*i*2*3* - Rapid Interactive 3D Reconstruction from a Single Image

Savil Srivastava, Ashutosh Saxena*, Christian Theobalt, Sebastian Thrun, Andrew Y. Ng

Department of Computer Science, Stanford University, Stanford, CA
*Department of Computer Science, Cornell University, Ithaca, NY
{savil,theobalt,thrun,ang}@cs.stanford.edu, *asaxena@cs.cornell.edu

Abstract

We present **i**23, an algorithm to reconstruct a 3D model from a single image taken with a normal photo camera. It is based off an automatic machine learning approach that casts 3D reconstruction as a probabilistic inference problem using a Markov Random Field trained on ground truth data. Since it is difficult to learn the statistical relations for all possible images, the quality of the automatic reconstruction is sometimes unsatisfying. We therefore designed an intuitive interface for a user to sketch, in a few seconds, additional hints to the algorithm. We have developed a way to incorporate these constraints into the probabilistic reconstruction framework in order to obtain 3D reconstructions of much higher quality than previous fully-automatic methods. Our system also represents an exciting new computational photography tool, enabling new ways of rendering and editing photos. [1]

1 Introduction

Authentic and photo-realistic 3D reconstruction from image data has been a long standing problem in both computer vision and graphics. Recent progress in computational algorithms has made it feasible to reconstruct decent 3D models from multi-view image data. Applications for these are mainly for professionals, which may explain why the related technology has not reached the layman.

At the same time, computational photography research has shown us that amateur photographers could enjoy groundbreaking new photo postprocessing functionality, if only photo cameras would also capture depth information [32]. Unfortunately, single-image 3D reconstruction is a complex problem, and most researchers propose some form of hardware modification to the camera's optics [8,21].

In contrast, we present an interactive algorithm that allows a user to extract 3D models from single images captured by any camera with standard optics. Our algorithm relies on a machine learning approach based on a Markov Random Field (MRF) (Sect. 4) [26, 27]. It phrases monocular 3D reconstruction as the problem of estimating the most likely 3D model given image features. The MRF represents statistical relations between scene depth and image features, learned from a training set of images with ground truth depth. Previous learning-based approaches [10, 27] have aimed at fully-automatic reconstruction that are unable to create plausible 3D models on a general set of images.

To boost quality, we get the user in the loop. While learning algorithms are good at low-level tasks (like segmentation), humans are good at understanding high-level cues, and a combination of the two complements each other to provide a fast and easy way to create 3D models. Specifically, after a first fully automatic 3D reconstruction using the MRF (Sect. 4), the user inspects the result and can optionally provide supporting input to the reconstruction method in the form of simple scribbles and strokes (Sect. 5). The inputs hint, for instance, at coherent 3D structures and likely foreground objects (Fig. 1b,c). We have created a method to feed the user's input back into the probabilistic reconstruction algorithm, and reconstruct a new most likely 3D model given this additional knowledge. The interface we created is simple and intuitive and allows even inexperienced users to create faithful 3D models with minimal effort, Fig. 1d. In experiments with a web prototype, 1238 mostly inexpe-

[1] A preliminary version of this work was informally presented in [25]

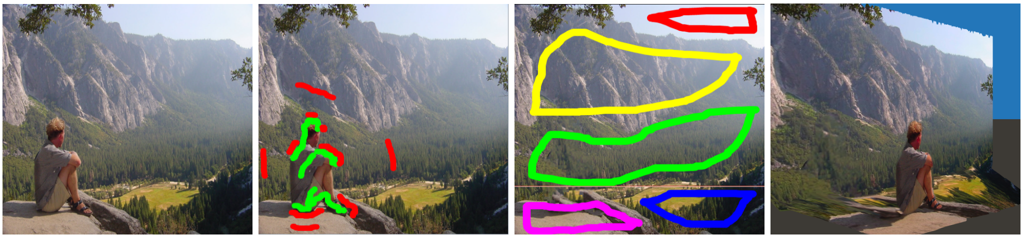


Figure 1: Algorithm Overview: (a) An original image. (b) Interactive tool: user indicates foreground object(s). (c) The foreground object is removed from the image, the background is inpainted, and the user points at an approximate horizon as well as scribbles areas that are on the same 3D planes. A background 3D model is then reconstructed, 3D geometry for the foreground portions are re-inserted, and finally the scene can be rendered from a novel viewpoint as in (d).

rienced users created 3D models for 3775 images using the interactive tool, and 84% of the 3D models created were considered by the users to be good.

