Non-parametric Depth Estimation for Images from a Single Reference Depth

Zhi Yang Varun Chandola
State University of New York at Buffalo

February 28, 2014

Abstract

We present a non-parametric method for estimating depth of a single still image. We start from a single reference image and its corresponding 3-d depth and use an unsupervised neural network to transform the reference depth to represent the target image. In doing so, we attempt to mimic the human vision capability of perceiving the depth of a given image. Existing depth recovery methods either work for scenes with perpendicular planar surfaces or assume availability of a training database of known images and depths. We propose a method that can recover depth of a target image from a single reference depth. We redesign Self-organizing map (SOM) to learn in an environment with only three input data points and each data point with a different semantic meaning. We combine the proposed Parallel SOM (PSOM) with Gabor wavelets to handle discrepancy between the target and reference images in lighting and orientation. The proposed method gives promising results on images of faces and of daily objects even when using reference image and depth obtained in a poorly lighted setting.

1 Introduction

Humans have a remarkable capability to perceive the 3-d surface by looking at a two dimensional or monocular image. Enabling computer vision systems to do the same still remains a challenging task. Mathematically, the challenge arises from the fact that the problem is inherently ill-posed, one could construct infinitely many surfaces from a single image.

Several existing methods tackle the 3-d reconstruction problem in varying forms, ranging from estimating the depth of a 2-d image [25] to estimating a complete 3-d shape or surface model [10]. The input to such methods could be a single monocular view [24], a sequence of images [7] or even video [27]. Based on the type of assumptions, some methods are more appropriate for handling single objects such as faces [10] while others are applicable for 3-d scene reconstruction [25, 8]. Depending on the type of output, all methods operating on a single 2-d image rely on one or more reference 3-d models or depths and the corresponding 2-d images. Internally, the reconstruction methods work in one of two ways. First way is to use a *deformation* model (which is either designed using physics principles [28] or learnt from the reference data [1]) to transform the reference model to "match" the input image. Second way is to learn a *correspondence* relationship between an image and the corresponding 3-d model or depth from the reference data [14], and then apply it to the input image to obtain the 3-d model or depth.

In this paper, we focus on estimating the depth for a single 2-d image using a single reference image and its corresponding depth. In particular, we focus on generating realistic surface depth for a image consisting of a single face, though the proposed method can be applied to other objects also. Existing methods that deal with depth recovery from a single 2-d image typically assume that the image is composed of a set of planar surfaces [25, 9] which is a valid assumption for outdoor or indoor scenes, but not for single objects with relatively smooth textures, such as a human face. Moreover, such methods and one existing work that focuses on faces [19], fall under the

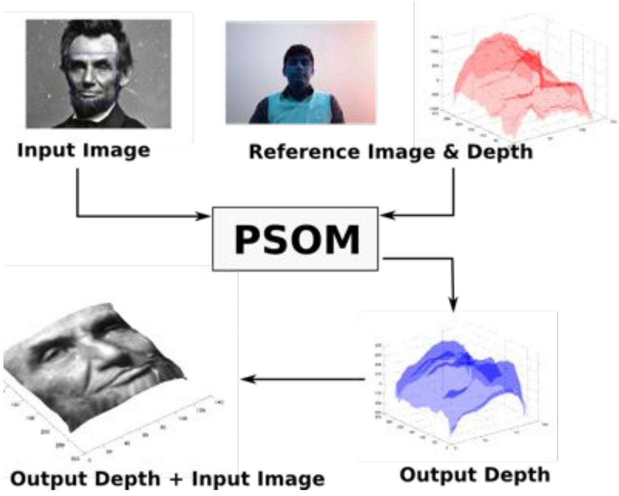


Figure 1: Overview of the proposed method. The target image (*top-left*) is combined with a reference image (*top-middle*) and the reference depth (*top-right*) to generate a depth map (*bottom-right*) which is combined with the original target image to produce a realistic 3-d surface (*bottom-left*).

correspondence category and hence require a representative training set of reference images and depths.

We propose a non-parametric method for depth estimating which does not involve learning a parametrized correspondence model. Instead, the proposed method uses the relationship between the input image and the reference image to "transform" the reference depth to match the input image, as shwn in Figure 1. Thus, the proposed method, is similar in spirit to the *deformation* model approaches proposed for 3-d surface recovery, except that the deformation is performed in a non-parametric fashion and the output is a depth map.

We combine two well-known learning mechanisms that have strong foundations in the functioning of human brain for visual recognition, viz., *Self-Organizing Maps* and *Gabor filters*. It has been shown that the visual cortex of the brain in mammals can be modeled as a set of Gabor filters [5]. Self-organizing Map (SOM) [11] is an unsupervised neural network with strong links to retina-cortex mapping. We propose *Parallel SOM* (PSOM), a completely novel way to use SOMs when the inputs have different semantic meanings. We demonstrate, both visually and quantitatively, that the proposed method, with modest requirements regarding the reference, show promising results for depth recovery for facial images and other objects.

