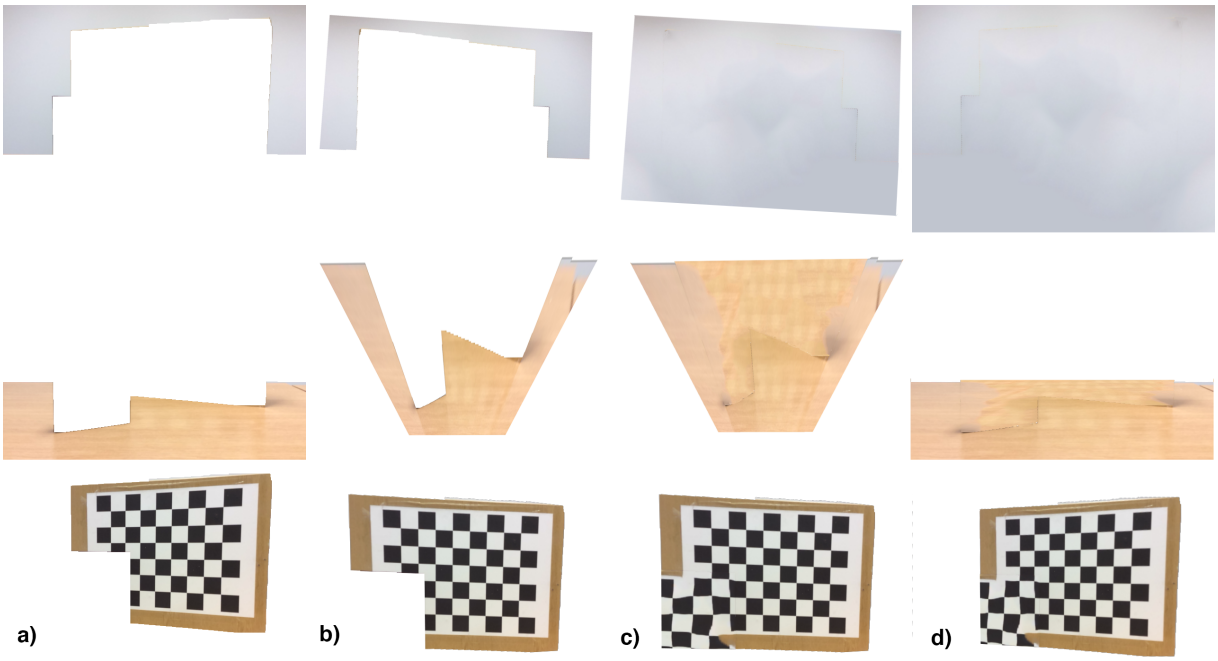

Figure 5: *Image Completion: (a) Segments in their original position from input image (b) Segments made front-parallel to the camera using 2D homography (c) Occluded regions determined and infilled using PatchMatch (d) Infilled segments returned back to original pose.*

Our proposed per-pixel degradation score consists of two terms: a *spatial* term measuring proximity to known pixels and a *texture* term measuring the plausibility of infilled textures.

The spatial term captures the intuition that deeper inside occluded regions, our guesses will have access to less local information for clues. We compute it by using a breadth-first region growing approach starting at the boundary of known and unknown pixels. The boundary grows by adding any of the occluded pixels in the eight-connected neighborhood of the current boundary. We repeat the process until all unknown pixels have been visited. The degradation score is set to 1 for the first layer and increments on each iterations.

The texture term captures the intuition that uniform (or structured) regions are more likely to be plausibly infilled. We compute it using a similar region growing approach. As the boundary region grows the degradation score is the average sum of absolute difference between each pixel and its $(2k+1)\times(2k+1)$ neighbourhood of visited or visible (i.e., known) pixels as:

$$texture(i,j) := \frac{1}{N}\sum_{x=-k}^{k}\sum_{y=-k}^{k}|I(i+x,j+y)-I(i,j)|.$$

where, N is the number of pixels in the neighbourhood that have been visited and the neighbourhood width $k = 10$. We estimate the final degradation score for a pixel simply as product of the two terms. Figure 6 shows an example.