The quality of our 3D reconstructions are now sufficient to create new forms of image edits or rendering, previously infeasible with 2D data (Sect. 6). We exemplify the potential of our method as a creative tool for image processing by creating virtual 3D flythroughs from photographs, virtual pan effects in a single image with correct scene parallax, as well as cinematographic camera shots. We can make depth of field modifications, even for images originally taken with small digital cameras lacking the optics to create such effects.

2 Related Work

Monocular 3D reconstruction is harder than multi-view reconstruction since stereo and correspondence cues that are essential for multi-view reconstruction are not available. [24, 28]

Approaches to estimating 3D information from a single viewpoint include acquiring several images of a scene with different camera settings: Depth-from-focus [19], depth-from-defocus [22], and [17, 20]. However, these typically fail on scenes with large depth, or those taken using small aperture lenses of regular cameras. They frequently rely on special active illumination which requires carefully designed illumination pattern removal strategies. [31] reconstruct piece-wise planar objects, as compared to the entire scene, and rely on a user to provide detailed geometrical constraints.

Other approaches include using several images and modifying the lighting settings in between shots: Shape-from-shading approaches [35], and using depth-dependent shading variations with flash [7]. In contrast, our algorithm works on single images and for scenes with large depth where controlled lighting is infeasible.

Fattal [4] estimates depth maps from hazy scenes. However, our method does not require specific environment conditions to work. [29, 30] use multiple images of the same scene found in community photo collections while we aim to reconstruct 3D geometry from a single image.

The optical system of the camera can be modified to encode information on scene geometry in the captured signal. Modifications include a special mirror in the optical path [8], coded aperture masks [13, 34] and micro-lens arrays [6, 21]. Our algorithm does not require any modification of the optical camera path, works for scenes with large depth complexity, and does not sacrifice image resolution.

Our method uses learned statistical relations between image features and 3D scene information. Similarly, Nagai et al. [18] use an MRF-based learning approach to perform single image reconstruction of known objects (e.g. faces), while [23] estimate depth from monocular images using local and global features, and modeling the relation between depths at different points. A different form of statistical relation is exploited by shape-from-texture approaches [15, 16] that perform texture distortion analysis to determine 3D structure. Torralba [33] used the Fourier spectrum of an image to estimate the mean depth. Hoiem et al. [10] classify an im-

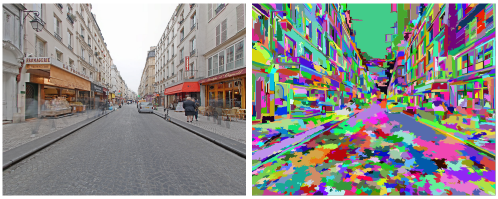


Figure 2: (a) Original image, and (b) superpixels, small segments with uniform color or texture. Our goal is to estimate the 3D location and orientation of the plane on which each superpixel lies.

age into ground and vertical areas to produce photo popups.

The algorithm presented in this paper is an extension of the Make3D method of Saxena et al. [26,27] who propose a learning algorithm to estimate depth from learned statistical relations between image features and 3D structure. Since we apply user-defined hints as constraints during inference, reconstruction quality is dramatically improved over these previous approaches.

3 Algorithm Overview

Our method begins by automatically reconstructing a 3D model from a single image using a machine learning method based on a Markov Random Field (MRF) (Sect. 4). The initial 3D model may be unsatisfying since the algorithm is dependent on the training set, which is unlikely to capture all statistical relations of natural images.

Therefore, we developed an intuitive interface to allow the user to specify additional constraints (see Sect. 5). The sequence of steps differs slightly depending on whether there is a prominent foreground object in the scene (situation I) or not (situation II).

For situation II, if the user is not pleased with the initial reconstruction, the user can first draw simple strokes on the image to denote regions that should lie on similar planes in 3D (Sect. 5.1) and/or indicate the location of a horizon (Sect. 5.2). Given these, a more accurate 3D model is reconstructed.

If situation I applies, the above sequence is preceded with specific steps to handle foreground objects (Sect. 5.3). Simple sparse strokes roughly denote a likely foreground object and background areas. This enables the method to cut out the foreground, inpaint the background image and reconstruct the background geometry using the step for situation II. Previously segmented foreground elements are reconstructed separately and re-inserted into the background model yielding a final high-quality 3D model of the entire scene (see Fig. 1).

