Peeking Template Matching for Depth Extension

Simon Korman
Tel-Aviv University

Eyal Ofek
Microsoft Research

Shai Avidan
Tel-Aviv University

Abstract

We propose a method that extends a given depth image into regions in 3D that are not visible from the point of view of the camera. The algorithm detects repeated 3D structures in the visible scene and suggests a set of 3D extension hypotheses, which are then combined together through a global 3D MRF discrete optimization. The recovered global 3D surface is consistent with both the input depth map and the hypotheses.

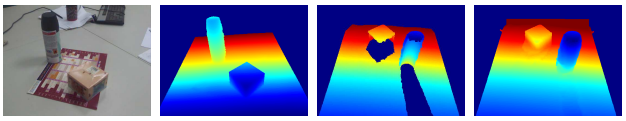
A key component of this work is a novel 3D template matcher that is used to detect repeated 3D structure in the scene and to suggest the hypotheses. A unique property of this matcher is that it can handle depth uncertainty. This is crucial because the matcher is required to "peek around the corner", as it operates at the boundaries of the visible 3D scene where depth information is missing. The proposed matcher is fast and is guaranteed to find an approximation to the globally optimal solution.

We demonstrate on real-world data that our algorithm is capable of completing a full 3D scene from a single depth image and can synthesize a full depth map from a novel viewpoint of the scene. In addition, we report results on an extensive synthetic set of 3D shapes, which allows us to evaluate the method both qualitatively and quantitatively.

1. Introduction

The popularity of depth cameras, such as the Microsoft Kinect, makes depth maps accessible to all. These depth maps are used for a variety of applications such as gesture recognition and 3D modeling. A depth map assigns a depth value to each pixel, generating a 2.5D representation of the visible scene. The resulting depth map typically lacks measurements of scene parts that are occluded from the camera point of view. Many applications, such as path planning, audio waves progress analysis and new view generation, to name few, require an access to the three dimensional data of the full scene, including the surfaces that are not visible by the depth camera.

In this work we make an attempt at inferring the entire invisible structure of a scene, which is an important under-investigated problem in 3D vision. Figure 1 shows an example of what we call a depth *extension* or depth *outpainting* task. An input depth image (b) is rotated by 180° revealing missing parts of the geometry, not seen by the camera. Our method offers a complete solution (d) to such a scene.



(a) RGB (not used) **(b)** depth map **(c)** New view **(d)** Our completion

Figure 1. **The '*Spray*' new viewpoint synthesis example.** See Fig. 9 for additional examples and the text for details.

Our Depth Extension algorithm follows a simple scheme. We generate multiple volumetric hypotheses to extend the current depth map, and these hypotheses are later all merged together. There are several unique challenges we faced while tackling this problem, which gave rise to the contributions of this work.

Novel 3D template matching with partial data To generate depth extension hypotheses, we rely on the existence of repeating local 3D structures in the scene, similar to the assumption made in 2D image inpainting. A depth camera provides us with limited information about the volumetric data, as each visible surface point occludes an unknown amount of solid matter. Generating 3D completion hypotheses therefor involves matching data with partially unknown values, which is an inherently ill-posed problem. To overcome this challenge, we present a novel and fast template matching algorithm, that can match volumetric regions with partial information, under 3D Euclidean transformations. The matching scheme is based on a rigorous analysis of the uncertainties that emerge due to the missing values and it leverages both known spatial data, as well as bounds we derive on the possible errors in areas with uncertainty. This scheme may also be applicable to other problems where matching under partial information is necessary.

Recovery of scene geometry using 3D hypotheses This new matching scheme enables us to detect repetitions in the scene. These repetitions are used to map sub volumes to target locations and generate a set of 3D hypotheses, which represent plausible extensions of the visible geometry. We present an optimization algorithm that recovers a full scene geometry that is as consistent as possible with both the input

depth map and the set of generated hypotheses.

Finally, we report results on a number of scenarios that demonstrate the effectiveness of the proposed method. These include both analyzing simulated views of known geometries and comparing the completion results to ground truth, as well as the processing of actual depth maps captured by a Kinect depth camera.

2. Background

Depth *inpainting* for filling holes in depth maps has been an active research topic in recent years. Torres-Mendez and Dudek [18] proposed a method for the reconstruction of 3D models from intensity images and partial depth, propagating depth information based on intensity values. Later, Wang *et al.* [19] proposed a stereoscopic inpainting algorithm that synthesizes depth and color using a stereo camera pair. Recently, Shen and Cheung [17] proposed a depth layers approach for handling scenes consisting of a static background and dynamic foreground objects, strongly exploiting the correlation between color and depth.

Common to these methods is that the set of hypotheses considered is limited to 2D or 2.5D proposals. Also, they fill holes within the (incomplete) input depth image itself, using complete RGB information. In contrast, we use only depth information and generate a much richer set of hypotheses, directly in 3D, that extend to new viewpoints of the scene.

Our work borrows from the field of image inpainting. Most notably, we are inspired by the work of He and Sun [4]. They observed that the statistics of matching patches in an image can be sufficiently described by a fairly small number k of possible shifts. In the inpainting process, each pixel is chosen from one of k respectively shifted versions of the image, while minimizing a global energy function.

Guo and Hoiem [3] propose a method for predicting support surfaces in indoor scenes from an RGBD image. They use a training set to label visible surfaces and then infer occluding surface labels using contextual cues and layout priors. Kim *et al.* [9] acquire the 3D structure of indoor environments. Their system uses scans of individual objects to construct primitive-based 3D models, which are then quickly recognized in a new scan. Pauly *et al.* [14] recover complete and consistent 3D models from incomplete surface scans using a database of 3D shapes that provide geometric priors for regions of missing data. These methods achieve impressive results in understanding the 3D structures of scenes, largely due to the use of tailored databases and efficient ways of finding shape occurrences in the scene. The completion scheme of Silberman *et al.* [12] is dedicated to uncovering the geometry of rooms, by completing primitives such as bounding walls and planar furniture.

