

# HIGH RESOLUTION POINT CLOUD GENERATION FROM KINECT AND HD CAMERAS USING GRAPH CUT

Suvam Patra, Brojeshwar Bhowmick, Subhashis Banerjee and Prem Kalra

*Department of Computer Science and Engineering, Indian Institute of Technology Delhi, New Delhi, India*

**Keywords:** Kinect, Resolution Enhancement, Graph Cut, Normalized Cross Correlation, Photo Consistency, VGA, HD.

**Abstract:** This paper describes a methodology for obtaining a high resolution dense point cloud using Kinect (Smisek et al., 2011) and HD cameras. Kinect produces a VGA resolution photograph and a noisy point cloud. But high resolution images of the same scene can easily be obtained using additional HD cameras. We combine the information to generate a high resolution dense point cloud. First, we do a joint calibration of Kinect and the HD cameras using traditional epipolar geometry (Hartley and Zisserman, 2004). Then we use the sparse point cloud obtained from Kinect and the high resolution information from the HD cameras to produce a dense point cloud in a registered frame using graph cut optimization. Experimental results show that this approach can significantly enhance the resolution of the Kinect point cloud.

## 1 INTRODUCTION

Nowadays, many applications in computer vision are centred around generation of a complete 3D model of an object or a scene from depth scans or images. This traditionally required capturing images of the scene from multiple views to generate a model of the scene. However, today with the advent of affordable range scanners, reconstruction of scenes from multi-modal data which include image as well as depth scans of objects and scenes help in more accurate modelling of 3D scenes.

There has been considerable work with time-of-flight (ToF) cameras which capture depth scans of the scene by measuring the travel time of an emitted IR wave from the device reflected back from the object (Schuon et al., 2008). Recently, a much cheaper range sensor has been introduced by Microsoft called the Kinect (Smisek et al., 2011) which has an inbuilt camera, an IR emitter and a receiver. The emitter projects a predetermined pattern whose reflection off the object provides the depth cues for 3D reconstruction. Though Kinect produces range data only in VGA resolution, this data can be very useful as an initial estimate for subsequent resolution enhancement. There have been several approaches to enhance the resolution of a point cloud obtained from range scanners or ToF cameras, using interpolation or graph based techniques (Schuon et al., 2009; Schuon et al., 2008). Diebel

et.al. (Diebel and Thrun, 2006) used a MRF based approach whose basic assumption is that depth discontinuities in scene often co-occur with intensity or brightness changes in the scene, or in other words regions of similar intensity in a neighbourhood have similar depth. Yang et.al. (Qingxiong Yang and Nistr, 2007) make the same assumption and use a bilateral filter to enhance the resolution in depth. However, the assumption is not universally true and may result in over smoothing of the solution.

Sebastian et. al. (Schuon et al., 2009; Schuon et al., 2008), use a super-resolution algorithm on low resolution LIDAR ToF cameras and they rely on the depth data for detecting depth discontinuities instead of relying on regions of image smoothness.

In this paper we propose an algorithm for depth super-resolution using additional information from multiple images obtained through HD cameras. We register the VGA resolution point cloud obtained from Kinect with what can be obtained from the HD cameras using multiple views geometry and carry out a dense 3D reconstruction in the registered frame using two basic criteria: i) photo-consistency (Kutulakos and Seitz, 1999) and ii) rough agreement with Kinect. The reconstructed point cloud is at least ten times denser in comparison to the initial point cloud. In this process we also fill up the holes of the initial Kinect point cloud.

## 2 PROPOSED METHODOLOGY

### 2.1 Camera Calibration

We determine the camera internal calibration matrices (Hartley and Zisserman, 2004) for the Kinect VGA camera and all the HD cameras offline using a state of the art camera calibration technique (Zhang, 2000). Henceforth we assume that all the internal camera calibration matrices are known and define the  $3 \times 4$  camera projection matrix for the Kinect VGA camera as

$$\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}] \quad (1)$$

where  $\mathbf{K}$  is the camera internal calibration matrix of the Kinect VGA camera. In other words, Kinect is our world origin.