2 Related Works

One category of 3D surface reconstruction methods adopt the *structure from motion* approach. For such methods, the input is either a sequence of monocular images [2], a video stream [21, 6], or a sequence of depth images taken by an RGB-D sensor, such as Kinect [16, 27, 7]. Given that the problem is ill-posed, additional constraints are employed to obtain a solution.

The second category of surface reconstruction methods use a template based approach [3, 30, 14]. Such methods start from a known reference 3D surface shape and then establish point correpondence between the reference shape and the input image. These correspondences are used to "deform" the reference shape and reconstruct the new shape. Obviously, this reconstruction is also an under represented problem and many solutions exist. To reach an acceptable solution, constraints are applied.

We deal with the problem of recovering 3D surface from a single two dimensional image or a view. This problem is also referred to as *monocular surface reconstruction* [3, 14]. Recovering 3D surface is fundamentally an ill-posed problem. This means that for a given 2D image, an infinite number of 3D surfaces can be constructed.

In general, all existing methods follow the same basic steps, to do such reconstruction. The first step is to start with an existing 3D surface **model**. The model is then *deformed* to *match* the input 2D image and the resulting model is the desired output shape. Existing methods can be grouped into different categories based on the model learnt and the deformation principles used.

In general, most existing methods for surface recovery can be grouped into following categories:

1. *Using physics-inspired deformable models* [28]. Such models rely on the knowledge about the surface material and use complex objective functions to model the deformation of the surface.

2. *Learning a deformable model from the data* [1]. Active appearance model is one such example of a sta-

tistical model learnt from the data.

3. *Using a structure from motion approach* [29]. Such methods use image sequences to reconstruct the surface and are only effective for small deformations.

4. *Using point correspondences between the input image and a reference 3D shape* [18, 22, 3, 30, 14]. Recently, such methods, also known as *template based reconstruction*, have gained popularity. Such methods reconstruct the surface as a deformation of the reference surface using certain constraints, which could be physical [18, 23, 14] or statistical [22, 17]. Our proposed method is closest to the third category of methods (template based reconstruction). Some methods [23, 22, 17] in this category assume that the deformation modes are available.

2.1 3D Surface Models

An example of a surface model, especially in the context of human faces, is an *Active Appearance Model* [4] which is a statistical model of the shape, and has been typically used for matching an input image with a model (recognition) [4, 15] rather than for surface recovery. The idea is to match the input image by finding a synthesized image from the surface model which is as close to the input image as possible, typically through an optimization mechanism.

2.2 Models for Extensible and Inextensible Objects

Models for extensible or non-rigid objects assume a wide variety of shapes and can track complex motions.

3 Background

In this section we briefly describe two analysis tools that we use for the depth estimation task. We use *Self-organizing Maps* as the core non-parametric learning algorithm and 2-d *Gabor Wavelet Analysis* for feature enhancement of the images.

3.1 Self-Organizing Maps

A Self-organizing Map (SOM) [11] is an unsupervised neural network which is used to map high dimensional data onto a low dimensional (typically 2) grid, such that each input is quantized into discrete nodes (or neurons) on the grid. A SOM also preserves topological relationship among the neurons. SOMs have been widely used for visualizing high dimensional data on 2-d grid and has also been shown to be a non-linear generalization of Principal components analysis (PCA) [32]. SOM have been an interesting concept for the vision community as it is strongly motivated from the retina-cortex mapping [20].

Traditionally, there are two operational modes for a SOM, training and mapping. In the training mode, the SOM updates the weights at each neuron using a sequence of learning examples and in the mapping mode a test input example is mapped from a high-dimensional space onto a low dimensional grid.

During training, the learning example is compared to the weight vectors associated with each neuron and the "closest" *winning neuron* is selected. Typically, Euclidean distance between the input and the weight is used for finding the winnder. The weights of all the neurons are then updated using the following update equation:

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \alpha(t)\eta(\nu, k, t)||\mathbf{w}_k(t) - \mathbf{x}|| \quad (1)$$

Here $\mathbf{w}_k(t)$ is the weight for the k^{th} neuron at iteration t, $\mathbf{x}$ is the input vector, and ν is the index of the winning neuron. $\alpha()$ gives the learning rate which monotonically decreases with t. $\eta(,,)$ is a neighborhood function which measures the distance between a given neuron and the winning neuron. Typically, η takes a Gaussian form, $\eta(\nu, k, t) = \frac{\Delta_{\nu,k}}{2\sigma(t)^2}$, where $\Delta(,)$ is the distance between two neurons on the grid, and σ is the monotonically decreasing neighborhood width.

