Im2Depth: Scalable Exemplar Based Depth Transfer

Mohammad Haris Baig†, Vignesh Jagadeesh∗, Robinson Piramuthu∗,
Anurag Bhardwaj∗, Wei Di∗, Neel Sundaresan∗
† eBay Research Labs, San Jose, CA-95125, ∗Dartmouth College, Hanover, NH-03755
haris@cs.dartmouth.edu, vjagadeesh, rpiramuthu, anbhardwaj, wendi, nsundaresan@eBay.com

Abstract

The rapid increase in number of high quality mobile cameras have opened up an array of new problems in mobile vision. Mobile cameras are predominantly monocular and are devoid of any sense of depth, making them heavily reliant on 2D image processing. Understanding 3D structure of scenes being imaged can greatly improve the performance of existing vision/graphics techniques. In this regard, recent availability of large scale RGB-D datasets beg for more effective data driven strategies to leverage the scale of data. We propose a depth recovery mechanism “im2depth”, that is lightweight enough to run on mobile platforms, while leveraging the large scale nature of modern RGB-D datasets. Our key observation is to form a basis (dictionary) over the RGB and depth spaces, and represent depth maps by a sparse linear combination of weights over dictionary elements. Subsequently, a prediction function is estimated between weight vectors in RGB to depth space to recover depth maps from query images. A final superpixel post processor aligns depth maps with occlusion boundaries, creating physically plausible results. We conclude with thorough experimentation with four state of the art depth recovery algorithms, and observe an improvement of over 6.5 percent in shape recovery, and over 10cm reduction in average L1 error.

1. Introduction

Inferring 3D structure from a single 2D image is a challenging problem receiving increased attention due to its wide applicability in augmented reality and scene understanding. Humans have the remarkable ability to infer scene structure with one eye closed. This is attributed to knowledge acquired over time on the perceptual organization of objects/scenes, as reported by extensive research in psychology [2, 11]. The capability of machines to replicate this effect would open new avenues and enhance capability of existing computer vision systems. For instance, depth estimation would enable real time measurements/photogrammetry from 2D images without explicit calibration. In contrast, a carefully calibrated camera setup or acquisition of multiple image snapshots for estimating 3D structure would require users to possess a certain level of expertise. With the ever increasing use of smartphones, the capability of monocular mobile cameras to perceive depth would have significant commercial implications.

In the field of object recognition and scene understanding, side information on depth has been shown to offer a significant performance and reliability boost to existing systems. A contemporary example is the reliable identification of body parts from depth images provided by the Kinect sensor [15], and recognition of gestures from RGB-D video feeds. Performance of pose estimators and gesture recognizers utilizing RGB channels is far below the performance obtained by including the depth modality. While the number of RGB-D images being acquired is constantly on the increase, acquisition of RGB images is progressing faster by several orders of magnitude. This leads to an obvious motivating question of whether it would be possible to reliably transfer depth from a large database of RGB-D images to a single query RGB image. Inspite of the promise offered by monocular depth estimation, there have been serious bot-
I. Introduction

Manifold on RGB descriptors Manifold on Depth descriptors

1.1. Motivation

• Identification of reliable visual descriptors for depth transfer, and structural post processing for aligning estimated depth maps with image boundaries

2. Related Work

Understanding 3D scene structure has a rich history in computer vision. Traditional techniques have predominantly worked with multiple images to make the problem of 3D recovery well posed. These techniques capture images of the same object from multiple views, leading to N-view reconstruction systems [5, 6]. Another technique for making the problem tractable is in utilizing frames that capture different viewpoints of an object over time, leading to successful structure from motion and SLAM systems. Further, techniques such as shape from shading [1, 21] are attractive alternatives for reconstructing 3D structure, but are unreliable due to strong modeling assumptions, as pointed out in [14]. Specifically [1, 21] have not been shown to work on arbitrary scenes in addition to large compute time and parametric prior assumptions on natural scenes. Active research in the area of single image depth transfer is more recent, and is usually motivated by the type of scene considered (indoor/outdoor), or the type of prediction desired (absolute/relative). We now summarize the key papers in single image depth estimation.