We use ASIFT (Morel and Yu, 2009) to obtain image point correspondences and for every HD camera we compute the extrinsic camera parameters using standard epipolar geometry (Hartley and Zisserman, 2004). For each HD camera we first carry out a robust estimation of the fundamental matrix (Hartley and Zisserman, 2004). Given a set of image point correspondences  $\mathbf{x}$  and  $\mathbf{x}'$ , the fundamental matrix  $\mathbf{F}$  is given by:

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (2)$$

and can be computed using eight point correspondence.

Once, the Fundamental Matrix is known, we can estimate the external calibration from essential matrix  $\mathbf{E}$ , derived from Fundamental matrix using the equation as in (Hartley and Zisserman, 2004)

$\mathbf{E} = \mathbf{K}'^T \mathbf{F} \mathbf{K} = [\mathbf{t}]_{\times} \mathbf{R} = \mathbf{R} [\mathbf{R}^T \mathbf{t}]_{\times}$  where,  $\mathbf{K}'$  is the internal calibration matrix of the HD camera. As this essential matrix has four possible decompositions, we can select one of them using the cheirality check (Hartley and Zisserman, 2004) on Kinect point cloud.

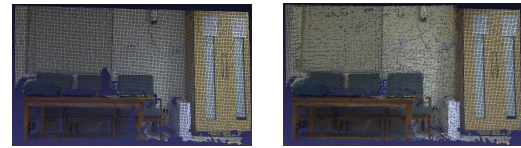
The projection matrix of the HD camera in the Kinect reference frame is then given as

$$\mathbf{P}' = \mathbf{K}'[\mathbf{R}|\mathbf{t}] \quad (3)$$

### 2.2 Generation of High Resolution Point Cloud

Normalized cross correlation(NCC) method, which tries to find point correspondences in an image pair by computing statistical correlation between the window centred at the candidate point, is an inadequate tool for finding dense point correspondences. Projecting the sparse Kinect point cloud on to an HD image leaves most pixels without depth labels, and one can attempt to establish correspondence for these pixels using normalized cross correlation along rectified

epipolar lines. Once the correspondence is found we can obtain the 3D point for this correspondent pair using stereo triangulation technique. In figure 1 we show a result obtained using NCC. The reconstruction has many holes due to ambiguous cross correlation results and incorrect depth labels.



(a) Initial Kinect point cloud. (b) High resolution point cloud generated by NCC.

Figure 1: Resolution enhancement using NCC.

The voxel labelling problem can be represented as one of minimizing an energy function of the form

$$E(L) = \sum_{p \in \mathcal{P}} D_p(L_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(L_p, L_q) \quad (4)$$

where  $\mathcal{P}$  is the set of voxels to be labelled,  $L = \{L_p | p \in \mathcal{P}\}$  is a 0-1 labeling of the voxel  $p$ ,  $D_p(\cdot)$  is data term measuring the consistency of the label assignment with the available data,  $\mathcal{N}$  defines a neighbourhood system for the voxel space and each  $V_{p,q}(\cdot)$  is a smoothness term that measures the consistency of labelling at neighbouring voxels.

When the above energy minimization problem is represented in graphical form (Boykov and Kolmogorov, 2004), we get a two terminal graph with one source and one sink nodes representing the two possible labels for each voxel (see figure 2). Each voxel is represented as a node in the graph and each node is connected to both source and sink nodes with edge weights defined according to the data term of the energy function. In addition, the voxel nodes are also connected to each other with edges, with edge strengths defined according to the neighbourhood interaction term. A minimum cut through this graph gives us a minimum energy of the configuration.

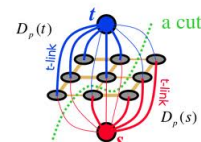


Figure 2: A two terminal graph from (Boykov and Kolmogorov, 2004).

### 2.2.1 Assigning Cost to the Data Term

Photo consistency (Kutulakos and Seitz, 1999) is one of the most frequently used measures for inter image consistency. However, in real situations, several voxels in a close neighbourhood in depth satisfy the photo consistency constraint resulting in a “thick” surface as demonstrated in top view in figure 3. In view of this, we use closeness to initial Kinect data as an additional measure to resolve this problem of thickness in the output high resolution point cloud.

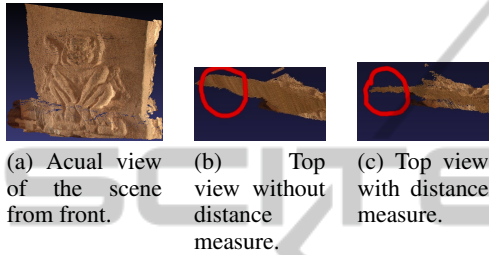


Figure 3: Comparison between resolution enhancement without and with distance measure.