The SOM algorithm assumes that the input vectors are *semantically homogeneous*. In Section 4.1 we relax this assumption to the case when different inputs have different semantic connotations.

3.2 Image Processing using Gabor Wavelets

We deal with reference and the target images which are expected to be different in terms of lighting conditions and orientation. To address this issue, we use local features extracted from the raw images using 2-d *Gabor functions*, which essentially are convolution functions representing plane waves restricted by a Gaussian envelope function. Gabor functions have been shown to have strong connections with the way human visual cortex processes images [5]. They have been especially effective for face recognition [31, 26] as they are robust to varying brightness in images and to limited variations in the orientation.

3.2.1 Gabor Wavelets

A 2-d Gabor function is parameterized by the frequency of the sinusoidal plane wave, the orientation of the major axis of the Gaussian envelope and the center location. Multiple functions can be created by starting from a *mother wavelet* and varying these parameters. A family of such functions is called *Gabor wavelets*. Given a mother wavelet defined as:

$$\psi(x,y) = \frac{f^2}{\pi\gamma\eta} \exp\left(\frac{f^2}{\gamma^2}x_r^2 - \frac{f^2}{\eta^2}y_r^2\right) \exp((j2\pi f x_r)) \tag{2a}$$

$$x_r = \quad x\ cos\theta + y\ sin\theta \tag{2b}$$

$$y_r = -x\ sin\theta + y\ cos\theta \tag{2c}$$

a family of Gabor functions (or wavelets) can be constructed by rotating and dilating[1] the mother wavelet as follows:

$$\psi_j(x,y) = 2^{-2u_j}\psi(x',y') \tag{3a}$$

$$x' = 2^{-u_j}[x\ cos\phi_j + y\ sin\phi_j] \tag{3b}$$

$$y' = 2^{-u_j}[-x\ sin\phi_j + y\ cos\phi_j] \tag{3c}$$

Note that u_j controls the scale and ϕ_j controls the rotation of a child wavelet with respect to the mother wavelet. The index j is obtained by using n_u different values for u and n_ϕ different values for ϕ to get $K = n_u \times n_\phi$ wavelets. Thus $1 \leq j \leq K$.

3.2.2 Image Recovery from Gabor Wavelets

Using the wavelet family, one can reconstruct a given image I using the following:

$$\hat{I} = \sum_{j=1}^{K} w_j\psi_j \tag{4}$$

where w_j is a weight assigned to the j^{th} wavelet. Let $\mathbf{w} = [w_j]_{j=1}^{K}$ denote the weight vector. The optimal weights are obtained by minimizing the energy function [12]:

$$E = ||I - \hat{I}||_2^2 \tag{5}$$

with respect to the weights w_j.

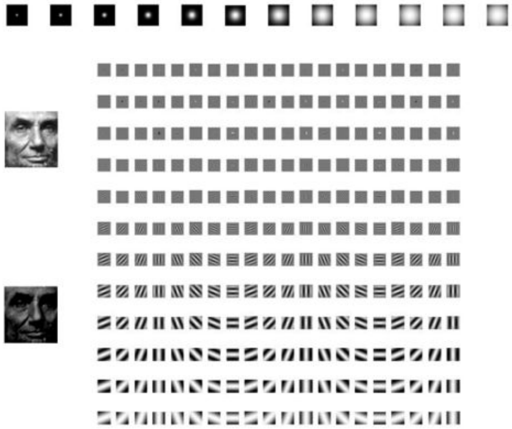


Figure 2: Gabor wavelets used in the paper composed of 12 different scales and 20 different orientations. First row represents the magnitude of the 12 scales. On the bottom-left corner, we show an input Lincoln's grayscale image and reconstructed figure using the Gabor wavelet family.

In this paper, we employ *Gabor wavelets* to extract features from the input images in the pre-processing step and also to "smooth" the output depth in the post-processing step. The wavelet family used in the paper is shown in Figure 2.

4 Methodology

We use a Self Organizing Map (SOM) based algorithm to solve the following problem:

Given a 2-d target image and a reference 2-d image and depth, estimate the depth for the target image.

In this paper we work with grayscale images though the same methodology can be easily extended for color images.

[1] One can also *translate* the mother wavelet but we ignore that in this paper.

The 2-d image and the depth map are represented as 2-d matrices. The reference image is denoted as R, the corresponding reference depth is represented as $\bar{R}$ and the target image is represented as T. The task is to estimate the corresponding target depth $\bar{T}$ using $R, \bar{R}$, and T. We assume that $R, \bar{R}, T$, and $\bar{T}$ are $r \times c$ in size.