Automatic Photo Pop Up (APP): Hoiem et al. [7, 8] utilize the notion of geometric context to reason on scene layout, dividing the scene into top/left/center/right/bottom regions, thus gathering strong cues for labeling scene depth. Their primary focus was on outdoor scenes where the sky, ground, buildings are commonly recurring concepts that correspond to top/bottom/side layout labels.

Make3D: Saxena et al. [14] proposed the Make3D system, which casts the problem of monocular depth estimation as a supervised learning problem. A Markov Random Field is learnt for mapping between RGB and depth space. The random field model also utilizes priors on the 3D properties of surfaces to refine solutions that may not adhere to prior knowledge on how natural scenes are organized.

Non-Parametric Sampling (NPS): Karsch et al. [9] approach the problem of depth estimation using a framework that retrieves visually similar images from a training dataset and utilizes retrieved depth maps for final depth estimation. Their retrieval is performed using the GIST descriptor, followed by SIFT flow warping of retrieved neighbors. Finally, a depth optimization step comprising of spatial smoothness and global priors is utilized to recover depth from the retrieved images.

BU/Google (BG): This is the most recent work [10] in data driven depth transfer. On comparison with NPS, it was shown to achieve higher performance on indoor scenes and lower performance on outdoor scenes. The crux of their method is similar to that of NPS in that they do retrieve visu-
3. Proposed Formulation

Notations and Preliminaries: Let us assume that we are given a set of $L$ RGB images and their corresponding depth maps denoted by $I_{\text{train}} = \{R_i \in \{0, 1, \ldots, 255\}^{M \times N \times 3}, D_i \in [0, 1, \ldots, 10]^{M \times N}\}^{L}_{i=1}$ along with their respective global image descriptors $\{r_i \in \mathbb{R}^{r_c}, d_i \in \mathbb{R}^{r_d}\}^{L}_{i=1}$. The goal of depth transfer algorithms is to estimate a mapping that can generate a depth map for an incoming RGB query image $R_q \mapsto \hat{D}_q$. The estimated depth map $\hat{D}_q$ is compared with the ground truth depth to quantify the quality of depth transfer, such as distance between prediction and ground truth $||D_q - \hat{D}_q||$.

We now describe the proposed formulation for depth transfer. Let us assume we have dictionaries, $W_r \in \mathbb{R}^{r_c \times p_r}$ and $W_d \in \mathbb{R}^{r_d \times p_d}$, comprising $p_r$ and $p_d$ elements representative of RGB and depth images respectively. The workflow can be summarized as in Figure 3. The training RGB and depth images are first transformed into a space spanned by the dictionary elements $W_r$ and $W_d$ respectively. Subsequently, a cross-domain mapping is estimated between the descriptors of RGB and depth images in the space spanned by their respective dictionaries. This learnt mapping is utilized in estimating a depth map from a query RGB image to generate depth predictions.

3.1. Sparse Positive/Pairwise Distance Descriptors

A basic ingredient of our approach is the transformation of features to a data dependent space, spanned by dictionary elements. In other words, recall that the $i^{th}$ image is represented using a set of global descriptors $\{r_i, d_i\}$. We now wish to transform $\{r_i, d_i\}$ to a data dependent space denoted as $\{\gamma_i, \delta_i\}$. The data dependent transformation is achieved by: $\gamma_i = g_r(W_r, r_i)$, $\delta_i = g_d(W_d, d_i)$. In other words, the global descriptors are represented in terms of their respective dictionary elements. The functional forms of $g_r : \mathbb{R}^{r_c \times p_r} \times \mathbb{R}^{r_c} \rightarrow \mathbb{R}^{p_r}$ and $g_d : \mathbb{R}^{r_d \times p_d} \times \mathbb{R}^{r_d} \rightarrow \mathbb{R}^{p_d}$ determine the type of relationships encoded in dictionary elements. The effectiveness of utilizing transformed descriptors is shown by the clustering behavior in Figure 2.
PD descriptor and Depth Dictionary PD descriptor respectively, where $K(x, y) = ||x - y||_2$.