Zheng *et al.* [21] recover solid 3D primitives from a point cloud. Their algorithm uses geometric reasoning to fit simple surfaces to a point cloud, and these are then interpreted as simple 3D shape primitives. Physical reasoning is then used to group the primitives into stable objects.

Another work related to ours is the context-based surface completion of Sharf *et al.* [16]. Defective regions, which are automatically detected as surface areas of low density in a point-cloud, are filled by copying patches with similar signatures from valid surface regions, achieving realistic results. Poisson Reconstruction [7, 8] and its extensions provide a widely used tool for converting a point-cloud to a smooth and highly detailed reconstructed surface. A recent work by Shan *et al.* [15] further constrains Poisson reconstruction through the detection of occluding contours in a multi-view stereo setup. These methods were all designed to provide accurate reconstructions of the *captured* part of a scene, which might be noisy and contain gaps and holes, but were not meant for the task of reconstructing the entire *unseen* part of a scene, in which the holes to be filled are much larger, with far less relevant data to use.

3. Method

In order to infer a 3D scene from a single depth map, our goal is to detect repeated sub-volumes and use their extended surroundings to extend the depth map into unseen parts of the scene. This raises two questions. The first is how to efficiently detect repeated sub-volumes in the scene, and the second is how to merge the different extensions (termed hypotheses) into a single and coherent result.

As for the first question, we extend the visible surface areas of the scene into its unseen surface areas, using a novel 3D template matcher. This is done by detecting multiple template sub-volumes, on the visible surface, and then searching for similar target volumes under the group of rigid Euclidean 3D transformations (including combinations of translations, rotations and reflections). Once such a transformation is found, the surface points in a larger vicinity of the 3D-template are mapped according to the transformation to form a potential hypothesis into the unseen areas. A key problem we face is that the 3D matcher must operate on the boundaries of the visible depth map where, by nature, there are large amounts of missing data. To address this issue, the template matcher uses novel scoring scheme over a scene representation, which is explicitly designed to take into account the uncertainty in the data.

As for the second question, merging the hypothesis proposals into a coherent result can be very challenging, especially in the cases where the unseen surface area is large compared to (or even larger than) the visible surface. In such cases we obtain a large number of hypotheses, which are typically inconsistent with each other and might not even completely 'cover' the unseen surface area. We therefore search for a surface that interpolates between the visible surface areas, in a way that agrees with as many hypotheses as possible and which produces a smooth as possi-

ble surface. This idea is formulated as a binary optimization problem on a 3D raster.

3.1. Volume Representation

Let $\mathcal{S} : \mathbb{R}^3 \mapsto \{0, 1\}$ denote the generally *unknown* binary indicator function of a *scene* (which equals to 1 in the interior). The scene *surface* is represented by a binary indicator function $\partial\mathcal{S}$, the $\frac{1}{2}$-sub-level set of $\mathcal{S}$. The *truncated-signed-distance-transform* (TSDF) representation of $\mathcal{S}$, with truncation parameter k, is given by:

$$\mathcal{A} = (-1)^{\mathcal{S}} \cdot \min\{k, DT(\partial\mathcal{S})\} \tag{1}$$

where $DT(\cdot)$ is the standard (non-signed) Euclidean Distance Transform. The three representations $\mathcal{S}$, $\partial\mathcal{S}$ and $\mathcal{A}$ are equivalent and one can switch between them easily, but unfortunately, they are not known to us.

On the other hand, we have access to the binary indicator function $\mathcal{V}$ of the *visible-volume* that equals 1 in non-occluded areas (i.e. free-space areas and the visible part of the surface $\partial\mathcal{S}$). Knowing $\mathcal{V}$ is equivalent to knowing the *visibility-boundary*, represented by a binary indicator function $\partial\mathcal{V}$, the $\frac{1}{2}$-sub-level set of $\mathcal{V}$, which is the 'boundary' between the visible volume and the occluded area.

Our goal is to reconstruct the unknown scene $\mathcal{S}$ (or surface $\partial\mathcal{S}$). As mentioned before, we do this by finding correspondences between sections of the volume, some of which we have partial information about. The most standard way of scoring a mapping of a sub-volume X under a transformation T is by the symmetric difference between the source and target areas of the mapping. Formally:

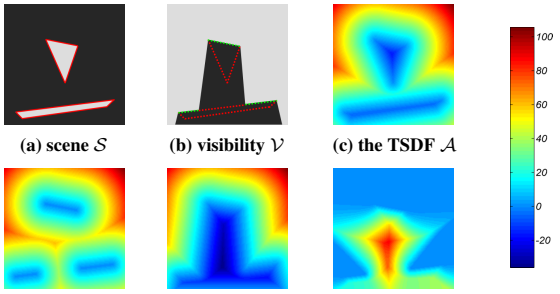
$$Score = \int_X |\mathcal{S}(x) - \mathcal{S}(T(x))| dx \tag{2}$$

Distance-transform representations (signed and/or truncated) have been shown in the past to be suitable for registering and fusing depth images. They were introduced in [2] and were later successfully used, e.g., in the Kinect-Fusion system [13]. Particular advantages are their probabilistic interpretations [5] and the ease of extracting an explicit surface, through the their zero-crossing.

In this work, we build on the Fast-Match method [10] for efficient matching of image templates, generalizing it to handle 3D volumetric templates. The method's runtime complexity (see [10]) depends on the total-variation (or smoothness) of the template representation and it is well known that TSDF representations lead to smoother templates, compared to indicator representations.