The core idea behind our proposed work is to *transform* (or deform) the reference depth, $\bar{R}$ to get the target depth $\bar{T}$ by exploiting the relationships between R and $\bar{R}$ and between R and T. Most existing methods perform this transformation in one step by applying a function or procedure. The function is either learnt from training data or designed using physics based principles. But given that our training data is just a single pair ($\langle R, \bar{R}\rangle$) and we are not making any assumptions regarding the physical characteristics of the objects, we perform the transformation incrementally using SOM.

The proposed method has three steps. In the first step, we use Gabor wavelets to pre-process the raw input and target images (See (5)) using the wavelets depicted in Figure 2. The preprocessed reference and target images are denoted as R' and T', respectively. In the second step (See Section 4.1), we use SOM to transform the reference depth $\bar{R}$ to the target depth $\bar{T}$. In the third step (See Section 4.2), we again use Gabor wavelets to smooth the estimated depth $\bar{T}$ based on the reference depth $\bar{R}$.

4.1 Using Parallel Self Organizing Maps for Depth Estimation

As described in Section 3, a SOM is traditionally used to represent a low-dimensional representation of input data. The representation is achieved by repeatedly updating the weight vectors associated with the neurons arranged on a low-dimensional (typically two) grid, using a sequence of input examples.

We have adapted traditional SOMs to work towards a completely different objective. We refer to this adaptation as a **Parallel SOM** or PSOM. In our setup, we have only three data points available for training, *viz.* R', $\bar{R}$, and T', and each of these have completely different semantics. To handle this issue we make two key modifications. First, weight vectors are replaced with weight matrices ($r \times c$). Second, each neuron has two associated weight matrices, one corresponding to the 2-d images (*image-weights*) and second corresponding to the depths (*depth-weights*).

The image-weights are "trained" using the reference and target images alternately. This training "induces" a bias to each neuron towards either the reference or the target image, also referred to as *polarization*. This bias is propagated to the depth-weights such that the neurons heavily biased towards the target image will *eventually* have the depth-weight corresponding to the desired target depth.

We represent the PSOM as a set of neurons on a two dimensional $a \times b$ lattice (for this paper $a = b = 10$). Each neuron is denoted as $\mathcal{N}_{ij}$, where i, j specifies its location on the 2-d lattice, and has two $r \times c$ matrices associated with it, denoted as I_{ij} and D_{ij}, corresponding to the image-weights and depth-weights, respectively.

4.1.1 Initialization

For each neuron $\mathcal{N}_{ij}$, a $r \times c$ random matrix is generated and both I_{ij} and D_{ij} are initialized to that matrix.

4.1.2 Single Round

After initialization, the algorithm runs as multiple rounds. Each round consists of two sequential steps. In the **first step**, a winning neuron is identified by comparing the image-weight matrices with the target image T. The index of the winning neuron is calculated as:

$$[\hat{i}, \hat{j}] = \operatorname*{argmin}_{i,j} ||T - I_{ij}||_F \tag{6}$$

where $||A||_F$ is the *Frobenius norm* for a matrix A. The image-weights for all neurons, with $[\hat{i}, \hat{j}]$ as the index of the winning neuron and T' as the input, is updated using the SOM update rule (See (1)). The depth-weights for all neurons are similarly updated, using the same winning neuron and $\hat{R}$ as the input.

In the **second step**, a (possibly) different winning neuron is identified by comparing the image-weight matrices with the reference image R' in the same manner as step 1. The image-weights for all neurons are again updated but using the new winning neuron and R' as the output. Similarly, the depth-weights are also updated using the new winning neuron and $\hat{R}$ as the output.

4.1.3 Convergence

The single round of the algorithm is executed multiples times. The learning rate and the neighborhood width parameters of the SOM (α and σ) are functions of the round number and decrease monotonically. The algorithm can be stopped using different convergence criteria, e.g., measuring the change in the weight vectors over successive iterations. For our experiments we allow the algorithm to execute for a fixed number of rounds (2000).

4.1.4 Output

At the end of the final round, the target image (T') is compared with the the image-weights (I_{ij}) of all neurons using PCA. All image-weight matrices and target image matrix are "flattened" and stacked together to get a matrix A with ($a \times b + 1$ rows and $r \times c$ columns. Each row of A is mapped into a k dimensional space using the top k principal components to get matrix A_k which has k columns. The row in A_k corresponding to T' is compared with all other rows using Euclidean distance. The neuron corresponding to the closest row is chosen as final winner. Let $[i^*, j^*]$ be the index of the closest neuron. Then $D_{i^* j^*}$ is the depth estimate for the target image (denoted as $\bar{T}'$).

The algorithm for estimating the target depth using PSOM is described in Algorithm 1.

