KINECT DEPTH RESTORATION VIA ENERGY MINIMIZATION WITH TV_{21} REGULARIZATION

Shaoguo Liu\textsuperscript{1}, Ying Wang\textsuperscript{1}, Jue Wang\textsuperscript{2}, Haibo Wang\textsuperscript{3}, Jixia Zhang\textsuperscript{1} and Chunhong Pan\textsuperscript{1}

1. Institute of Automation, Chinese Academy of Sciences
2. Adobe Systems 3. BME Department, Case Western Reserve University
E-mail: sgliu@nlpr.ia.ac.cn

ABSTRACT

Depth maps generated by Kinect cameras often contain a significant amount of missing pixels and strong noise, limiting their usability in many computer vision applications. We present a new energy minimization method to fill the missing regions and remove noise in a depth map, by exploiting the strong correlation between color and depth values in local image neighborhoods. To preserve sharp edges and remove noise from the depth map, we propose to add a TV\textsubscript{21} regularization term into the energy function. Finally, we show how to effectively minimize the total energy using an alternating optimization approach. Experimental results show that the proposed method outperforms commonly-used depth inpainting approaches.

Index Terms— Depth Inpainting, Depth Denoising, Energy Minimization, TV\textsubscript{21} Prior

1. INTRODUCTION

Depth information plays an important role in many computer vision applications such as augmented reality, scene reconstruction, 3DTV, etc. The recently proposed Microsoft Kinect is a revolutionary product that enables real-time, low-cost depth acquisition. However, the depth maps provided by Kinect often contain large holes (i.e., pixels without depth value) and significant noise, as shown in Fig. 1. These artifacts greatly limit the practical usage of the Kinect device in real applications. Therefore, depth map restoration, i.e., filling holes and removing noise, becomes an essential pre-processing step for systems that use Kinect for depth acquisition.

Many methods have been proposed for restoring Kinect depth maps, which can be roughly classified into two categories: filtering-based methods and reconstruction-based ones. Filtering-based methods seek to apply various image filters to enhance a Kinect depth map. The simplest choice is to recursively apply a median filter \cite{1} in the RGB-D space to fill in the holes in a depth map, however it will also significantly blur the sharp edges in it. To fill the holes while preserving sharp edges, Camplani and Salgado \cite{2} instead iteratively applied a joint bilateral filter \cite{3}. Matynin et al. \cite{4} took temporal information into account for restoring depth maps, but to restore a target frame this approach requires multiple consecutive frames around it, thus yields delay. Qi et al. \cite{5} proposed a fusion-based method with the non-local filtering \cite{6} scheme for restoring depth maps. He et al. \cite{7} proposed a guided filter that can preserve sharp edge and avoid reversal artifacts when smoothing a depth map. However, these filter-based approaches often yield unsatisfactory results when large holes exist in the depth map.

On the other hand, reconstruction-based methods apply image inpainting techniques to reconstruct missing depth values. Tea et al. \cite{8} proposed a fast marching method (FMM) for image inpainting. Since it is designed for color images rather than depth maps, it often yields unsatisfactory results for the latter. Liu et al. \cite{9} proposed an extended FMM approach for depth inpainting, followed by the guided filtering \cite{7} for post-processing. Miao et al. \cite{10} proposed a texture-assisted approach in which the texture edge information is extracted for assisting depth inpainting. Nevertheless, these inpainting methods do not adequately consider depth discontinuities between objects which cause sharp edges in the depth maps, thus are not able to faithfully recover them.

In this paper, we propose a new energy minimization-based depth restoration technique that can fill in large depth holes, remove severe noise and preserve sharp edges simultaneously. Inspired by recent progress in image matting research \cite{11}, we assume that the depth values and image colors have a linear correlation in a small image neighborhood, and design an energy function based on it. Furthermore, considering the characteristics of depth maps, we choose to incorporate a TV\textsubscript{21} prior into the total energy, which is capable of keeping sharp boundaries and removing noise. This prior leads to a significant improvement on the final depth inpainting results. To the best of our knowledge, we are the first to introduce the TV\textsubscript{21} prior to the Kinect depth restoration problem.

2. METHOD

Our approach consists of three main technical components. Firstly, we present a novel energy minimization framework for depth
restoration. Secondly, in order to remove noise while preserving sharp edges in the depth map, a $TV_{21}$ regularization is incorporated into the framework. Finally, we show how the total energy can be effectively minimized using an alternating minimization scheme.