These facts motivate us to replace the binary shape representation $\mathcal{S}$ from Equation (2) using the TSDF representation $\mathcal{A}$ from Equation (1) and hence we obtain:

$$Score = \int_X |\mathcal{A}(x) - \mathcal{A}(T(x))| dx \tag{3}$$



(a) scene $\mathcal{S}$ **(b)** visibility $\mathcal{V}$ **(c)** the TSDF $\mathcal{A}$

(d) upper bound $\mathcal{U}$ **(e)** lower bound $\mathcal{L}$ **(f)** Uncertainty $\mathcal{U}$-$\mathcal{L}$

Figure 2. **Upper and Lower bounds: (a)** An *unknown* 200×200 2D scene $\mathcal{S}$, where the camera is located above the top side at $(100, -100)$ looking down. The interior (gray) and surface (red). **(b)** The *known* visibility $\mathcal{V}$ (gray), the visible surface $\mathcal{V} \cdot \mathcal{S}$ (green), the visibility boundary $\partial\mathcal{V}$ (the boundary between gray and black) and the *unknown* occluded surface (red). **(c)** The *unknown* TSDF $\mathcal{A}$. **(d)** The *known* upper bound $\mathcal{U}$. **(e)** The *known* lower bound $\mathcal{L}$. **(f)** The uncertainty of the TSDF $\mathcal{A}$, given by $\mathcal{U} - \mathcal{L}$. It is evident in (c)-(e) that $\mathcal{L} \leq \mathcal{A} \leq \mathcal{U}$, and equality holds (bounds are tight), where the uncertainty is zero (pale blue) in (f). Notice that there is some uncertainty even in visible areas.

3.2. Template Matching in 3D

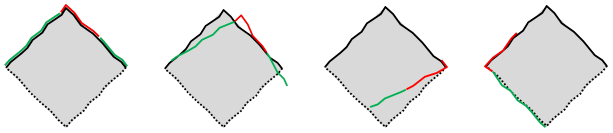
Unfortunately, the values of $\mathcal{A}$ are generally not known for the entire volume. They can be determined exactly in areas that are far enough from occluded areas, but they are not known in occluded areas, where the existence of a shape surface is totally unknown, or even in visible areas that are close to occluded ones. This implies that we cannot compute $Score$ from equation (3). Nevertheless, we show that the TSDF values can be bounded, from above and below, based on the partial depth information. Let us define $\mathcal{U}$ and $\mathcal{L}$ to be the tightest possible upper and lower bounds on the unknown TSDF function $\mathcal{A}$. Figure 2 illustrates the meaning of $\mathcal{S}$, $\mathcal{V}$, $\mathcal{U}$ and $\mathcal{L}$ in flatland and the following claim specifies how the bounds can be computed from the input.

Claim 1. *The TSDF upper and lower bounds are given by:*

$$\mathcal{U} = DT(\mathcal{S} \cdot \mathcal{V}) \quad \textit{and} \quad \mathcal{L} = (-1)^{(1-\mathcal{V})} \cdot DT(\partial\mathcal{V}) \tag{4}$$

Proof. The bounds follow from looking at the limits of the extent of the unknown shape $\mathcal{S}$. On one hand, $\mathcal{S}$ surely *contains* the visible surface $\mathcal{V} \cdot \mathcal{S}$ (and equality is possible) and in this extreme case the TSDF A is simply the distance from $\mathcal{V} \cdot \mathcal{S}$ and the upper bound follows. For the lower bound, similarly, the unknown shape $\mathcal{S}$ is surely *contained* in $\bar{\mathcal{V}} \cup (\mathcal{V} \cdot \mathcal{S})$ (which is the union of the occluded area with the visible surface) and here too - equality is possible. The boundary of the set $\bar{\mathcal{V}} \cup (\mathcal{V} \cdot \mathcal{S})$ is just $\partial\mathcal{V}$ and therefore, in this case, the TSDF A is the signed distance from $\partial\mathcal{V}$. □

Given these bounds, we attempt replacing the full Score (3) with a complementary one. Each point x now has an



(a) Shape (gray), map source (red) | (b) High support (but low consistency) | (c) High consistency (but low support) | (d) High support and high consistency

Figure 3. **Score considerations: the consistency-support tradeoff. (a)** The gray shape is seen from above. Its two top edges (solid black) are visible, while the bottom edges (dotted black) are occluded. A surface red area is mapped to 3 different locations in (**b**), (**c**), and (**d**). The green areas are the rest of the visible surface, which is mapped with the source red area as part of a hypothesis. **(b)** optimizing for support only - results in an inconsistent extension; **(c)** optimizing for consistency only - results in an unreliable extension; **(d)** optimizing for both gives a desirable extension.

interval of values $[\mathcal{L}(x),\mathcal{U}(x)]$, rather than a single value $\mathcal{A}(x)$. For a point x and a transformation T, if we denote $y = T(x)$, then the cost of matching x to y in the original score can be written as:

$$cost(x) = |\mathcal{A}(x) - \mathcal{A}(y)| \tag{5}$$

By definition, $\mathcal{A}(x)$ and $\mathcal{A}(y)$ can take any value in the respective intervals $[\mathcal{L}(x),\mathcal{U}(x)]$ and $[\mathcal{L}(y),\mathcal{U}(y)]$ and therefore, if we define:

$$\Delta_1(x) = \mathcal{L}(x) - \mathcal{U}(y) \quad \text{and} \quad \Delta_2(x) = \mathcal{L}(y) - \mathcal{U}(x) \tag{6}$$

it is easy to verify that:

$$cost_L(x) \leq cost(x) \leq cost_U(x) \tag{7}$$

where:

$$cost_L(x) = \max(0, \Delta_1(x), \Delta_2(x)) \tag{8}$$

$$cost_U(x) = \max(|\Delta_1(x)|, |\Delta_2(x)|) \tag{9}$$

and notice that $cost_L$ and $cost_U$ are tightest possible bounds on $cost$, following from the tightness of $\mathcal{L}$ and $\mathcal{U}$.