Algorithm 1: Parallel SOM Depth Estimation

```
Input: Preprocessed Reference Image $R'$, Reference
       Depth $\bar{R}$, Preprocessed Target Image $T'$,
       Number of rounds $n$
Output: Target Depth Estimate $\bar{T}'$
/* Initialization                                   */
for $i \leftarrow 1$ to $a$ do
    for $j \leftarrow 1$ to $b$ do
        $I_{ij} \leftarrow random(r, c)$
        $D_{ij} \leftarrow I_{ij}$
for $c \leftarrow 1$ to $n$ do
    /* Step 1 -- Using target                       */
    Find winning neuron $\nu = \mathcal{N}_{\hat{i},\hat{j}}$ for $T'$ using (6)
    for $i \leftarrow 1$ to $a$ do
        for $j \leftarrow 1$ to $b$ do
            Update $I_{ij}$ w.r.t. $\nu$ and $T'$ (See (1))
            Update $D_{ij}$ w.r.t. $\nu$ and $\bar{R}$ (See (1))
    /* Step 2 -- Using reference                    */
    Find winning neuron $\nu = \mathcal{N}_{\hat{i},\hat{j}}$ for $R'$ using (6)
    for $i \leftarrow 1$ to $a$ do
        for $j \leftarrow 1$ to $b$ do
            Update $I_{ij}$ w.r.t. $\nu$ and $R'$ (See (1))
            Update $D_{ij}$ w.r.t. $\nu$ and $\bar{R}$ (See (1))
/* Finding Best Estimate                            */
Find final winning neuron $\mathcal{N}_{i^*,j^*}$ using PCA (See
Section 4.1.4)
return $D_{i^*,j^*}$
```

4.2 Using Gabor Wavelets for Smoothing Depth Estimates

The output of PSOM ($\bar{T}'$) is typically not smooth since it is derived from the target image which could have shaded or dark areas with irrecoverable depth. To handle this issue we perform a final smoothing step on the depth estimate obtained from the PSOM estimation ($\bar{T}'$). Using the K Gabor wavelets discussed in Section 3.2 we calculate K convolutions (also known as *Gabor jets*), denoted by $\mathcal{J}_i$ as:

$$\mathcal{J}_j(x, y) = \int_{x',y'} \bar{T}'(x, y)\psi_j(x - x', y - y')\, dx'\, dy' \quad (7)$$

We then find a set of weights $\{\bar{w}_j\}_{j=1}^{K}$ such that the depth reconstructed using the Gabor jets is as close to the reference depth as possible, i.e.,

$$\{\bar{w}_j\}_{j=1}^{K} = \operatorname*{argmin}_{\bar{w}_j} ||\bar{R} - \sum_{j=1}^{K} \bar{w}_j \mathcal{J}_j||_2^2 \quad (8)$$

The final smoothing step "adjusts" regions where the corresponding area in the target image is dark with information from the reference depth. Figure 3 shows the effect of this smoothing. In the original target image (See Figure 1), the *chin* area of the face is dark and hence the region is not well recovered in the depth estimated by PSOM (shown on left in Figure 3). But after smoothing the chin region is more pronounced (shown on right in Figure 3).

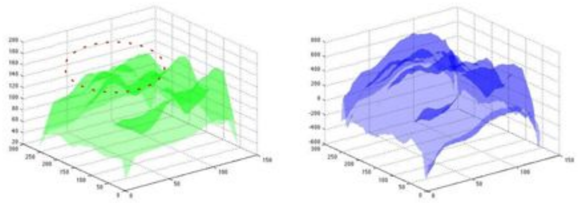


Figure 3: Effect of smoothing. Using Gabor jets, the missing chin area in the left depth estimate (top left area of the graph) gets filled out.

5 Experiments

We experimented with two types of images. We first used our methodology to estimate depth for a set of face images. We used the image shown in Figure 1 and the corresponding depth as reference. The depth was captured using a ASUS XTION RGB-D sensor. The second evaluation was conducted on a public dataset of RGB and depth images of several objects [13].

5.1 Analysis of PSOM

We first show how the PSOM estimation algorithm performs the parallel learning such that after each round the depth-matrix associated with every neuron adapts itself according to the corresponding image-matrix. For this experiment we recover depth for Lincoln's face (See Figure 1). Figure 4 shows how the depth-matrices adapt over multiple iterations. In the figure, results at every 100^{th} iteration is shown. The top row is for the winning neuron corresponding to the reference image (as obtained in Line 6 in Algorithm 1). The bottom row is for the winning neuron corresponding to the target image (as obtained in Line 11 in Algorithm 1). The middle row is for a randomly picked neuron.