2.1. Depth Inpainting by Energy Minimization

The Kinect device captures a color image along with a depth map at the same time. Since the depth map and the color image are two descriptions of the same scene, there exists a strong correlation between them. Based on the observation, we assume that within a small local window $N_k$ centered at pixel $k$ (i.e., $5 \times 5$ or $7 \times 7$), the depth values $r$, and pixel colors $c$, satisfy the following linear regression model:

$$r_i = w_i^T c_i + b_k, \quad \forall i \in N_k,$$

where $w_k = (w_{k1}, w_{k2}, w_{k3})^T \in \mathbb{R}^3$, $b_k \in \mathbb{R}$ are the linear regression coefficients. Based on this local color-depth regression model, for depth restoration we seek to find $w, b, r$ that can minimize the following energy function:

$$J(w, b, r) = \int_{k \in S} \left( \int_{i \in N_k} G_0(k, i)|w_i^T c_i + b_k - r_i|^2 di + \lambda_1|w_k|^2 \right) dk,$$

where $G$ is a Gaussian kernel with width $\phi$, which weights the contribution of each pixel in the neighborhood (e.g. pixels near the center of the window have higher weights). $\lambda_1|w_k|^2$ is a regularization term to avoid over-fitting. For the purpose of hole filling, we would like to maintain the depth values for known pixels, and use them to recover unknown pixels. This can be achieved by adding a constraint in Eqn. 2 as:

$$\min J(w, b, r) \text{ s.t. } r_i = S_i, \forall i \in S,$$

where $S$ is the set of pixels with known depth values. Minimizing this energy yields regression parameters $w_k$ and $b_k$ for every unknown pixel $k$, which allows us to recover its depth value with the use of Eqn. 1.

2.2. Adding $TV_{21}$ Prior

Minimizing the energy in Eqn. 3 alone allows us to fill the holes in the depth map, however it cannot produce sharp edges in these holes, as well as removing noise from known regions. To resolve these two issues, we introduce an additional $TV_{21}$ prior on the depth map. The $TV_{21}$ prior was originally proposed by Alvaro and Suvrit [12] for image denoising and deconvolution. Here we describe the $TV_{21}$ prior on depth maps in its continuous form:

$$TV_{21}(r) = \int \left( \int r_x^2(x, y) dy \right)^\frac{1}{2} dx + \int \left( \int r_y^2(x, y) dx \right)^\frac{1}{2} dy$$

$$= G_1(x)dy + \int G_2(y)dx,$$

where $G_1(x) = (\int r_x^2(x, y) dy)^\frac{1}{2}$, and $G_2(y) = (\int r_y^2(x, y) dx)^\frac{1}{2}$. Intuitively, the $TV_{21}$ prior encourages the gradient of the depth map to be sparse both horizontally and vertically, which is essential for restoring sharp edges and removing noise in the depth map, as shown in the illustrative example in Fig. 2.

Finally, the complete energy function for depth restoration thus becomes:

$$\min J(w, b, r) + \lambda_3 TV_{21}(r), \quad \text{s.t. } r_i = S_i, \forall i \in S,$$

where $\lambda_3$ is the weight of the $TV_{21}$ prior.

2.3. Alternating Energy Minimization

To minimize the constrained energy function in Eqn. 5, a Lagrange multiplier can be used to convert it into an unconstrained minimization problem as:

$$\min_{k \in S} \left( \int \left( \int G(k, i)|w_k^T c_i + b_k - r_i|^2 di + \lambda_1|w_k|^2 \right) dk + \int_\lambda_3 TV_{21}(r), \right)$$

where $\lambda_i = \begin{cases} \lambda_2 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$. There are three unknown variables $w, b, r$ in Eqn. 6. In this work we utilize a two-step energy minimization approach which alternatively fixes some variables and solves for others using gradient descent. The initial condition of the alternating approach is $r = r_0$, the input depth map.