Looking at Equation (7), these measures have a clear interpretation. A large $cost_L(x)$ implies a large $cost(x)$ and therefore the point x is surely mapped inconsistently by T (this happens when a fully visible point is mapped incorrectly to a fully visible point). On the other hand, a large $cost_U(x)$ means that the value of $cost(x)$ is largely unknown (this happens, e.g., when a visible point is mapped to a totally occluded point) and in this case x does not provide any information regarding the mapping quality. When summing over $x \in \mathcal{X}$, $cost_L$ quantifies the mapping *inconsistency*, while $cost_U$ quantifies the mapping *support*.

Clearly, one would prefer mappings with low inconsistency and high support, but there is an inherent tradeoff between the two. On one hand, insisting on minimal inconsistency will favor mappings that map mostly into unknown areas and these have very low support (few points that actually prove the map consistency) and can not be reliable enough for producing hypotheses. On the other hand, insisting on maximal support could come at the cost of imperfect consistency and might limit the potential matches to fully visible areas, but these would not be likely to extend into the unknown regions, which we wish to complete. Figure 3 illustrates the tension between consistency and support.

We therefore define the score for T as a linear combination of two scores:

$$Score(T) = \alpha \cdot Score_L + (1-\alpha) \cdot Score_U \tag{10}$$

where:

$$Score_L = \int_X \left(1 - e^{-\frac{cost_L^2(x)}{2\sigma_L^2}}\right) dx \tag{11}$$

$$Score_U = \int_X \left(1 - e^{-\frac{cost_U^2(x)}{2\sigma_U^2}}\right) dx \tag{12}$$

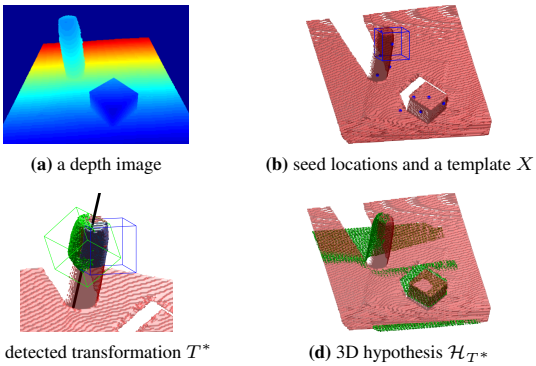
Note that σ_L and σ_U control the degradation rates of each of the scores and α controls the tradeoff between them.

3.3. Generating Completion Hypotheses

We are now ready to describe the entire process that leads to the generation of completion hypotheses, which are the input to our optimization. The starting point is a seed location on the visible surface, around which we take an axis-aligned sub-volume X and search for a transformation T^* that minimizes $Score(T)$.

Using the Fast-Match algorithm [10], the minimization consists of efficiently sampling the combined space (6 degrees of freedom) of all 3D rotations, translations and reflections, evaluating each transformation and returning the best one found. A couple of comments are in order here. First, the sampling density of the transformation space is inversely proportional to the total variation (smoothness) of X (see [10]) and these volumes are rather smooth due to the TSDF representation. Second, as is done in [10] we follow a branch-and-bound scheme, where the transformation space is first sampled sparsely and then, regions with high scores are discarded and a denser sampling is performed in the remaining regions.

We now use each high quality mapping T^* in order to 'copy' the visible surface around the source location into the occluded areas. At this stage we discard the original sub-volume X, that was used for finding the local similarity and instead we choose a (visible) surface area X_{T^*} that will be mapped to form a hypothesis. To do so, we apply the transformation T^* on the entire volume and take the largest possible region, around the source location, that adheres to the transformation T^*. This region typically includes most of the surface region from within the original sub-volume (but not necessarily all), as well as surface areas from outside the original volume X. More specifically, we perform a *hysteresis* process to determine the exact region, where the intuition is to take areas that are not known to be inconsistent (low $cost_L(x)$), and which are not far (geometrically)



(a) a depth image **(b)** seed locations and a template X

(c) detected transformation T^* **(d)** 3D hypothesis $\mathcal{H}_{T^*}$

Figure 4. **Stages of hypothesis generation (a)** An example depth map. **(b)** The visible surface (red), with the automatically detected seed locations and an example template X around one of them. **(c)** A detected transformation T^*, mapping the sub-volume X (blue) to $T^*(X)$ (green). **(d)** The resulting 3D hypothesis $\mathcal{H}_{T^*}$ (green).

from areas that are known to be consistent (low $cost_U(x)$). Formally, we define:

$$X_{T^*} = [DT(cost_U(x) < t_U) < \epsilon] \cdot [cost_L(x) < t_L] \cdot (V \cdot \mathcal{S}) \tag{13}$$

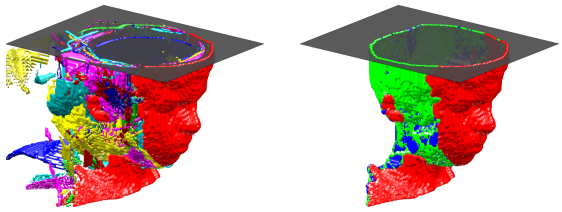
where $[\cdot]$ is 1 if the condition inside the square brackets is true and 0 otherwise. The constants t_U, t_L are score thresholds and ϵ is a (Euclidean) distance threshold. Note that the multiplications are between indicator matrices (the third one being the visible surface) and therefore stand for intersections between the relevant sets. Finally, the resulting 3D hypothesis, denoted by $\mathcal{H}_{T^*}$ is defined by

$$\mathcal{H}_{T^*} = T^*(X_{T^*}) \tag{14}$$

Figure 4 summarizes the entire hypothesis generation process. Given a single depth map (a), we use seed locations on the visible surface and define a small sub-volume around each of them (b). We run our 3D template matcher to detect a potential candidate (c) and use the transformation between the template and target to map a larger region of the scene, which serves as our 3D hypothesis (d) the final input to the optimization.