As shown in Figure 4, as the image-weights of the neurons get biased by the reference and target images, the information from the image-weights is *propogated* to the corresponding depth-matrices. At any given iteration, there is one neuron that is most biased by the reference image and one neuron that is most biased by the target image. As shown in the figure, the depth-matrix for these neurons show most affinity to the corresponding depth, while the randomly selected neuron shows a "mixed" evolution.

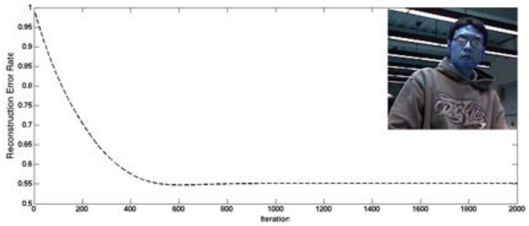


Figure 5: Error rate between estimated depth and true depth for the target image shown in inset.

To further understand how PSOM improves the target depth estimate, we measure the *reconstruction error* as the Frobenius norm of the estimated and true depth matrices for a target image with known depth. Figure 5 shows how the error varies over iterations. Beginning with a very high error (since the estimate is a random matrix), the error decreases over iterations and finally stabilizes after 1000 iterations.

5.2 Recovering Depth for Faces

Starting with the reference image and depth, we use the proposed method to recover depths for different faces, including oil paintings and sculptures. The results are shown in Figure 6.

Our method is able to satisfactorily recover the depth for most of the target images, starting with only one reference image and depth map. Note that the reference image is also obtained under poorly lighted condition and hence has an impact on the final estimates. The second row shows the reconstructed 2-d images using Gabor wavelets and shows how the variations in shading and lighting conditions are normalized by the pre-processing step.

5.3 Recovering Depth for Objects

To understand the performance of the proposed method on images of non-face objects we used several publicly available object images and depths [13]. For each type of

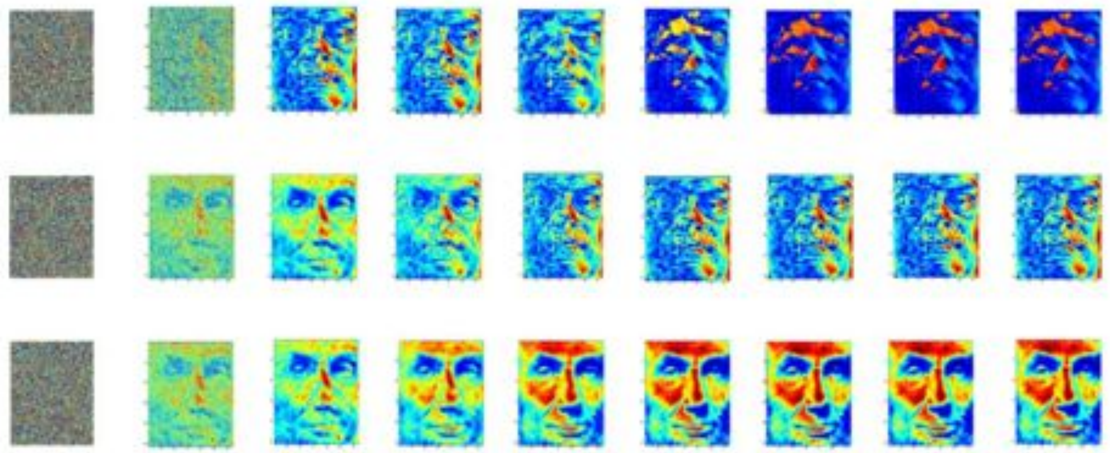


Figure 4: Evolution of depth-matrices over PSOM iterations. Results after every 100^{th} iteration are shown starting with the initial random matrices. *Top:* winning neuron for reference image, *middle:* randomly selected neuron, *bottom:* winning neuron for target image.

object (e.g., apple), we chose one image as reference and another image as target. We also measure the reconstruction error to quantitatively assess the performance of our method. The results are summarized in Table 1.

Table 1 show that for most objects our method is able to reconstruct the depth within 10% error. Flat surfaces such as notebook and food bag are recovered poorly since the depth does not vary significantly enough across the surface to cause the "polarization" in the PSOM neurons.

6 Conclusions and Future Work

We have shown a depth recovery method that uses only one reference image and depth to reconstruct the depth for a target image. Given the limited information required by our method, the results are promising, even when reference image is taken under poor lighting conditions. The method uses Gabor filters which are able to handle modest discrepancies between the reference and target images in terms of lighting and orientation. While we mainly talked about recovering depth for facial images, our experiments on other types of objects (See Section 5.3) show that the method is equally applicable to those.