**RGB Dictionary PD Descriptors**

$$\gamma^k = [K(r_i, W_r(:, 1)) ... K(r_i, W_r(:, p_r))]$$

**Depth Dictionary PD Descriptors**

$$\delta^k = [K(d_i, W_d(:, 1)) ... K(d_i, W_d(:, p_d))]$$

### 3.1.2 Sparse Positive Descriptors

Our second strategy is to perform a sparse decomposition to predict weights on basis elements using a sparse set of coefficients (at most $\Theta$ nonzero entries) over the basis elements. We utilize algorithms published in [12] for estimating weights on dictionary elements.

**RGB Dictionary Sparse Positive**

$$\min_{\gamma^i \in \Re^p} \|r_i - W_r \gamma^i\|_2 \text{ s.t. } \|\gamma^i\|_0 \leq \Theta, \gamma^i \geq 0$$

**Depth Dictionary Sparse Positive**

$$\min_{\delta^i \in \Re^p} \|d_i - W_d \delta^i\|_2 \text{ s.t. } \|\delta^i\|_0 \leq \Theta, \delta^i \geq 0$$

### 3.2. Creating the Dictionary

The problem of creating a dictionary that is representative of the entire dataset is an interesting problem in its own right. Since the focus of this work is on depth transfer, we utilize straightforward techniques for dictionary formation while observing that there is a possibility of creating better dictionaries than used here. The first technique we utilize for dictionary creation is k-means clustering. Descriptors from the entire dataset are clustered into a set of representative $p$ centroids. The images closest to these cluster centroids are then picked as basis elements of the dictionary. Since clustering selects diverse representatives of the dataset, the clusters formed are most likely to represent different parts of the feature space where input data points exist. The alternative approach for creating the dictionary is to utilize the entire training dataset as a dictionary. This is in line with sparse coding approaches used for face recognition [20]. This leads to a much larger dictionary and higher dimensionality of projected features, that can still be handled by sparse matrix decomposition techniques.

### 3.3. Cross Domain Mapping

The crucial step after the dictionary dependent descriptors are computed is the estimation of a transformation between RGB features and depth features. While there are a variety of techniques one could leverage for establishing this transformation, we model this transformation as a linear model given by,

$$\Gamma = [\gamma^k \ \gamma^2 \ ... \ \gamma^l]$$

$$\Delta = [\delta^k \ \delta^2 \ ... \ \delta^l]$$

$$\Delta = TT \rightarrow T = \Delta \Gamma^{-1}$$

The transformation matrix $T \in \Re^{pd \times pr}$ defines a mapping from the RGB feature space to the depth feature space assuming a linear model stated above.

### 3.4. Structural Post Processing

It is useful to recall that the visual descriptors utilized for estimating the PD/sparse descriptors are global image descriptors that attempt to describe the entire image. However, we are aiming to predict pixel level depth information during test time. In other words, pixel level information pertaining to local shape/texture in RGB images was never considered till this point. As a result the depth maps predicted from the framework are coarse, and do not latch on to the true image borders. In order to refine the predicted depth map, we propose utilizing the superpixel partitioning [4] of the input RGB image $R_i$, denoted by $S_i = \{s_{i1},...s_{i|S_i|}\}$, where $|.|$ denotes set cardinality and $s_{ij}$ denotes the $k$th pixel in superpixel $s_{ij}$. Recall that the predicted depth map is denoted by $D_i$. We refine the predicted depth map under the assumption that the pixels constituting a superpixel are more likely to have similar depth values. The refinement is performed using $D_i[s_{ij}] = \sum_k D_i[r_{ij}] / |D_i[r_{ij}]|$. The above procedure fits a piecewise constant value to the entire superpixel resulting in a depth map that is more interpretable since it aligns well to image edges, see Figure 4. Subsequently, the superpixels are deformed to a ramp by utilizing RANSAC to fit planes. The overall workflow of the proposed technique is summarized in Algorithm 1.

### 4. Experimental Results

We now turn our attention to the experimental validation of the proposed depth transfer mechanism, and explain the dataset utilized, metrics employed, and comparisons with state of the art. Qualitative results of the proposed technique, along with state of the art is illustrated in Figure 4.

**Dataset:** We report experimental results on the labeled NYU-V2 dataset, made up of 1449 RGB-D images of size $480 \times 640$. The dataset comprises 27 scene categories, with a total of 464 unique scenes. Images within every unique scene are captured from different viewpoints. We used a subset of this dataset for which the authors have provided post-processing (in-painting for hole-filling) on depth-maps.