3.4. Optimization in 3D

Our goal now is to merge all hypotheses together with the original surface evidence in a consistent manner. In a similar fashion to what is done in 2D image completion (see e.g. [4]), this can be posed as a discrete optimization problem on a 3D raster where each voxel is assigned a label and labels denote different hypotheses. This optimization is challenging because the domain (volume) and the label space (number of hypothesis) are very large. Moreover, unlike the case of image completion, neighboring voxels with labels originating from different hypotheses may cause inconsistencies in the solution, in the form of incomplete surfaces or surfaces with undesirable topology.



(a) input and hypotheses **(b)** input and solution

Figure 5. **From multiple hypotheses to a coherent solution.** In this example, the input is a frontal scan of a child (bright red surface) captured by a Kinect sensor. **(a)** The input scan and 51 generated hypotheses. **(b)** The input scan and our completion, which is encouraged to coincide with hypotheses (green areas), but can deviate from them (blue areas) in order to create a smooth completion. Looking from above (shown sliced) - the contours of the hypotheses can be seen to contain noise and outliers. The solution (blue and green contour) manages to create a consistent boundary.

Instead, we formulate a binary optimization problem on the 3D raster, where each voxel x is classified to be either in the interior ($L(x) = -1$) or the exterior ($L(x) = 1$) of the scene. The solution is driven to be coherent with the hypotheses by a carefully designed energy term. This approach enables modeling scenes with more complex surface topologies and has the advantage that the resulting solution allows for a clear interpretation of the scene surface. Refer to Figure 5 for an example result of applying our optimization method on the input (which is the visible surface and 3D hypotheses) producing a valid completion. As can be seen, we obtain from the previous stage a large number of hypotheses, which are typically inconsistent with each other and do not completely 'cover' the unseen surface area.

The solution we propose searches for a surface that 'interpolates' between the visible surface areas. Following the work of Lempitsky and Boykov [11] we derive a binary MRF, which is minimized using graph-cuts [1]. Our energy term, however, is more related to the TV-$L1$ energy of Zach *et al.* [20], even though they optimize for a complete field, rather than for a binary partition of the space. It is given by:

$$E = \sum_{\mathbf{x}} E_D(L(\mathbf{x})) + \lambda \sum_{(\mathbf{x},\mathbf{x}')} E_S(L(\mathbf{x}), L(\mathbf{x}')) \tag{15}$$

where the summations are over all voxels $\mathbf{x}$ in the volume and all pairs $(\mathbf{x}, \mathbf{x}')$ of neighboring voxels. The data fidelity term E_D, that is defined at each voxel $\mathbf{x}$ by

$$E_D(L(\mathbf{x})) = L(\mathbf{x}) \cdot \sum_{\mathcal{H}} DT^K_{\mathcal{H}}(\mathbf{x}) \tag{16}$$

measures the average agreement of the labeling $L(\mathbf{x})$, with the set of hypotheses $\mathcal{H}$. This average is weighted by the truncated signed distance transform $DT^K_{\mathcal{H}}(\mathbf{x})$, which measures the distance of the voxel $\mathbf{x}$ from the hypothesis $\mathcal{H}$, truncated to the interval $[-K, K]$ for robustness. $DT^K_{\mathcal{H}}$ is positive on the inner side of the hypothesized surface and negative outside.

Algorithm 1 ***Depth Extension***

Input: the visible-surface $\mathcal{S} \cdot V$ (in raster representation)
Output: the full scene $\mathcal{S}$ (in raster representation)

Volume Representation (see Secs. 3.1 and 3.2)

1. Compute visible-volume $\mathcal{V}$ and visibility-boundary $\partial\mathcal{V}$
2. Compute the TSDF bounds $\mathcal{L}$ and $\mathcal{U}$ (as in Claim 1)

Hypothesis Generation (see Sec. 3.3)

1. Select a set $\mathcal{X}$ of interest sub-volumes (seed locations)
2. For each sub-volume $X \in \mathcal{X}$:
   (a) Run the $3D$ variant of Fast-Match, using the $\mathcal{L}$ and $\mathcal{U}$ volume representation, to find matching sub-volumes $\{T_i(X)\}$, under transformations $\{T_i\}$, whose matching scores $\{Score(T_i)\}$ are below a threshold t.
3. For each $T \in \mathcal{T}$ ($\mathcal{T}$ is the set of discovered mappings):
   (a) Compute the source area X_T and the $3D$ hypothesis $H_T = T(X_T)$

Scene Reconstruction Optimization (see Sec. 3.4)

1. Construct the energy terms E_D and E_S as in Eq. (15)
2. Solve for the scene $\mathcal{S}$ using Graph-Cuts [1]

The pairwise smoothness term E_S, which is a location-dependent Potts model, is given by:

$$E_S(L(\mathbf{x}), L(\mathbf{x}')) = W_{\mathbf{x},\mathbf{x}'} \cdot [L(\mathbf{x}) \neq L(\mathbf{x}')] \qquad (17)$$

where $[\cdot]$ is the indicator function. This term is a Total-Variation regularizer, that measures the area of the boundary. As can be seen in Figure 5(b), the surface (boundary) in our solution is divided into three kinds. Red surface regions are the input visible surface; green regions are ones that coincide with some completion hypothesis; and blue ones are the rest. The location-dependent $W_{\mathbf{x},\mathbf{x}'}$ takes three different values, depending on which of the three types does the boundary edge $(\mathbf{x}, \mathbf{x}')$ belong to, allowing to weight the boundary areas of each type differently. Passing through the input surface is obligatory and therefore $W = 0$ over $\mathcal{S} \cdot V$. Regarding non-input voxels - passing through hypothesis voxels is preferable and therefore we set $W = 1$ compared to $W = 2$ in non-hypothesis locations. Algorithm 1 summarizes the main steps of our method.