We use two processing and learning tools, viz. self organizing maps and Gabor filters which have strong connections with how the visual cortex in human brains work. A key contribution of our paper is the use of SOM to simultaneously learn from three semantically different inputs. Traditionally, SOMs, and neural networks in general, have assumed the inputs to have homogeneous semantic sense. We completely redesign SOMs to handle the heterogeneous case.

Obviously, the performance of the method is strongly tied to the reference image and depth. For a given target image, one reference image might work better than another. It makes sense to have a database of reference images and depths and then select the "best match" for a target image to be used in the proposed depth estimation system. Another possibility is to train the SOM with multiple reference images (have $k + 1$ steps in each round in Algorithm 1). This is will be investigated as future work.

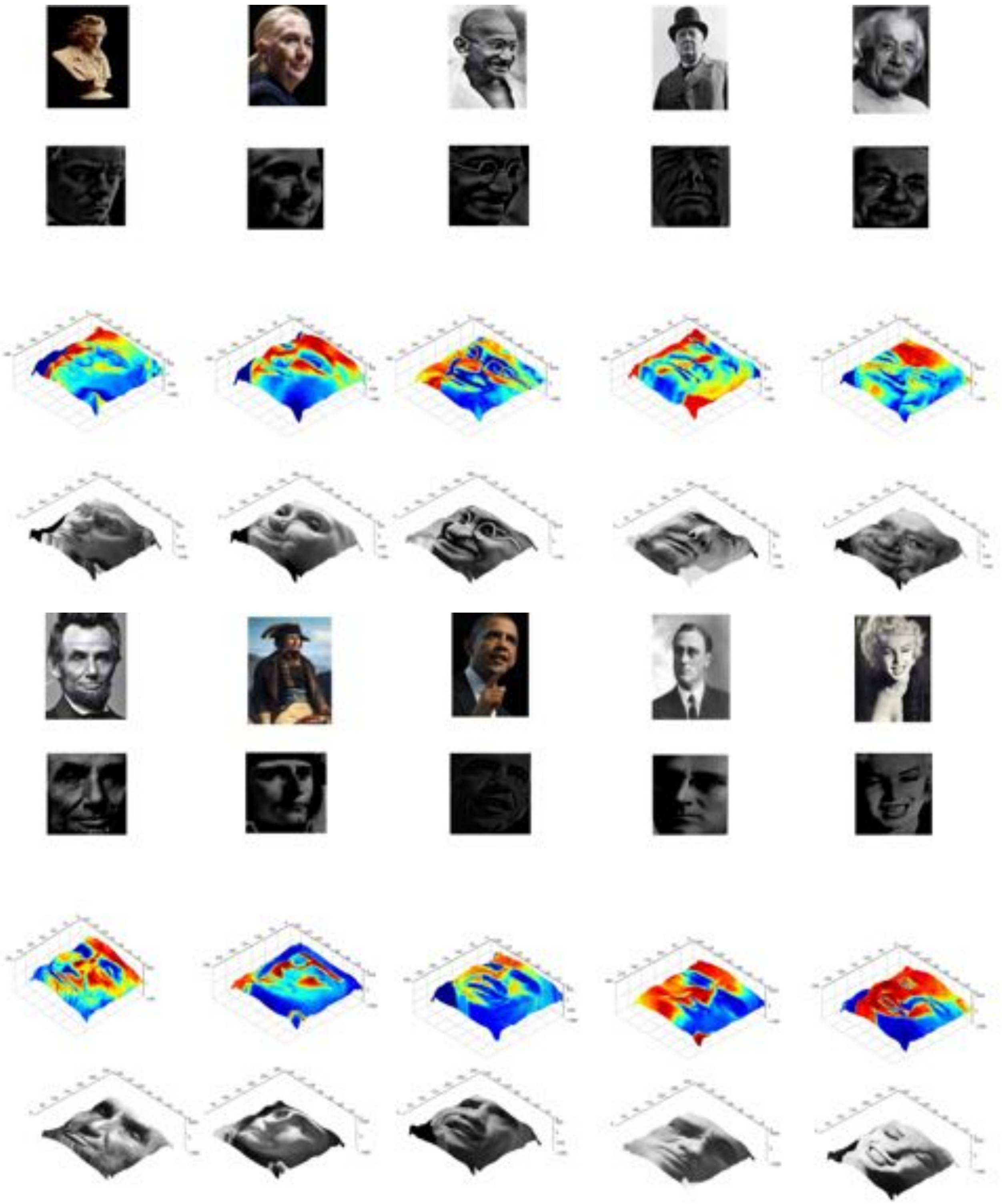


Figure 6: Depth recovery using PSOM for faces of 10 well known people. First row: Original image with bounding box. Second row: Reconstructed image using Gabor wavelets. Third row: Estimated depth shown as intensity map. Fourth row: Depth warped on the original image.