**Step 1:** We fix $r_i$, and take derivatives of Eqn. 6 with respect to $b_k$ and $w_k$, and then set them to zero:

$$\int G(k, i)(w_k^T c_i + b_k - r_i)di = 0,$$

$$\int G(k, i)c_i(w_k^T c_i + b_k - r_i)di + \lambda_1 w_k = 0.$$

For simplicity, we define the following operators:

$$\hat{e} = c \otimes G, \hat{e} = r \otimes G, \hat{F} = (re) \otimes G, \hat{D} = (ae^T) \otimes G, \hat{w} = w \otimes G,$$

where $\otimes$ denotes the standard filter operator. Eqn. 7 thus can be rewritten as:

$$\hat{w}_k = (\lambda_1 I + \hat{D}_k - \hat{e}_k \hat{c}_k^T)^{-1}(\hat{F}_k - \hat{r}_k \hat{c}_k),$$

$$\hat{b}_k = \hat{r}_k - \hat{w}_k^T \hat{c}_k,$$

where $I$ is an identity matrix. Since the computation in Eqn. 8 mainly involves image filtering operations, $\hat{w}_k$ and $\hat{b}_k$ can be computed very effectively.

**Step 2:** We fix $\hat{b}_k, \hat{w}_k$ and take derivative of Eqn. 6 with respect to $r_i$:

$$2 \int G(k, i)(r_i - w_k^T c_i - b_k)dk + 2\lambda_i(r_i - S_i) + \lambda_3 \left. \frac{\partial TV_{21}(r)}{\partial r_i} \right|.$$
Similar to step 1, the first term of Eqn. 9 can also be formulated as image filtering operations as:

\[ 2(r_i - \bar{r}_i^T c_i - \bar{b}_i), \] (10)

The third term of Eqn. 9 is the derivative of Eqn. 4. According to Euler-Lagrange Equations, it can be computed as:

\[ \lambda_3 \frac{\partial TV_2(r)}{\partial r} = -\lambda_3 \left( \frac{r_{xx}(x,y)}{G_1(x)} + \frac{r_{yy}(x,y)}{G_2(y)} \right). \] (11)

To improve the efficiency of the optimization scheme, we introduce an additional labeling matrix \( Q \) which allows us to use the depth image as a whole for computing the gradient for \( r \) as:

\[ \nabla r = 2(r - \bar{w}^T \cdot c - \bar{b}) + 2Q \cdot (r - S) \]
\[- \lambda_3 \left( (r \otimes I_w) \cdot G_1' + (r \otimes I_y) \cdot G_2' \right), \] (12)

where \( \cdot \) denotes the standard element-by-element multiplication, and \( Q \) is the labeling matrix where \( Q(x,y) = \lambda_2 \) if pixel \((x,y)\) is included in \( S \), and \( Q(x,y) = 0 \) otherwise. \( I_w = [1 \cdots 2] \), \( I_y = [1 \cdots 2] \) and the elements in \( G_1 \), \( G_2 \) are the inverse of elements in \( G'_1 \), \( G'_2 \), respectively. This allows us to take the gradient descent method to update the estimated depth map as:

\[ r^{k+1} = r^k - \frac{\rho}{k+1} \nabla r, \] (13)

where \( r^{k+1} \) and \( r^k \) are the restored depth maps obtained at the \( k \)-th and \((k+1)\)-th iterations, respectively. \( \rho \) is the step length. When \( k \) increases, the solution is getting closer to the final solution, thus, we gradually decrease the step length. When \( \|r^{k+1} - r^k\|^2 \) is less than a threshold \( \tau \), the state \( r^{k+1} \) is regarded as the final result.

The complete algorithm for this alternating energy minimization approach is formally described in Algorithm 1. In our implementation, the parameters \( \lambda_1, \lambda_2, \lambda_3, \phi, \rho \) and \( \tau \) are set at 0.001, 1000, 0.8, 2, 2 and 0.001, respectively.

**Algorithm 1: Alternating Energy Minimization**

**Input:** \( r = r_0 \)  
**Output:** \( r_{final} \)

**Step 1:** Fix \( r^k \) and get the \( W, b \) with the Eqn. 8.

**Step 2:** Fix \( W, b \) and obtain \( r^{k+1} \) according to Eqn. 13.

- if \( \|r_{k+1} - r_k\|^2 \leq \tau \)  
  \( r_{final} = r_{k+1} \), exit the iterative process.
- else \( r_k = r_{k+1} \), go to Step 1.

### 3. EXPERIMENTAL RESULTS

We implemented our method using Matlab, and the computational time for a 640 \times 480 depth map is roughly 1 minutes on a Windows machine with an Intel 4-core i5 CPU. To demonstrate the effectiveness of the proposal approach, we first show depth map restoring results of real-world scenes captured by a Kinect device. We then apply it on the Middlebury [13] benchmark dataset with synthetically created depth holes for comparing our method with previous depth inpainting techniques.