The user inputs translate into constraints that can be directly incorporated into MRF reconstruction (such as plane and horizon information), or indicate areas of the scene that merit a special reconstruction strategy (as in the case of foreground objects).

4 Initial 3D Model Reconstruction

We assume the world comprises of small 3D planes that are projected into the image as regions with similar color and texture, called superpixels (Fig. 2, [5]). Our goal is to infer their location/orientation.

We parameterize the plane on which the superpixel lies by $\alpha \in \mathbb{R}^3$, giving its position/orientation. $1/|\alpha|$ is the distance from the camera center to the closest point on the plane, and the normal vector $\hat{\alpha}$ gives the orientation of the plane.

Given α for each superpixel, we create textured 3D models by approximating each superpixel with triangles in the 2D image. 3D triangles are computed by intersecting rays through the 2D triangles' vertices with their respective planes in 3D.

We take a supervised learning approach similar to Saxena et al.'s Make3D to reconstruct an initial 3D model from an image, by exploiting statistical relations between image features and depth. We describe their algorithm's initial steps to enable the reader to understand how we combine it with the user's hints (Sect. 5). For details, see [27].

For each superpixel, we try to infer the parameter α of its 3D plane. Formally, an MRF models α as a function of the image features X. In the MRF, we have terms modeling pairwise relations between the superpixels, and a quantity, y_{ij}, the strength of this relation for each pair. For instance, two neighboring superpixels, like those on the road in Fig. 2 are more likely coplanar if they have similar image features. Implicitly the MRF defines a neighborhood system (graph) on the set of superpixels that enforces 3D relations between immediately adjacent superpixels of the image. Mathematically, our MRF is formulated as

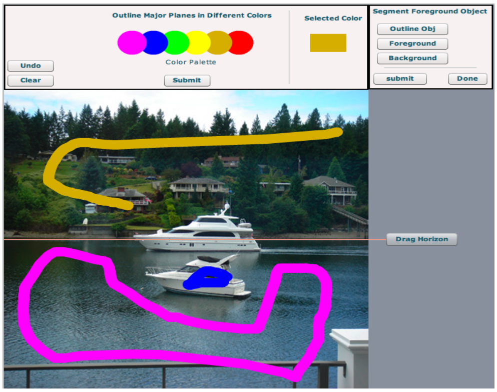


Figure 3: The user interface.

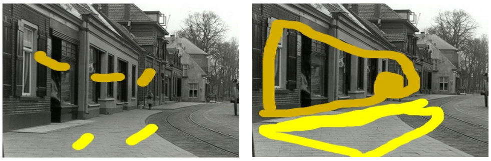
$$P(\alpha|X;\Theta) = \frac{1}{Z}\prod_i f_1(\alpha_i|X_i;\Theta) \prod_{i,j} f_2(\alpha_i,\alpha_j,y_{ij};\Theta) \tag{1}$$

Thus we infer the maximum-a-posteriori (MAP) set of positions and orientations of 3D surfaces $P(\alpha|X;\Theta)$ given the set of model parameters Θ (learned from labelled ground-truth data) and features extracted for each superpixel, X_i.

The first term in Eq. (1) models depth and orientation as a function of the image features. The second term $f_2(.)$ statistically models the relations between two superpixels such as coplanarity, connectivity and colinearity.

Here, $y_{ij} = f(X_i, X_j; \Theta) \in \mathbb{R}^+$ is the weight of the particular pair-wise term (a larger weight implies a stronger relation). A y_{ij} exists for all pairwise relations, and is dependent on the image features. Parameters Θ weigh the influence of each image feature on the relation between features and depth, captured by f_1, and the strength of each pairwise constraint, captured by f_2.

Fully-automatically reconstructed 3D models of images that are very different from the training data are typically of poor quality. Designing a training set to capture all possible natural variation is impossible. Instead, we propose to get the user into the loop, and incorporate user input into our probabilistic framework.



(a) Small rough strokes (b) Exact strokes

Figure 4: Examples for the variability of user scribbles on the same image. Large parts are occupied by two major planes: ground and the vertical walls.

5 User Interface

We designed an intuitive user interface for even non-expert users to give guidelines to the 3D reconstruction algorithm in a few seconds, Fig. 3.