We define the data term based on the following two criteria: i) Adaptive photo consistency measure for each voxel. ii) Distance of each voxel from its nearest approximate surface.

We use the photo consistency measure suggested by Slabaugh et. al. (Slabaugh and Schafer, 2003). We project each voxel  $i$  on to the  $N$  HD images and calculate the following two measures:

1.  $S(i)$ , the standard deviation of the intensity values in the projection neighbourhoods calculated over all  $N$  images.
2.  $\bar{s}(i)$ , the average of the standard deviation in the projection neighbourhoods for each image projection.

The voxel  $i$  is photo consistent over the  $N$  images if the following condition is satisfied

$$S(i) < \tau_1 + \tau_2 * \bar{s}(i) \quad (5)$$

where  $\tau_1$  and  $\tau_2$  are global and local thresholds to be suitably defined depending on the scene. The overall threshold specified by the the right hand side of the above inequality changes adaptively for each voxel. For each voxel we assign a weight  $D_{photo}(\cdot)$  for the terminal edges in the graph based on this threshold.

$$D_{photo}(i) = photocost * \exp\left(-\frac{S(i)}{\tau_1 + \tau_2 * \bar{s}(i)}\right) \quad (6)$$

with the source and

$$D_{photo}(i) = photocost * \left(1 - \exp\left(-\frac{S(i)}{\tau_1 + \tau_2 * \bar{s}(i)}\right)\right) \quad (7)$$

with the sink

where,  $S(i)$  and  $\tau_1 + \tau_2 * \bar{s}(i)$  is the standard deviation and the adaptive threshold respectively for the  $i^{th}$  voxel and  $photocost$  is a scale factor. Here the expression inside the exponential gives the normalized standard deviation of  $i^{th}$  voxel.

As a pre-processing step before applying graph cut, we create an approximate surface (Alexa and Behr, 2003) for each non-Kinect voxel using the Kinect voxels in its neighbourhood  $\mathcal{N}_{\mathcal{K}}$  of size  $K \times K \times K$ . We pre-process the Kinect point cloud to generate an approximate surface for each non-Kinect voxel in our voxel space in the following way:

We consider  $S_p$  as the surface that can be constructed with the voxels  $P = \{p_i\}$  captured by the Kinect. Then, as suggested in (Alexa and Behr, 2003), we try to replace  $S_p$  with an approximate surface  $S_r$  with reduced set of voxels  $R = \{r_i\}$ . This is done in two steps: A local reference plane  $H = \{x | \langle n, x \rangle - D = 0, x \in \mathbb{R}^3, n \in \mathbb{R}^3, \|n\| = 1\}$  is constructed using the moving least squares fit on the the point  $p_i$  under consideration. The weights for each  $p_i$  is a function of the distance from the projected current voxel on to the plane. So,  $H$  can be determined by locally minimizing

$$\sum_{i=1}^N (\langle n, p_i \rangle - D)^2 \theta(\|p_i - q\|) \quad (8)$$

where  $\theta$  is a smooth monotonically decreasing function,  $q$  is the projected point on the plane corresponding to the voxel  $r$ ,  $n$  is the normal and  $D$  is the perpendicular distance from the origin of the plane. Assuming  $q = r + tn$  with  $t$  as a scale parameter along the normal, equation(8) can be rewritten as

$$\sum_{i=1}^N (\langle n, p_i - r - tn \rangle)^2 \theta(\|p_i - r - tn\|) \quad (9)$$

Let  $q_i$  be the projection of  $p_i$  on  $H$  and  $f_i$  be the height of  $p_i$  over  $H$ . We can find the surface estimate  $Z = g(X, Y)$  by minimizing the least squares equation given by:

$$\sum_{i=1}^N (g(x_i, y_i) - f_i)^2 \theta(\|p_i - q\|) \quad (10)$$

where  $x_i$  and  $y_i$  are the  $x$  and  $y$  values corresponding to the  $i^{th}$  voxel and  $\theta$  is a smooth monotonically decreasing function which is defined as:

$$\theta(d) = e^{-\frac{d^2}{h^2}} \quad (11)$$

where,  $h$  is the fixed parameter which depicts the spacing between neighbouring voxels. It reflects the smoothness in the surface. For our experiment we have taken a fourth order polynomial fitting.