4. Results

Implementation Details The input partial visible surface ($\mathcal{S} \cdot V$) is represented by a 256^3 voxel grid (raster). For each scene we use a fixed radius r (i.e. half the dimension) of the axis-parallel cubes that form the source search sub-volumes. It is chosen manually according to the general scale of the scene. The seed voxels are taken as r-separated uniform cover of the visible surface. Each seed point is potentially discarded if the surface voxels in its vicinity are too few or if they approximately lie on a plane (determined by a coordinate eigen-decomposition) - cases in which the respective subvolume is not sufficiently discriminative.

In the template matching stage, we run a 3D version of Fast-Match [10], where we collect the 3 best possible mappings per seed location and discard those with $Score > t$, for $t = 0.035$. The parameters in *Score* (10) are fixed throughout our experiments: $\alpha = 0.5$, $\sigma_{\mathcal{U}} = 3$ and $\sigma_{\mathcal{L}} = 1$. Regarding the computation of the source area X_T that adheres to the transformation T (Equation (13)), we take hysteresis thresholds t_U and t_L to be the 40^{th} and 70^{th} percentiles, respectively, of the original sub-volume $cost_U$ and $cost_L$ distributions. The hysteresis parameter ϵ was taken to be twice the radius r of the original sub-volume.

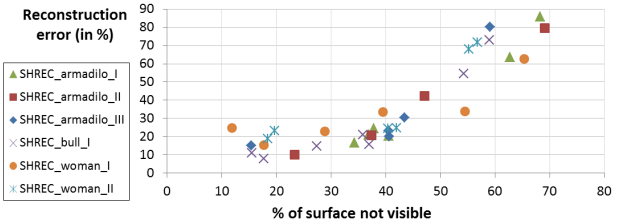


Figure 6. **Statistics on SHREC [6] shape completions**: We generated 6 completion instances for each of the 6 SHREC shapes. **x-axis**: the percent of shape surface that is occluded, representing the instance difficulty. **y-axis**: *reconstruction error*. Up to an occlusion rate of ∼50% the algorithm performs well (see Figure 7 for visual comparisons with the original shapes). The performance deteriorates at higher occlusion levels (see Figure 8 for such cases).

4.1. Reconstruction of 3D shapes

In this experiment, we create controlled surface completion tasks by removing surface parts from 3D modeled shapes and then attempting to reconstruct the entire surface. Unlike the more realistic scenario of completion from partial scans, this scenario lets us compare our results to the original shape both quantitatively and qualitatively.

The data For this experiment we use the SHREC07 dataset [6] which consists of a variety of closed triangulated meshes of CAD models. We chose in particular six shapes, which are especially challenging, since they include several models in a variety of different poses, with complex surfaces. This is to emphasize that our method works without knowledge of the shape class, surface primitives or global symmetry assumptions. We use shapes 288, 291, 283, 14, 8 and 386 which we term 'armadillo 1', 'armadillo 2', 'armadillo 3', 'woman 1','woman 2' and 'bull 1' respectively.

We then randomly generated multiple completion tasks (instances) of varying levels of difficulty for each of the six shapes. In each instance, the partial surface is generated in one of three ways: 'single view', where we keep only sur-