First, the user can roughly scribble on an image to give high-level cues about which parts of the image belong to the same 3D planes (Fig. 4). Areas scribbled with the same color are to belong to one 3D plane. In the automatic algorithm, areas of similar color (e.g., a uniformly painted wall) are considered more likely to be coplanar, but it may fail to infer that the windows belong to the same 3D plane, which even a partial scribble would help fix. Formally, the algorithm would increase y_{ij} wherever two neighboring superpixels share the same scribble color (Sect. 5.1).

Second, the user can drag the horizon line to the correct location (Sect. 5.2). This cue enables the learning algorithm to reuse parameters that were learned on a subset of training images having the horizon line in roughly the same location.

Third, the user can make small strokes indicating a foreground object and the scene's background. Our motivation for this work was the difficulty in training a learning algorithm for all possible foreground objects and the observation that even small mistakes in such objects are glaring to users. In our UI, the user gives a high-level cue by scribbling a few strokes to indicate the foreground object allowing the algorithm to plausibly infer the foreground geometry (Sect. 5.3).

5.1 Scribbles for Coplanar 3D Segments

The user is to mark the parts of the image belonging to the same 3D plane, with the same color. Each pixel should have one color/plane assignment. It is not necessary to color each individual pixel manually, since a plausible color assignment from sparse approximate strokes can be inferred. This high-level cue modifies y_{ij}. Intuitively, if we know that two superpixels belong to the same 3D plane, we can increase the value of y_{ij} in Eq. (1) to increase the effect of the connectivity and coplanarity terms. The function $y_{ij} = f(X_i, X_j, U_i, U_j; \Theta)$ has to weigh the user input U_i and U_j against the feature evidence from the images, X_i and X_j.

When designing f we also consider that input from inexperienced users may contain errors: coloring one 3D plane with two colors, or having imprecise strokes that bleed into an incorrect area. Using these as a hard constraint would give poor results.

User scribble representation: Assuming K colors, $U(x, y) \in \mathbb{R}^K$ represents the scribble color at each point (x, y) in the image. If the user draws a loop without any other color inside it (Fig. 4b), then we "fill" the "holes" with the same color.

Strokes from different users can vary from sparse to precise even on the same image (Fig. 4). Given the scribble color that superpixels i and j have, our goal is to infer y_{ij}. Three situations arise:

1. If i and j have the same color, y_{ij} should be high.
2. If i and j have different colors, y_{ij} should be low.
3. One or both superpixels have no color (were not scribbled by the user). This corresponds to the tricky case where the user expects the algorithm to complete the grouping (Fig. 4a). We want y_{ij} to be a function of the distance from the nearest stroke. If the superpixels have nearby strokes of same color, then y_{ij} should be larger compared to when they have strokes further away. This weighs the input from the user as well as evidence from the images.

This weighting strategy translates into a blurring function Ω, which we convolve with $U(x, y)$ to produce a blurred scribble image $V(x, y)$

$$V(x, y) = \sum_{p=-\Delta x}^{\Delta x} \sum_{q=-\Delta y}^{\Delta y} U(x+p, y+q) \quad \Omega(p, q) \tag{2}$$

Finally, we define V_i for superpixel i as the mean of each of the values at each pixel. We now change y_{ij} in Eq. (1) to incorporate the user input $V \in \mathbb{R}^K$:

$$y_{ij} = f(X_i, X_j; \Theta)/||V_i - V_j||_2 \tag{3}$$

$f(X_i, X_j; \Theta)$ is a logistic function of the image features X_i and X_j similar to that in [27].

5.2 Horizon Information

The horizon location plays an important role in estimating the 3D structure of a scene. For example, blue color at the top of the image is more likely sky, while at the bottom is likely water. The relation of depth to the image features varies as a function of the vertical distance from the horizon. The same feature (e.g. blue color) can sometimes mean different depths which makes learning hard when training data is limited. Therefore, we propose a new parametrization of the weights Θ in the model, in which we explicitly consider the location of the horizon. Assuming the horizon is horizontal, we represent its location by its image row $h \in [-1, 1]$.