**Evaluation Metrics:** As mentioned previously, the task at hand dictates the type of depth maps to be recovered, i.e. absolute depths or relative depths. Further, there does not
Figure 4. **Qualitative Comparison:** of the proposed technique (Proposed) with plane fitted post processing (Proposed/Planes) with state of the art methods (BU, NPS, Global Priors, Make3D). Observe the similarity in depth magnitude and gradient between the ground truth and proposed method, in contrast to all other competing techniques.

seem to be a consensus on metrics to be utilized for judging quality of depth recovery. In order to avoid clutter in results, and to present our findings on the proposed method concisely, we resort to using the Normalized Cross Correlation (NCC) score for quantifying relative depth recovery and the L1 error for absolute depth recovery. When we compare the proposed techniques with state of the art algorithms, we utilize three additional metrics, namely Root Mean Square Error (RMSE) [14], Relative Error [9] and Log10 Error [9] to remain consistent with metrics used by the original authors. We now explain the NCC and L1 metrics briefly, while referring the reader to cited papers for the remaining metrics. The normalized cross correlation (NCC) metric measures the extent of shape recovery, where the prediction of relative depths is more important than the prediction of absolute depths. In other words, NCC would indicate a high score if relative depth values are preserved. Denoting by \( \mu_x, \sigma_x \) the mean and standard deviation of the vector \( x \), the NCC metric is defined as, 

\[ NCC = \frac{(D_1 - \mu_{D_1})^T (D_2 - \mu_{D_2})}{\sigma_{D_1} \sigma_{D_2} |D_1|} \]

The L1 error measures the absolute variation between the ground truth depth and predicted depth. The metric is defined as, 

\[ L1 = ||D_1 - D_2||_1 \].

We now analyze the proposed system, along with design decisions made. To this end, we first report results on the performance of several global descriptors used to quantize the RGB space. Subsequently, we quantify performance of dictionaries learnt using three different approaches, the effect of dictionary size, and possible means of compressing the dictionary.

**Train/Test Splits:** For validating our techniques, we experiment with multiple train/test splits. This assumes significance since the NYU-V2 dataset is a video dataset from which a subset of images are sampled and labeled to cre-
Algorithm 1: im2depth: Exemplar Based Depth Transfer

Require: \( I_{\text{train}} = \{R_i, D_i, r_i, d_i\}, 1 \leq i \leq L, I_{\text{query}} = \{R_q, r_q\} \)

**TRAINING PHASE**

\( W_F = \text{Create Dictionary(} r_{1 \leq i \leq L} \)\)

\( W_d = \text{Create Dictionary(} d_{1 \leq i \leq L} \)\)

for \( i = 1 : L \) do

- Estimate Sparse Positive Depth Descriptor \( \delta_i \) using Eq 4
- Estimate RGB PD Descriptor \( \gamma_i \) using Eq 1

end for

\( \Gamma = [\gamma_1^k \ \gamma_2^k \ \ldots \ \gamma_L^k] \)

\( \Delta = [\delta_1^s \ \delta_2^s \ \ldots \ \delta_L^s] \)

\( \Delta = T \Gamma \rightarrow T = \Delta \Gamma^{-1} \)

**TEST PHASE**

Estimate RGB PD Descriptor \( \gamma_q^k \) using Eq 1

Estimate Depth Weights \( \delta_q = T \gamma_q^k \)

Recover query depth map \( D_q = W_d \delta_q \)

PostProcess \( D_q \) and fit planes using RANSAC

---

Figure 5. **Quantitative Comparison on Global Descriptors:** Performance comparison of various global descriptors on depth recovery task using L1 error metric and NCC. The performance of techniques were recorded using the three different splits of the training/test set as discussed in the experimental section. This experiment points to the impressive performance of classemes on both metrics across all three splits of the data.