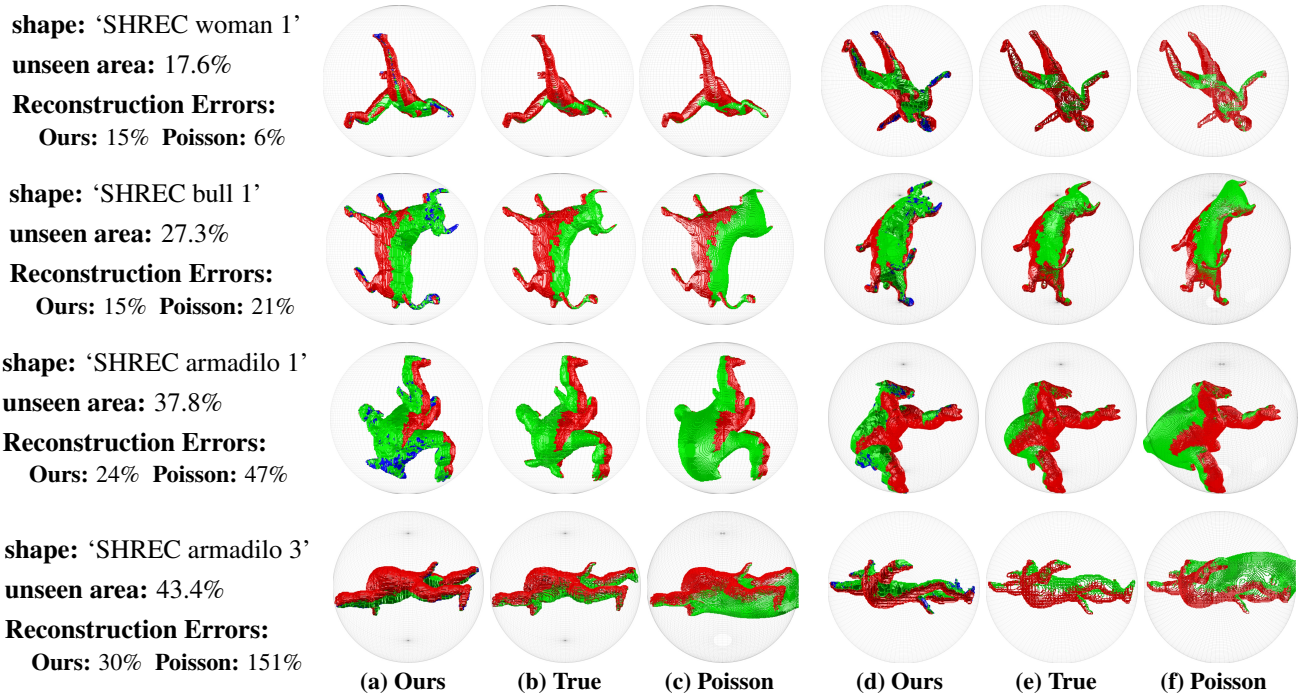


Figure 7. **Shape completion examples** (with unseen area $< 50\%$): Each row shows a completion instance generated from the SHREC dataset. These are 4 out of the 36 instances (additional examples are provided in Supplementary Materials) that we generated randomly (see text), resulting in a partial view of the shape (shown in red). In each example, reconstructions are shown from 2 different viewpoints: (**a**-**c**) and (**d**-**f**). Our surface completion is shown in **(a)** and **(d)**, where completed areas are colored in green if they originate from a completion hypothesis or blue otherwise. For reference, the true completion is shown in green from the same views in **(b)** and **(d)** and the Poisson Reconstruction in **(c)** and **(f)**. In addition, we report reconstruction errors on the left. Note: **details are best seen when viewed in zoom**.

face areas visible from a single randomly chosen viewing direction (at a fixed distance). This option creates the hardest instances where typically over 40% of the shape surface is unseen; 'two-view-orthogonal', where we keep only surface areas visible from either one of a random pair of viewpoints, which are 90° apart. Here, typically $20\% - 50\%$ of the surface is unseen; 'two-view-opposite', where we use random opposite viewpoints, with unseen area in the range $10\% - 30\%$. In all cases, the shape surface is rastered in an orthogonal 256^3 volume centered at the shape's center of gravity. Overall, we created 6 instances for each shape (2 of each option) resulting in a total of 36 instances. The distribution of their degree of difficulty (unseen surface area) can be seen by looking at the $x-$axis of the chart in Figure 6.

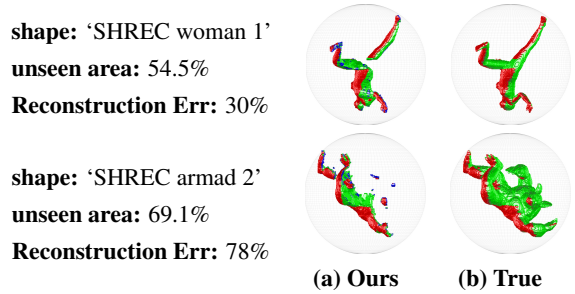
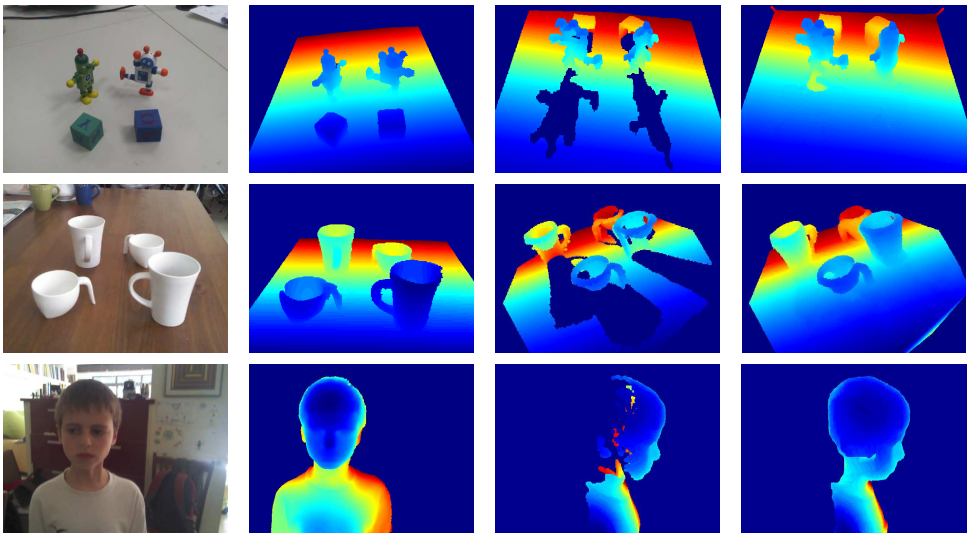


Figure 8. **Shape completion hard examples** (unseen area $> 50\%$): See Figure 7 for explanations and text for interpretation.

Results The completed shapes are evaluated by the *reconstruction* error, which is the volume of the symmetric difference between original and reconstructed shapes, as a $\%$ of the original shape volume. In the Supplemental Materials we consider a related measure - the *area prediction* error for which we draw similar conclusions. The chart in Figure 6 shows some statistics of this experiment, showing that the algorithm performs well when the occluded surface area is up to $\sim 50\%$ of the entire surface and beyond that the performance degrades rather rapidly. Figure 7 shows four examples, each from 2 different viewpoints, of our reconstructions in which the unseen area is below 50%. In these examples (ordered by increasing difficulty) our completed surfaces fit nicely to the input (red) surface and suggest overall plausible completions. Notice that most of the completion is based directly on the generated hypotheses (green surface areas), while the remaining areas (blue) are obtained due to the total-variation regularization in our 3D optimization. In the Supplementary Materials, we provide visualizations for many other instances of this experiment.