Specifically, Ψ encodes a MRF parametrization for each row r, assuming a "standard" horizon line in the center of the image, $h = 0$. Given an image with a horizon line that is not in the standard location, we have to decide which MRF parameters to use for each row r. Therefore, we define a transformation that maps the $\Theta(r|h)$ into $\Psi(r')$. For instance, if the horizon is at the bottom one-fourth of the image ($h = -\frac{1}{2}$), then we want Θ at this horizon to be equal to Ψ at its reference horizon. More formally, the mapping from Θ to Ψ is

$$\begin{aligned} \Theta(r|h) &= \Psi\left(\frac{r-h}{1-h}\right), \quad r > h \\ &= \Psi\left(\frac{r-h}{1+h}\right), \quad r < h \end{aligned} \tag{4}$$

Now, even if the horizon was always in one location during training (e.g., we use the data set of [27], where all the images have the horizon in the center), we would effectively train the horizon-normalized weights Ψ. During inference, we can now use the horizon-corrected MRF parameters Θ.

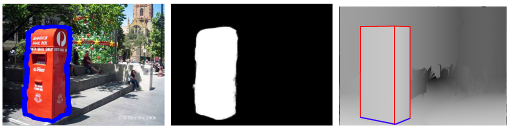


Figure 5: Handling foreground objects: (a) original with segmentation outline, (b) alpha-matte of foreground object, (c) depth map with impostor geometry in the foreground. The red and blue strokes were added for better visibility of the billboards.

5.3 Foreground Objects

Automatic methods [10,27] make most reconstruction mistakes for objects in the scene foreground, see Fig. 6. These are often the focus of attention, and results where they are ”pasted” on the ground (see video), are visually unacceptable.

Therefore, we treat the foreground separately and change the reconstruction process: First, the user provides sparse strokes on the images to roughly indicate foreground and background regions (Fig. 1b), These serve as input to a graph-cut segmentation method [14] which separates out the foreground region. The user provides additional strokes and iterates the segmentation. Alternatively, if the image’s color distribution is challenging for graph-cut segmentation, the foreground object’s boundary can be manually traced (Fig. 5a).

Next, we create an alpha matte for the foreground region (Fig. 5) by assuming a trimap area around its outline and performing Bayesian matting [1]. Quality foreground object occlusion boundaries, including those of fuzzy materials like hair and fur, enable convincing novel viewpoint renderings and image postprocessing.

Foreground elements can now be removed from the image, and their 3D geometry can be estimated separately and re-inserted into the scene. First, however, we need to in-paint the parts of the image behind the foreground objects, which will be exposed in 3D renderings from new viewpoints (Fig. 1c) by performing exemplar-based inpainting [2]. On the inpainted background image, we now employ the previously described interactive 3D reconstruction to obtain a background 3D model.

Foreground objects can now be re-inserted into the background model by approximating their geometry as a collection of planar billboards (Fig. 5c). Shape and position of these billboards in 3D are estimated from the bottom section of the foreground element’s outline curve (blue line) by piecewise linear segments. Each such segment defines one billboard (two in Fig. 5c, one in Fig. 1). Assuming all billboards are upright, we trace rays through the end-points of bottom curve segments and intersect them with the background geometry to get their 3D positions. By projectively texturing the alpha-matted foreground images onto the 3D billboards, one convincingly renders the entire 3D model.

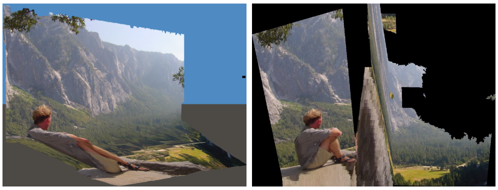


Figure 6: Results of prior art for the image in Fig. 1a. (a) Make3D [27], (b) Photo-Popup [12]

6 Results and Applications

Here, we report our tests illustrating the improvement over prior art (Sect. 6.1), as well as the result of an internet user study for the interactive reconstruction (Sect. 6.2), and show why our system is an exciting computational photography tool (Sect. 6.3).

6.1 Rapid High-quality Reconstruction

Fig. 8 shows snapshots of 3D models produced by **i**23 and the input images. In the postbox and the boat images (left and bottom-left), **i**23 placed the foreground objects correctly (else they would appear distorted when viewed from such drastically novel viewpoints), and realistically filled in the occluded parts to create a visually-pleasing effect. Our algorithm creates good occlusion and parallax effects such as in the waterfall and the ski-slope image. However, **i**23 is not flawless: the model of Prague (Fig. 8, bottom right) is satisfactory but the ground and statue geometry isn’t perfect.

We also compare our performance with fully automatic single-image reconstruction methods by [27] and [12] (Fig. 6). If a foreground object is present, our method clearly performs better. Further, we experimentally validated that even state of the art Cascaded Classification Models [9] that combine inputs from object detectors [3] and occlusion boundary detectors [11], fail to produce decent 3D models on images with foreground objects.