4.3. Novel view synthesis

We can now use the layered texture-infilled planar proxies to generate novel view images, and also score the plausibility of the synthesized view using the proposed degradation model.

In the context of parallax photography, we have to generate a new image for each camera view along a path. The path is defined by the user who selects two key frames parameterized by camera location and rotation; the remaining poses on the path are linearly interpolated. Changing the camera pose is equivalent to applying the same transformation to all of the points in the scene, so our camera actually stays in one place and the scene is moved. To generate the new image equivalent to moving the camera position we warp the planar proxies to a new pose using homographies. To find the homographies we simply transform all the 3D points by the transformation from the original camera pose to the new pose, and project them onto the image plane. We store the points' original positions in the image plane and their new positions. Then, we estimate homographies to map the two sets of points using a RANSAC-based approach from the OpenCV library. Finally, we transform the points back to their original positions. We create the final image com-

posite using the painters algorithm, iterating over the layers, and updating the output image pixel if a lower depth value (closer to camera) is found. Figure 7 shows some example novel views and their degradation models.

5. Results

We evaluate our framework for creating parallax photographs with the degradation providing feedback on the quality of the results. Figure 8 shows both of these in action in a variety of settings. All the scenes are captured using a StructureSensor. Please refer to the accompanying video for full sequences.

Figure 8-kitchen shows how occluded regions can be determined and completed effectively. The book stand, which is partly occluded by the cereal box is reliably infilled (due to fronto-parallel rectification) and still ensures that the background remains visible.

We demonstrate how we can deal with large regions of missing depth, due to range limitation of consumer depth sensors, by allowing segments be approximated by a far away plane in Figure 8-park1 and Figure 8-park2 . In these scenes, we are still able to have a parallax effect with only two proxies in the scene.

The two statues in scenes park1 and park2, the pot in scene kitchen, and the table a chairs in scene office show how

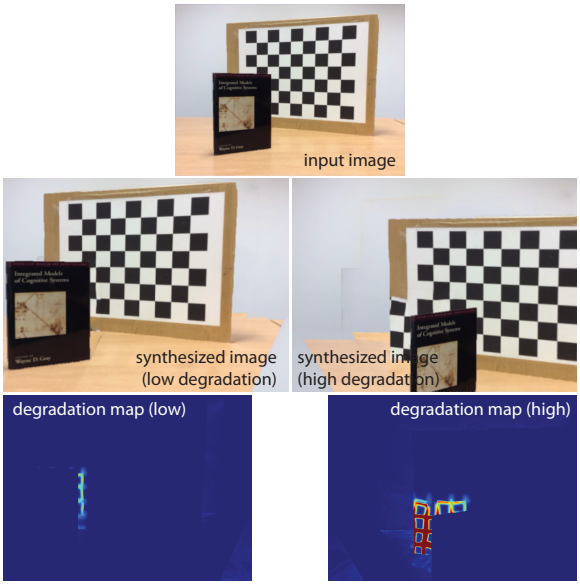

Figure 7: *Novel View Synthesis: Input image (top) is used to create two novels views (middle) with degradation models (bottom). The left example has a low degradation score as the revealed region only has moderate texture and is close to known pixels. The right example has a high degradation score as it reveals a high texture region and far from known pixels.*

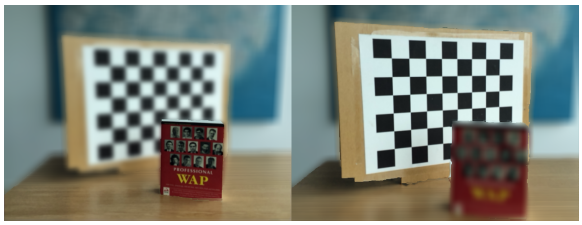

Figure 9: *Depth of Field: A DoF effect can be created using the primitives. The left image shows the first frame of parallax photograph with the book currently in focus. The right image shows the final frame with the checkerboard in focus. Throughout the sequence the camera's depth of field remains the same but as the camera moves forward the object in focus changes.*

non-planar objects can be approximated by a plane primitive. By using the plane fitting error we can identify such objects and set their normal facing the camera and using the segments centroid. This does, however, lead to inaccurate perspective scalings in scene-office. Note that for non-planar objects, we can also add a degradation term for views deviating from fronto-parallel projection.