-----

Figure 6. **Quantitative Comparison on Dictionary Size:** The size of dictionary has a major impact in the running time of algorithms. The above plots illustrate three experimental settings for transforming from the RGB weight space \( \gamma_q^k \) to depth weight space \( \delta_q^s \). It is clear that transforming between two 50 dimensional spaces (blue bars) loses considerable information, while transformation between two 1200 dimensional spaces preserves maximum information (green bars). However, transformation between a 1200 dimensional \( \gamma_q^k \) to a 50 dimensional \( \delta_q^s \) matches up to the performance of transforming between two 1200 dimensional spaces (green bars), resulting in huge computational savings.

**Varying Constraints on Weights:** We experimented with variations in the projection step for estimating depth weights \( \delta_r \), whereby the weights for linear combination are estimated through three different methods, Least-Squares (LSQ), Quadratic Programming (QP) for Positive dictionaries, Orthogonal Matching Pursuit (OMP) with positivity and sparsity constraints. This experiments we conducted with these three forms of constraints resulted in an NCC shape recovery metric of 0.6892 for OMP and QP, while LSQ resulted in 0.6810. The trend we observed in NCC is consistent with the trend in L1 errors, leading us to choose OMP as the method for choice for estimating weights.

**Varying Dictionary Sizes:** The next experiment we perform is in varying the size of dictionary \( (p_r, p_d) \), illustrated in Figure 6. The blue bar denotes an experiment with number of basis elements set to 50 selected using a clustering approach on the RGB feature and depth spaces, while the green bar denotes utilizing the entire set of training images as the dictionary in RGB and depth spaces. As can be observed from experiments, utilizing the entire dataset results in the best performance. This would mean that the input global descriptors, including Classemes [18], GIST [13] and HOG [3] for RGB images. We observed classemes to offer the best performance (see Figure 5), and we report results in the following utilizing Classemes as a global descriptor. Classemes capture object properties extremely well using RGB (color) channels, as shown by their impressive performance in object category recognition [18]. We utilize this capability of classemes to retrieve images with similar depth profiles to a given query image. For the depth channel, we utilize the entire depth image itself to be the feature since the goal is to predict the depth estimates at a pixel level.

**Global Descriptors:** We experimented with a variety of...
variable space is a PD descriptor $\gamma_i^k$ of size 1200, and the output space is a sparse positive descriptor $\delta_i^k$ also of size 1200, leading to large memory and computation requirements. Subsequently, we investigate if there is an alternate form of mapping that would increase speed of weight estimation, and leave a low memory footprint while preserving the performance of the original mapping. In other words, if there is a possibility of having a depth space of 50 dimensions, that would lead to creating a dictionary on the output space that is an order of magnitude lower. Towards this end, we first estimate a low dimensional manifold embedding of the data using t-SNE [19], and cluster data in the lower dimensional space using k-means clustering. Subsequently, we pick exemplars from each cluster, that share maximum correlation with other data points within the same cluster. The selected exemplars form the dictionary in the output space, yielding a significant speed up in the training procedure. The red bar in Figure 6 denotes the use of all training images as dictionary in the RGB space, while using only 50 exemplar images for the depth dictionary and demonstrates performance comparable to using full sized dictionaries on the output (green bars), but with much lower memory and time complexity.

Table 1. Performance comparison of proposed technique on 5 different metrics on Shuffled-I split of the data. As can be observed, proposed method outperforms state of the art techniques on all metrics.

<table>
<thead>
<tr>
<th>Methods</th>
<th>L1 Err (m)</th>
<th>NCC Metric</th>
<th>Relative Err</th>
<th>RMS Err</th>
<th>Log10 Err</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>0.6901</td>
<td>0.68529</td>
<td>0.2941</td>
<td>0.85205</td>
<td>0.11241</td>
</tr>
<tr>
<td>BG [10]</td>
<td>0.81695</td>
<td>0.61387</td>
<td>0.33981</td>
<td>1.0092</td>
<td>0.13172</td>
</tr>
<tr>
<td>NPS [9]</td>
<td>0.82514</td>
<td>0.6079</td>
<td>0.35482</td>
<td>1.0133</td>
<td>0.13301</td>
</tr>
<tr>
<td>GLB</td>
<td>0.90222</td>
<td>0.58058</td>
<td>0.38923</td>
<td>1.0988</td>
<td>0.14415</td>
</tr>
<tr>
<td>MK3D</td>
<td>3.4781</td>
<td>0.40379</td>
<td>1.6228</td>
<td>3.9521</td>
<td>0.36463</td>
</tr>
</tbody>
</table>

Table 2. Performance comparison of proposed technique on 5 different metrics on Scene-I split of the data. As can be observed, proposed method outperforms state of the art techniques on all metrics.