6.2 User Experiments

To better understand the working range of our system and the improvement over fully-automatic methods, parts of the interface (scribbling and horizon line specification) were used for an online ex-

Table 1: Results on 300 random images, with an experienced user providing input.

User input led to:	No. of Images	%
bad result	0	0.0
no difference	11	3.7
some improvement	73	24.3
significant improvement	216	72
Total	300	100

periment. Despite not being a controlled experiment, we obtained insightful answers from a large user group to two questions: 1) Are the results generally perceived as good, and 2) how much of a improvement can be achieved through user input?

Users could upload any image, although a note mentioned that the system performs better on images of “environments“ than objects. After seeing the initial 3D model, users gave inputs ranging from one small stroke to extensive strokes, e.g. for the waterfall image (video). Anyone could vote a “Thumbs-up“ or “Thumbs-down“, but each user was restricted to one vote per model.

Here, 1238 (inexperienced) users created 3D models for 3775 images and 84% of models were rated good. Some results in this paper are from this experiment: the waterfall, the Reichstag, Prague and ski slope pictures (Fig. 8). Admittedly, this assessment is not indicative of absolute reconstruction accuracy, but answered our first question: users are generally pleased with the visual quality.

To remove the user’s experience as an influencing factor, we did a second experiment with a single user having moderate experience. The user was provided 300 random images with poor initial reconstruction (adjudged by online voters), and 50 images with good models (from the Make3D database [27]). The user spent 30 minutes providing input to 350 images (under 5 sec/image). He then compared the result against the automatic reconstruction based on four discrete grades. Table 1 and Table 2 summarize the results. Notably, 72% of random images were significantly improved and 96.3% (= 72% + 24.3%) were convincing. We also see that user input helps, even if the initial reconstruction is good. These experiments indicate that **i**23 faithfully constructs pleasing 3D models.

Table 2: Results on 50 images with good initial 3D reconstruction, with an experienced user providing input.

User input led to:	No. of Images	%
bad result	0	0.0
no difference	12	24.0
some improvement	34	68.0
significant improvement	4	8.0
Total	50	100

6.3 Picture Display and Photographic Postprocessing

The 3D information enables display and editing in ways previously impossible for images from regular cameras (see video): 3D flythroughs, *Ken Burns Effect*, *Dolly Zoom* and depth-of-field modifications. Dramatic 3D flythroughs provide an immersive experience of the spatial structure of the scene (Fig. 8). The *Ken Burns Effect*, where the camera slowly pans and zooms over a static image (used in documentaries), can be created *perspectively-correct* with actual depth and parallax. Cinematographic techniques like the *Dolly* or *Hitchcock Zoom*, where the virtual camera moves towards or away from the scene and concurrently the zoom is adjusted so that a certain scene element remains unchanged, can now be created. This creates a dramatic perspective distortion of the peripheral scene. Advanced editing such as selectively adjusting the depth-of-field and refocussing to any distance can be achieved due to the per-pixel depth information (Fig. 7).

7 Discussion and Conclusion

Feedback to the user isn’t immediate, but computation times are reasonable: computing features (50-60s, once per image), model inference with scribbles (5-6s), foreground segmentation (10s), matting (30s) and inpainting (10-160s depending on resolution); on a Quad Core 2.4 GHz with 4GB RAM, with unoptimized MATLAB code.

Despite convincing results, our approach has some limitations. On difficult scenes, foreground segmentation may need several attempts for satisfying results. On scenes with similar foreground and background color, Bayesian matting may not

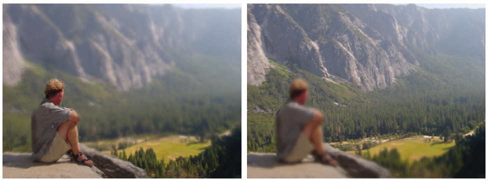


Figure 7: Virtual image refocusing: (a) focus on foreground, (b) focus on background.

always create an accurate matte. In practice, these artifacts are not strongly noticeable.

We also make some heuristic assumptions. Our placement method for foreground billboards is effective in general but sometimes incorrect (e.g. a scene with no obvious ground area). Our 3D models are *plausible*, not metrically correct: important spatial relations between 3D scene elements are captured approximately, permitting faithful novel viewpoint rendering, but sometimes leading to visual artifacts: in Fig. 1, the tree branches are part of the background. Normally, these are barely noticeable.