This surface is locally smooth and usually lacks geometric details, but provides a good measure for the approximate depth of the surface.

Hence, the second cost that we include in the data term is based on the distance of the current voxel from the pre-computed surface that fits that voxel. So, we project each of the non-Kinect voxel on to the pre-computed surface (Alexa and Behr, 2003). Ideally if the voxel is on the surface then the difference between its actual coordinates and projected coordinates should be small, which encourage us to use this measure in the data term. Accordingly, we assign a cost to  $D_p$  on the basis of the euclidean distance between its actual coordinates and projected coordinates on the approximate surface.

$$D_{dist}(i) = \frac{P(r_i) - r_i}{dist\_threshold} \quad (12)$$

with the source and

$$D'_{dist}(i) = 1 - D_{dist}(i) \quad (13)$$

with the sink. Here, the threshold  $dist\_threshold$  is experimentally determined on the basis of the scene under consideration. The total cost is expressed as:

$$D_p(i) = D_{dist}(i) * D_{photo}(i) \quad (14)$$

Table 1: Assignment of  $D_p$ .

$D_p(i)$	Type of Voxel
$\infty$ with source and 0 with sink	Kinect voxel
Based on equation(6,7,12,13,14)	Non-Kinect voxel

The cost  $D_p(\cdot)$  is assigned to a Kinect voxel so that it is turned “ALWAYS ON”. After that, for each non-Kinect voxel first a distance check is done followed by a photo consistency check over all the  $N$  HD images. Then accordingly a cumulative cost is assigned based on the equations above.

### 2.2.2 Assigning Cost to the Smoothness Term

We have assigned a constant smoothness cost to the edges between each of the voxels and its neighbourhood  $\mathcal{N}$ . Here, we have taken  $\mathcal{N}$  to be the 6-neighbourhood of each voxel.

Smoothness cost is assigned according to the Potts model(Kolmogorov and Zabih, 2004; Boykov et al., 2001). We can represent  $V_{p,q}$  as

$$V_{p,q}(f_p, f_q) = U_{p,q} \cdot \delta(f_p \neq f_q) \quad (15)$$

Here, we have taken  $V_{p,q}$  from Potts model as in Table 2. After assigning the data and smoothness costs to the graph edges, we run the min-cut on this graph.

Table 2: Assignment of  $V_{p,q}$  based on Potts model.

$V_{p,q}(f_p, f_q)$	Condition
0	$f_p = f_q$ (Both are Kinect voxels)
100	Otherwise

## 3 RESULTS

We provide experimental results on both indoor and outdoor scenes. For capturing the HD images we have used the SONY HVR-Z5P camera which has an image resolution of  $1440 \times 1080$ . This camera was placed at multiple positions to capture images of the same scene from different viewpoints. The experimental set-up for capturing a typical scene by one Kinect and three HD cameras has been depicted in figure 4.

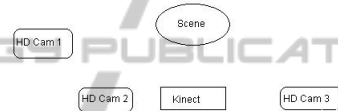
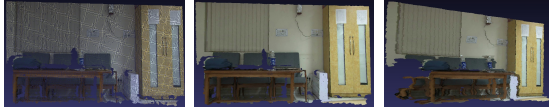


Figure 4: Our experimental set-up for capturing a typical scene.

We have used a Dell Alienware i7 machine with 6GB RAM support for producing the results. In our case the number of voxels that we take for the scene depends largely on the amount of physical memory of the machine. The figure 5 shows the resolution enhancement of an indoor scene done using one Kinect and two HD cameras. Figure 5b shows the high resolution point cloud generated with our method. In this all the holes have been filled up in contrast to the point cloud generated using NCC based method as shown in figure 1b. There are almost no outlier points. Here we have used  $300 \times 300 \times 100$  voxels and the value of  $\tau_1 = 60$  and  $\tau_2 = 0.5$ . Figure 6 shows the result of resolution enhancement on an outdoor scene in the archaeological site of Hampi using one Kinect and two HD cameras. The point cloud is at least 10 times denser than the initial point cloud. The value of  $\tau_1$  and  $\tau_2$  were chosen to be 80 and 0.5 respectively. Figure 7 also shows the resolution enhancement on another sculpture at Hampi using one Kinect and two HD cameras. The values of  $\tau_1$  and  $\tau_2$  were similar to figure 6.