<table>
<thead>
<tr>
<th>Methods</th>
<th>L1 Err (m)</th>
<th>NCC Metric</th>
<th>Relative Err</th>
<th>RMS Err</th>
<th>Log10 Err</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>0.95427</td>
<td>0.69016</td>
<td>0.30543</td>
<td>1.2067</td>
<td>0.12886</td>
</tr>
<tr>
<td>BG [10]</td>
<td>1.1043</td>
<td>0.6214</td>
<td>0.33439</td>
<td>1.3933</td>
<td>0.1503</td>
</tr>
<tr>
<td>NPS [9]</td>
<td>1.0869</td>
<td>0.59858</td>
<td>0.34606</td>
<td>1.3593</td>
<td>0.14716</td>
</tr>
<tr>
<td>GLB</td>
<td>1.1421</td>
<td>0.606</td>
<td>0.34718</td>
<td>1.429</td>
<td>0.15581</td>
</tr>
<tr>
<td>MK3D</td>
<td>3.1141</td>
<td>0.39651</td>
<td>1.3075</td>
<td>3.6212</td>
<td>0.31102</td>
</tr>
</tbody>
</table>

4.1. Performance Comparison with State of the Art

In order to place our method in context with state of the art, we utilize four different techniques for comparison. The first method we utilize is BG [10] which is the most recent and directly related method to ours. We re-implement the technique that they reported to give best performance and utilize the same for comparisons. NPS [9] also provide source code for their technique, which we directly utilize for benchmarking. The third technique we use is Make3D. Since the code for training their system is not publicly available, we attempted to use their Plane Parameter MRF directly for benchmarking, where their models are trained with a bias towards outdoor scenes. Finally, we employ a vanilla method that computes a global prior by averaging across all ground truth depth maps.

From Table 1 and Table 2, it is evident that the proposed method has much better L1 and NCC scores in comparison to all other techniques considered. Further, the next best technique is that of BG [10], followed by NPS [9] and the utilization of global priors. The Plane Parameter MRF from Make3D has larger errors in absolute depth estimation since its training set has outdoor imagery. Since results produced by Make3D are in the form of smaller depth maps, ground truth depth maps were resized to compare with predictions. The trend we observe of all other techniques that trail the proposed method in performance, are consistent with those reported by the authors of BG [10] and NPS [9].

Dissecting Scene Specific Performance: The NYU-V2 dataset contains 27 unique scene categories, with a variable number of images per category. For instance, the dataset has a lot more living room scenes than study rooms. It is possible that the proposed technique, along with state of the art techniques may have variable performance across different scene categories. We study this distribution of performance in our next experiment. What we observe from extensive experiments (see Table 3, Table 4) indicates that the proposed technique consistently outperforms state of the art on a majority of scene categories.

Analysis of Running Time: In order for depth estimation techniques to be used in real world applications, they should be efficient with low running times. We now study the running times of all techniques given a query image. Figure 7 illustrates the running time of proposed technique in comparison with competing methods. The proposed technique only requires weight estimation and has fastest
running time, followed by BG [10] since it only requires a median estimation after nearest neighbor lookup. It should be noted that BG [10] requires a nearest neighbor lookup that could become very expensive with large scale datasets. This is in contrast to the fixed dictionary used in the proposed approach. The technique proposed by Saxena et al has a bottleneck in runtime since it has to infer plane parameters of an MRF at a pixel level. The work by Karsch et al is slowed down considerably due to the usage of a SIFT-flow matching procedure. In short, this section establishes computational advantages of using the proposed technique, without taking a massive hit on accuracy.

5. Conclusion

In summary, this work has presented a technique for depth recovery from static images. The salient features of the proposed approach are better representation of the data, and estimation of fusion weights (induction of a metric) for depth recovery. Since the proposed technique yields promising performance with low computation and memory footprints, it is ideally suited for mobile applications. Future work includes better local analysis schemes for fusing global and local cues in a unified framework.

References