Lacking a suitable alternative method for completing large, out-of-viewpoint holes, we compare to Poisson Reconstruction [7, 8] as a baseline. Poisson Reconstruction is known to produce high quality surfaces in visible parts of a scene. However, as the unseen regions get larger it



(a) RGB (not used) **(b)** depth map **(c)** New view **(d)** Our completion

Figure 9. **New viewpoint depth synthesis.** Results are shown for the '*Robots*', '*Cups*', and '*Child*' data-sets. **(a)** RGB image, shown for reference. **(b)** The input - a single-view depth-map. **(c)** A new viewpoint depth-map, computed from the input depth-map (b). **(d)** Our completed depth-map. See Fig. 1 for the '*Spray*' example and the text for details.

interpolates extremely smooth continuations of the occlusion boundaries which typically do not resemble the original shape, as can be seen in Figure 7. To provide it with ideal conditions, we calculate normals on the full shape and transfer them to the partial shape, to avoid artifacts around occlusion boundaries and areas with low point density.

Figure 8 shows some limitations of our method, through 2 cases where the unseen area is over 50%. The woman example was reconstructed as two 3D objects (the left leg is separated from the body), due to the lack of evidence in the existing view for the connection between the parts. In a similar fashion, the Armadillo reconstruction example (with only 30% visible surface) preferred to generate a geometry where the surface area is minimal, as long as it can be explained by existing hypotheses. The addition of prior knowledge of body shape, or the usage of additional assumptions (such as the need of the recovered geometry to be stable relative to gravity) could possibly be incorporated to improve reconstructions in such cases, which are extreme for methods that do not assume such prior knowledge.

4.2. Synthesizing new viewpoint depth-maps

In this experiment we complete an entire scene, under severe occlusion, given a *single* depth image. Many methods deal with filling holes in depth maps that are due to acquisition faults or due to slight viewpoint changes. In contrast, our method is capable of completing large unseen surface regions and this is demonstrated here through the application of novel viewpoint depth-map synthesis.

The data For this set of experiments, we collected data using a Kinect sensor. Since our focus is on the task of completing surface areas that are *not* visible from the single viewpoint, we wish to avoid dealing with the typical missing values or noisy ones, especially around depth discontinuities. We therefore collected our data examples using the Kinect Fusion [13] system, making slight viewpoint changes during the scan. We then projected the output point-cloud to a single viewpoint, resulting in a *single* depth-map, which is the only input to our algorithm.

Results We created depth maps of 4 scenes ('*Spray*', '*Robots*', '*Cups*' and '*Child*'), as described above. Our new-view depth synthesis results can be seen in Figures 1 and 9. The first two examples in Figure 9 are extremely challenging as we generate new views that are 180° from the input view point. Despite the drastic view change, the algorithm fills the holes nicely and reasons about the depth relationships between the robots, the cubes and the table surface. Similarly, the algorithm performs well on the '*Cups*' example, whose input depth-map has many artifacts. The last row of Figure 9 shows the case of generating a view point of a head that is orthogonal to the original viewing angle, for which the algorithm produces a plausible solution.

5. Conclusions

We proposed an algorithm for depth extension from a single depth image. The algorithm detects repetitive 3D structures and uses them to generate a set of hypotheses, which are merged in a coherent manner using 3D discrete optimization. The method was shown to be able to complete a variety of scenes and we believe it could be extended to solve more complex task-specific challenges, by incorporating shape priors or physical based assumptions.

Such a capability can enable applications that expect full geometry, such as robot path planning or simulation of lighting and audio, that have access to a scene from a limited range of view points. The 'peeking' template matching, proposed in this work, offers a new formulation of template matching in the presence of uncertainty.

Acknowledgments: This research was supported in part by the Israeli Science Foundation and the Ministry of Science.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11), 2001. 5, 6

[2] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of ACM SIGGRAPH*, 1996. 3

[3] R. Guo and D. Hoiem. Support surface prediction in indoor scenes. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 2

[4] K. He and J. Sun. Statistics of patch offsets for image completion. In *Proc. European Conference of Computer Vision (ECCV)*, 2012. 2, 5

[5] C. Hernández, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 2007. 3

[6] S. Jayanti, Y. Kalyanaraman, N. Iyer, and K. Ramani. Developing an engineering shape benchmark for cad models. *Computer-Aided Design*, 38(9), 2006. 6

[7] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Eurographics symposium on Geometry processing (SGP)*, volume 7, 2006. 2, 7

[8] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 2013. 2, 7

[9] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. J. Guibas. Acquiring 3d indoor environments with variability and repetition. *ACM Transactions on Graphics (TOG)*, 31(6), 2012. 2

[10] S. Korman, D. Reichman, G. Tsur, and S. Avidan. Fast-match: Fast affine template matching. In *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 2013. 3, 4, 6

[11] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 2007. 5

[12] S. Natan, S. Lior, G. Ran, and K. Pushmeet. A Contour Completion Model for Augmenting Surface Reconstructions. *Proc. European Conference of Computer Vision (ECCV)*, 2014. 2

[13] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE symposium on Mixed and augmented reality (ISMAR)*, 2011. 3, 8

[14] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing (SGP)*, 2005. 2

[15] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz. Occluding contours for multi-view stereo. In *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014. 2

[16] A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. *ACM Transactions on Graphics (TOG)*, 23(3), 2004. 2

[17] J. Shen and S.-C. S. Cheung. Layer depth denoising and completion for structured-light rgb-d cameras. In *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[18] L. A. Torres-Méndez and G. Dudek. Reconstruction of 3d models from intensity images and partial depth. In *Conference on Artificial Intelligence (AAAI)*, 2004. 2

[19] L. Wang, H. Jin, R. Yang, and M. Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[20] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust TV-L1 range image integration. In *IEEE International Conference on Computer Vision (ICCV)*, 2007. 5

[21] B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S. Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 2013. 2