References

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of SIGGRAPH*, pages 187–194, New York, NY, USA, 1999. 1, 2

[2] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 690–696 vol.2, 2000. 2

[3] F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. Monocular template-based reconstruction of smooth and inextensible surfaces. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part III*, ACCV'10, pages 52–66, Berlin, Heidelberg, 2011. Springer-Verlag. 2, 3

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In H. Burkhardt and B. Neumann, editors, *Computer Vision ECCV98*, volume 1407 of *Lecture Notes in Computer Science*, pages 484–498. Springer Berlin Heidelberg, 1998. 3

[5] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7):1160–1169, 1985. 2, 4

[6] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. June 2013. 2

[7] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *Int. J. Rob. Res.*, 31(5):647–663, Apr. 2012. 1, 2

[8] D. Hoiem, A. A. Efros, and M. Hebert. Recovering Surface Layout from an Image. *Int. J. Comput. Vision*, 75(1):151–172, Oct. 2007. 1

[9] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *Proceedings of the 12th European conference on Computer Vision - Volume Part V*, ECCV'12, pages 775–788, Berlin, Heidelberg, 2012. Springer-Verlag. 1

[10] I. Kemelmacher-Shlizerman and R. Basri. 3d face reconstruction from a single image using a single reference face shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2):394–405, 2011. 1

[11] T. Kohonen. Neurocomputing: foundations of research. chapter Self-organized formation of topologically correct feature maps, pages 509–521. MIT Press, Cambridge, MA, USA, 1988. 2, 3

[12] V. Kruger and G. Sommer. Efficient head pose estimation with gabor wavelet networks. In *British Machine Vision Conference*, pages 8.1–8.10, 2000. 4

[13] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, pages 1817–1824. IEEE, 2011. 7

[14] A. Malti, R. I. Hartley, A. Bartoli, and J.-H. Kim. Monocular template-based 3d reconstruction of extensible surfaces with local linear elasticity. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1522–1529, 2013. 1, 2, 3

[15] I. Matthews and S. Baker. Active appearance models revisited. *Int. J. Comput. Vision*, 60(2):135–164, Nov. 2004. 3

[16] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136. IEEE, 2011. 2

[17] M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. *Int. J. Comput. Vision*, 95(2):124–137, Nov. 2011. 3

[18] M. Prasad and A. Fitzgibbon. Single view reconstruction of curved surfaces. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1345–1354, 2006. 3

[19] M. Reiter, R. Donner, G. Langs, and H. Bischof. 3d and infrared face reconstruction from rgb data using canonical correlation analysis. In *Proceedings of the 18th ICPR*, pages 425–428, 2006. 1

[20] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps; An Introduction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1992. 3

[21] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. *ACM Trans. Graph.*, 21(3):438–446, July 2002. 2

[22] M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1054–1061, 2009. 3

[23] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3d shape recovery. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008. 3

[24] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, 2005. 1

[25] A. Saxena, M. Sun, and A. Y. Ng. Make3d: depth perception from a single still image. In *Proceedings of the 23rd AAAI*, pages 1571–1576, 2008. 1

[26] L. Shen and L. Ba. A review on gabor wavelets for face recognition. *Pattern Analysis and Applications*, 9(2):273–292, 2006. 4

[27] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng. Point-plane SLAM for hand-held 3D sensors. In *IEEE International Conference on Robotics and Automation*, 2013. 1, 2

[28] L. V. Tsap, D. Goldgof, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 728–734, 1998. 1, 2

[29] A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A constrained latent variable model. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2248–2255, 2012. 3

[30] A. Varol, A. Shaji, M. Salzmann, and P. Fua. Monocular 3d reconstruction of locally textured surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(6):1118–1130, 2012. 2, 3

[31] L. Wiskott, J.-M. Fellous, N. Kroger, and C. V. D. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:775–779, 1997. 4

[32] H. Yin. Learning nonlinear principal manifolds by Self-Organising maps. 60:1, 2007. 3

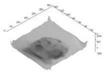
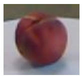
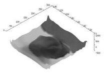
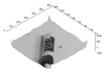
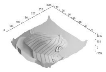
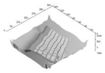
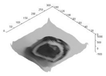
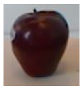
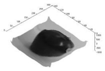
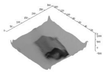
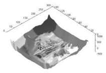
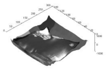
Object	Reference	Target	Recovered Depth	Error (%)
Mushroom				0
Peach				2
Marker				2
Cap				5
Keyboard				7
Ball				9
Apple				11
Dry Battery				14
Food Bag				34
Notebook				67

Table 1: Performance of PSOM Based Depth Recovery for Objects