Despite these limitations, in this paper we presented an effective new algorithm to reconstruct convincing 3D models from a single image. It is based on a fruitful combination of a learning-based reconstruction method with supporting input from the user. We have shown that our algorithm enables even inexperienced users to rapidly create realistic models that are superior in quality to previous methods, and enable advanced image display and editing capabilities.

Acknowledgments

We thank Nuwan Senaratna and Gabriel Yu for their insights and help with experiments.

References

[1] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[2] A Criminisi, P Perez, and K Toyama. Object removal by exemplar-based inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[3] N Dalai, B Triggs, I Rhone-Alps, and F Montbonnot. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[4] Raanan Fattal. Single image dehazing. In *ACM's Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, 2008.

[5] Pedro S. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. In *International Journal of Computer Vision (IJCV)*, volume 59, 2004.

[6] T Georgeiv, K. Zheng, B. Curless, D. Salesin, S. Nayar, and C. Intwala. Spatio-angular resolution tradeoffs in integral photography. In *Proc. Eurographics Symposium on Rendering (EGSR)*, 2006.

[7] Mashhuda Glencross, Gregory Ward, Caroline Jay, Jun Liu, Francho Melendez, and Roger Hubbold. A perceptually validated model for surface depth hallucination. In *ACM's Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, 2008.

[8] Paul Green, Wenyang Sun, Wojciech Matusik, and Frédo Durand. Multi-aperture photography. In *ACM's Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, page 68, 2007.

[9] Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. Cascaded classification models: Combining models for holistic scene understanding. In *Neural Information Processing Systems Conference (NIPS)*, 2008.

[10] D. Hoiem, A.A. Efros, and M. Hebert. Automatic photo pop-up. In *ACM's Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, 2005.

[11] D. Hoiem, A.N. Stein, A.A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. *International Conference on Computer Vision (ICCV)*, 2007.

[12] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision (IJCV)*, 80(1), 2008.

[13] A. Levin, R. Fergus, F. Durand, and W.T. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM's Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, page 70. ACM, 2007.