For each scene, we give examples of synthesized views with low and high degradation scores. Qualitatively, the degradation models captures image blemishes reasonably. For example, in the scene-office, moving the camera too far into the scene reveals a poorly infilled region that gets flagged by a high degradation score. In scene-kitchen, panning right and forward reveals a much smaller segment on high-texture infill, compared to panning right. Similarly with the scene-living room the poorly infilled floor is also flagged by the degradation model.

Some of the scenes required user-interaction for the segmentation step. Figure 8-office required the table legs to be highlighted to ensure the legs were segmented with the table top; Figure 8-kitchen required user interaction to ensure the orange tray was assigned to the back wall, not the wooden stand; and Figure 8-living-room required the sofa and chair to be tagged as separate objects.

Depth of Field. The primitive abstraction and depth can be used in creating a depth of field effect, see Figure 9. The user can control the camera's depth of field by setting a focus depth value and range: pixels within the depth of field remain the same but those outside are blurred with a Gaussian kernel. To get the complete depth of field effect the variance parameter for the Gaussian Filter is made dependent on the difference in each pixel's depth with the depth of field range.

Limitations. Our approach works best when there are only a handful of intersecting primitives. In scenes such as Figure 10 where there are too many intersecting primitives in close depth proximity and appearance, we are unable to segment and fit primitives correctly. The problem is complicated as

Figure 8: *From top to bottom scenes: office, kitchen, living room, park1, park2. In each row, we show the input RGBD image, the abstracted layered scene, novel view synthesis with low degradation (inset showing degradation map), novel view synthesis with high degradation (inset showing degradation map), respectively.*

the noise level in the depth measurement is higher than the depth separation of the scene planes. The initial synthesis looks plausible for small view changes, but when the user makes bigger view change it reveals glaring artefacts breaking the illusion.

We only used planes as proxy geometry in our implementation. While we demonstrated that planes can solve many of the challenges, more complex primitives are likely to provide more interesting results. For example, cylinders, where appropriate, would provide more accurate occlusions and perspective changes as the camera moves.

Figure 10: *Failure Case: For this input scene (left) where there are many intersecting objects we are unable to accurately fit primitives and determine occlusions (right). Without accurate proxies we are unable to correctly complete the scene or synthesise new views.*

6. Conclusion

We have presented a scene abstraction and image degradation model for single RGB-D images. We demonstrated how a variety of objects, or even groups of objects, can be approximated by simple planar proxies created out of rough depth information loosely synchronized with RGB information. These proxies can then be used to determine occlusions in a scene and assist with image completion in these occluded regions. This scene abstraction allows for an image degradation model to be created that captures the confidence in the quality of the image completion step. We use the model to assist the user in performing edits. We demonstrated the use of the degradation model in the context of parallax photography from single images.

In the future, we would like to further explore applications of the image degradation model. One particular area of interest is using it to create a smart interface for image editing. Such an interface could allow the user to perform 3D edits in the scene - by way of geometric proxies - and have the output degradation evaluated. If the degradation is high the system could suggest a similar alternative edit by moving objects in the scene or adjusting the camera pose to one that has less degradation.

Acknowledgements

We thank the reviewers for their comments and suggestions for improving the paper. We would like to thank Moos Hueting and Aron Monszpart for their invaluable comments, support and discussions. This work was supported in part by ERC Starting Grant SmartGeometry (StG-2013-335373) and gifts from Adobe.