#### 3.1. Comparisons on Real Examples

In Fig. 3, we show two depth maps of an indoor office scene captured by a Kinect device, and the restored results by FMM [8], GFMM [9] and our approach. Due to occlusion, both depth maps contain large holes around depth boundaries, and it is crucial to restore sharp edges in these holes. In addition, there are occasional missing pixels in non-occluded regions, as well as strong noise. As we can see for the top example in Fig. 3, the highlighted region, FMM generates over-smoothed edges and introduces ringing artifacts in the recovered depth maps. GFMM restores sharper edges, but fails to recover image details (see the hair region). In contrast, our algorithm restores both sharp depth boundaries and image details, as shown in the highlighted region. For the bottom example, due to the strong noise along the edges on the depth map, both FMM and GFMM do not produce satisfactory results. Thanks to the \( TV_{21} \) constraint, our method is able to obtain a better result.

#### 3.2. Comparisons on the Middlebury Dataset

We also conduct quantitative evaluations on the Middlebury benchmark data set, which has nine examples, each contains a ground truth depth map. We choose two of them to form our test data set: Art and Moebius, which have clutter depth values and are quite appropriate to evaluate performance. For each example we manually create some large missing regions, as shown in Fig. 4. We again apply FMM, GFMM and our method to recover depth values in these regions, and the results are shown in Fig. 4. In the highlighted regions, we can see FMM yields over-smoothed result and GFMM discards some image details, while our method restores both sharp boundaries and small details. The quantitative results in terms of Root Mean Square Error (RMSE) are reported in Table 1, which shows our method generates the lowest RMSE in all cases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Art</th>
<th>Moebius</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMM</td>
<td>9.36</td>
<td>5.29</td>
</tr>
<tr>
<td>GFMM</td>
<td>8.08</td>
<td>4.40</td>
</tr>
<tr>
<td>Our Method</td>
<td>7.13</td>
<td>3.78</td>
</tr>
</tbody>
</table>

**Table 1.** Results of the FMM, GFMM and Ours in terms of RMSE against the ground truth on the Middlebury dataset. The performance of our approach is the best among all the three algorithms.

To further demonstrate that our approach is robust to noise, we conduct another experiment by adding Gaussian noise on the above two depth maps. The Gaussian noise is formulated as follows:

\[ G(r, \rho, \sigma) = \rho \exp(-\frac{r}{2(1+\sigma)^2}), \] (14)

where \( \sigma \) is the standard variation of the depth image and \( \rho \) is the magnitude of the Gaussian noise. The results in terms of RMSE are summarized in Table 2. It suggests that the performance of FMM and GFMM rapidly degrade when the noise level increases, while our method generates the most accurate results in all cases, and the performance degradation is much more subtle compared with other approaches.

### 4. CONCLUSION AND FUTURE WORK

We presented an energy minimization based approach for restoring depth maps generated by Kinect. The energy function is derived from the local linear color-depth regression model, which also includes an
Fig. 3. Depth map restoration using FMM, GFMM and our method. (a) Color image. (b) Input depth map. (c) Missing pixels on the depth map (shown in black). (d) Result of FMM. (e) Result of GFMM. (f) Our result.

Fig. 4. Comparisons with FMM and GFMM on the Middlebury Dataset [13]. (a) Color image. (b) Corrupted depth map. (c) Mask for the corrupted pixels. (d) Result of FMM. (e) Result of GFMM. (f) Our result. (g) Ground truth.

Table 2. Quantitative comparisons on the Middlebury dataset [13] with additional Gaussian noise. The error is measured in term of RMSE for 3 different values of the noise magnitude ρ. It shows that our method generates the most accurate results and is the most robust against noise level increase.

<table>
<thead>
<tr>
<th></th>
<th>Art</th>
<th>Moebius</th>
</tr>
</thead>
<tbody>
<tr>
<td>ρ</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>FMM</td>
<td>9.41</td>
<td>11.29</td>
</tr>
<tr>
<td>GFMM</td>
<td>8.10</td>
<td>9.58</td>
</tr>
<tr>
<td>Ours</td>
<td>7.13</td>
<td>7.42</td>
</tr>
</tbody>
</table>

5. REFERENCES