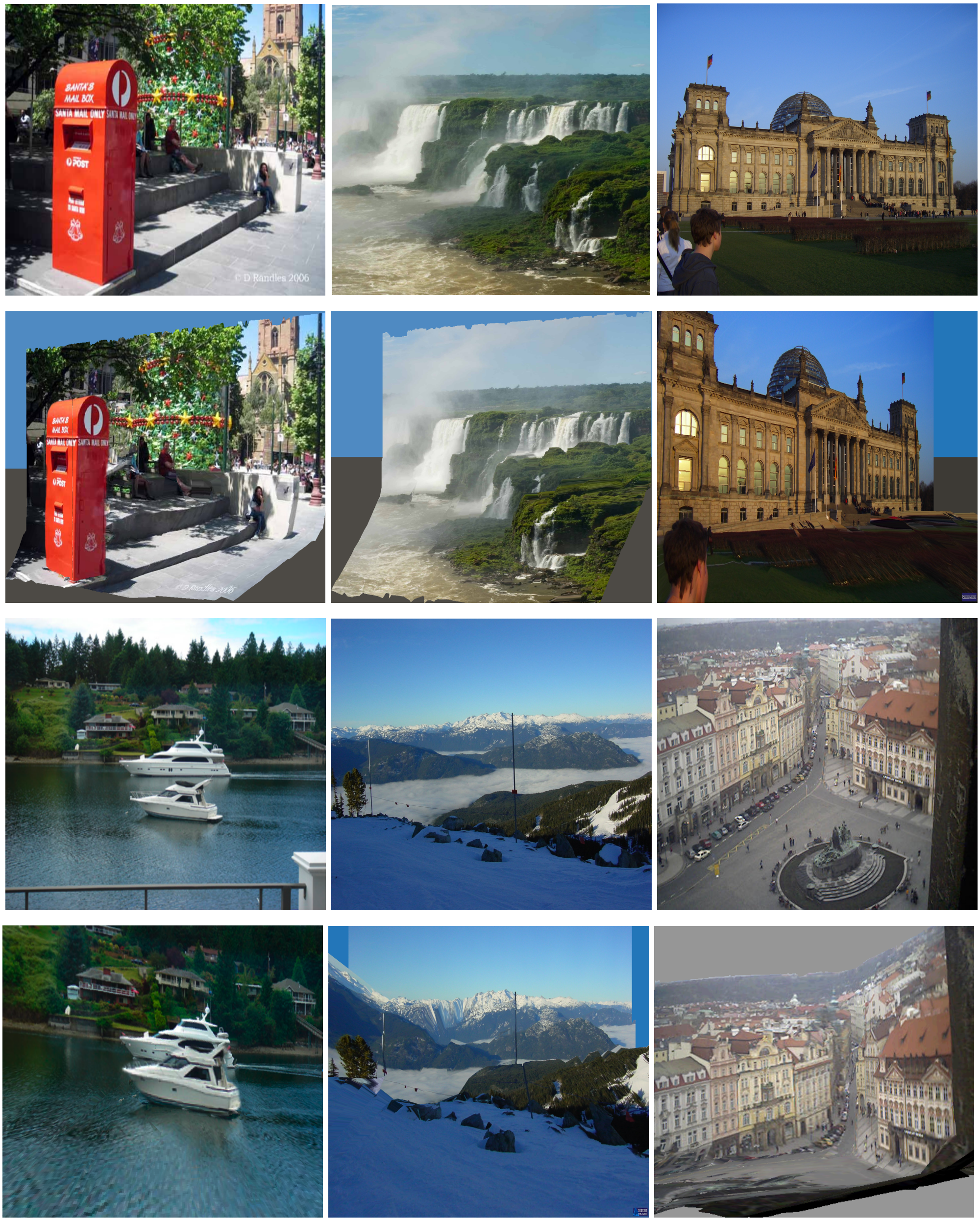


Figure 8: Input images and new viewpoints on the models created with **i**23 – the postbox and boat scenes (top and bottom left) were created by the authors; the waterfall model (top middle), the Reichstag model (top right), the ski slope (bottom middle) and the model of Prague were created by users on the internet.

[14] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *ACM's Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, 2004.

[15] T. Lindeberg and J. Garding. Shape from texture from a multi-scale perspective. In *International Conference on Computer Vision (ICCV)*, 1993.

[16] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision (IJCV)*, 23(2):149–168, 1997.

[17] F. Moreno-Noguer, P.N. Belhumeur, and S.K. Nayar. Active refocusing of images and videos. *ACM Transactions On Graphics (TOG)*, 2007.

[18] T. Nagai, T. Naruse, M. Ikehara, and A. Kurematsu. Hmm-based surface reconstruction from single images. In *Proc. ICIP*, volume 2, 2002.

[19] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 16(8):824–831, 1994.

[20] S.K. Nayar, M. Watanabe, and M. Noguchi. Real-time focus range sensor. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 1996.

[21] R Ng, M Levoy, M Brdif, G Duval, M Horowitz, and P Hanrahan. Light field photography with a hand-held plenoptic camera. Technical report, Stanford, 2005.

[22] A. P. Pentland. A new sense for depth of field. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 1987.

[23] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In *Proc. NIPS 18*, 2005.

[24] Ashutosh Saxena, Jamie Schulte, and Andrew Y. Ng. Depth estimation using monocular and stereo cues. In *International Joint Conference on Artificial Intelligence(ICJAI)*, 2007.

[25] Ashutosh Saxena, Nuwan Senaratna, Savil Srivastava, and Andrew Y. Ng. Rapid interactive 3d reconstruction from a single still image. *SIGGRAPH Late Breaking Research*, 2008.

[26] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Learning 3-d scene structure from a single still image. In *ICCV workshop on 3D Representation for Recognition*, 2007.

[27] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3d: Learning 3-d scene structure from a single still image. In *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, volume 30, pages 824–840, 2008.

[28] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[29] Noah Snavely, Rahul Garg, Steven M. Seitz, and Richard Szeliski. Finding paths through the world's photos. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, 2008.

[30] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 835–846, 2006.

[31] Peter F Sturm and Stephen J Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single image. In *British Machine Vision Conference*, 1999.

[32] Corey Toler-Franklin, Adam Finkelstein, and Szymon Rusinkiewicz. Illustration of complex real-world objects using images with normals. In *Proc. of Symposium on Non-photorealistic Animation and Rendering (NPAR)*, pages 111–119, 2007.

[33] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 24(9):1–13, 2002.

[34] Ashok Veeraraghavan, Ramesh Raskar, Amit Agrawal, Ankit Mohan, and Jack Tumblin. Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. In *ACM's Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, page 69, 2007.

[35] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 21(8):690–706, 1999.