References

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SUSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE PAMI 34*, 11 (2012), 2274–2282. 4

[BK01] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE PAMI 26* (2001), 359–374. 4

[BSFG09] BARNES C., SHECHTMAN E., FINKELSTEIN A., GOLDMAN D. B.: Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM SIGGRAPH 28*, 3 (2009), 24:1–24:11. 2, 5

[CAA10] CARROLL R., AGARWALA A., AGRAWALA M.: Image warps for artistic perspective manipulation. *ACM SIGGRAPH 29*, 4 (2010), 127:1–127:9. 2

[CZM*10] CHENG M.-M., ZHANG F.-L., MITRA N. J., HUANG X., HU S.-M.: Repfinder: finding approximately repeated scene elements for image editing. *ACM SIGGRAPH 29*, 4 (2010), 83:1–83:8. 2

[CZS*13] CHEN T., ZHU Z., SHAMIR A., HU S.-M., COHEN-OR D.: 3-sweep: extracting editable objects from a single photo. *ACM SIGGRAPH Asia 32*, 6 (2013), 195:1–195:10. 2

[EPD12] ERDOGAN C., PALURI M., DELLAERT F.: Planar segmentation of rgbd images using fast linear fitting and markov chain monte carlo. In *Proceedings of the 2012 Ninth Conference on Computer and Robot Vision* (2012), CRV '12, pp. 32–39. 2

[HAA97] HORRY Y., ANJYO K.-I., ARAI K.: Tour into the picture: using a spidery mesh interface to make animation from a single image. *ACM SIGGRAPH* (1997), 225–232. 2

[HKAK14] HUANG J.-B., KANG S. B., AHUJA N., KOPF J.: Image completion using planar structure guidance. *ACM SIGGRAPH 33*, 4 (2014), 129:1–129:10. 5

[KSES14] KHOLGADE N., SIMON T., EFROS A., SHEIKH Y.: 3d object manipulation in a single photograph using stock 3d models. *ACM SIGGRAPH 33*, 4 (2014), 127:1–127:12. 2

[KSH*14] KARSCH K., SUNKAVALLI K., HADAP S., CARR N., JIN H., FONTE R., SITTIG M., FORSYTH D.: Automatic scene inference for 3d object compositing. *ACM SIGGRAPH 33*, 3 (2014), 32:1–32:15. 2

[LRL14] LU S., REN X., LIU F.: Depth enhancement via low-rank matrix completion. *IEEE CVPR* (2014), 3390–3397. 2

[MSM11] MCCRAE J., SINGH K., MITRA N. J.: Slices: A shape-proxy based on planar sections. *ACM SIGGRAPH Asia 30*, 6 (2011), 168:1–168:12. 4

[OCDD01] OH B. M., CHEN M., DORSEY J., DURAND F.: Image-based modeling and photo editing. *ACM SIGGRAPH* (2001), 433–442. 2

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: Grabcut: interactive foreground extraction using iterated graph cuts. *ACM SIGGRAPH 23*, 3 (2004), 309–314. 2

[SMZ*14] SHAO* T., MONSZPART* A., ZHENG Y., KOO B., XU W., ZHOU K., MITRA N.: Imagining the unseen: Stability-based cuboid arrangements for scene understanding. *ACM SIGGRAPH Asia* (2014). * Joint first authors. 2

[WCM15] WONG Y.-S., CHU H.-K., MITRA N. J.: Smartannotator an interactive tool for annotating indoor rgbd images. *CGF Eurographics* (2015). 2

[ZCA*09] ZHENG K. C., COLBURN A., AGARWALA A., AGRAWALA M., SALESIN D., CURLESS B., COHEN M. F.: Parallax photography: creating 3d cinematic effects from stills. *Proceedings of Graphics Interface 2009* (2009), 111–118. 2

[ZCC*12] ZHENG Y., CHEN X., CHENG M.-M., ZHOU K., HU S.-M., MITRA N. J.: Interactive images: cuboid proxies for smart image manipulation. *ACM SIGGRAPH 31*, 4 (2012), 99:1–99:11. 2