Figure 8 shows the resolution enhancement of a toy model where the surface is not smooth. This experiment was performed using one Kinect and three HD cameras. We have shown the dense point cloud corresponding to both the low resolution scene as well

as the high resolution scene and finally overlapped their coloured depth map to show that the geometry is not distorted in any way. In order to do a quantitative evaluation of our methods we have adopted two approaches.



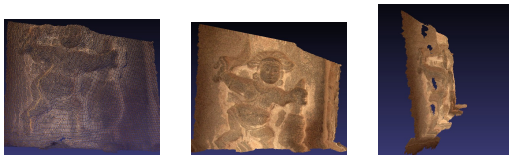
(a) Initial point cloud. (b) High resolution point cloud. (c) Side view.

Figure 5: Indoor scene- A typical room. (a) Initial low resolution point cloud from Kinect, (b) and (c) front and side view of the high resolution point cloud generated by our method with  $\tau_1 = 80$  and  $\tau_2 = 0.5$ .



(a) Initial point cloud. (b) High resolution point cloud. (c) Side view.

Figure 6: Archaeological scene1- A sculpture depicting a monkey on a pillar. (a) Initial low resolution point cloud from Kinect, (b) and (c) front and side view of the high resolution point cloud generated by our method with  $\tau_1 = 60$  and  $\tau_2 = 0.5$ .



(a) Initial point cloud. (b) High resolution point cloud. (c) Side view.

Figure 7: Archaeological scene2- A sculpture depicting a goddess on a pillar. (a) Initial low resolution point cloud from Kinect, (b) and (c) front and side view of the high resolution point cloud generated by our method with  $\tau_1 = 60$  and  $\tau_2 = 0.5$ .

### 3.1 Verification through Projection on Another Camera

In order to demonstrate the efficiency of our method we have computed the projection matrix of a different camera which is seeing the same scene as of figure 6, little displaced from the original cameras used for resolution enhancement and whose external calibration matrix  $[\mathbf{R}|\mathbf{t}]$  is known beforehand. We have used this projection matrix to project the HD point cloud onto



(a) Initial point cloud. (b) High resolution point cloud. (c) Two depth maps overlapped.

Figure 8: Indoor Scene- A model of a dog. (a) Initial low resolution point cloud from Kinect, (b) front view of the high resolution point cloud generated by our method with  $\tau_1 = 70$  and  $\tau_2 = 0.5$ , (c) blue HD depth map overlapped with red low resolution depth map showing that the geometry is preserved.



(a) Original image. (b) Projected image. (c) Difference image.

Figure 9: Verification through projection on another camera for the scene in figure 6. The difference image in which around 90% is black, shows that the geometry is preserved.

a 2D image and have taken the difference between the projected image and the ground truth. The difference image in figure 9c is around 90% black showing that the HD point cloud generated by our method was geometrically accurate.

### 3.2 Verification through Interpolation and Comparison

In order to show that the depth map of the HD point cloud generated by our method conforms to the point cloud generated by Kinect, we generated an interpolated point cloud for the initial point cloud of figure 6 by fitting an MLS surface of order four through it. In order to quantify that our result show better depth variations than the interpolated point cloud, we took a part of each of the point clouds generated by the interpolation method and our method; and compared them with that of the Kinect point cloud. The standard deviation of the depth variations in the selected part of the point cloud generated by interpolation was 0.010068 whereas the same by our method was 0.021989, which is much closer to the standard deviation generated by original point cloud i.e. 0.024674.

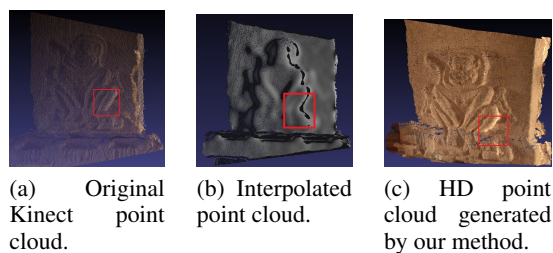


Figure 10: Verification through interpolation and comparison. The area selected by the red rectangle shows the part selected for quantitative estimation of the depth variations.

## 4 CONCLUSIONS

We have presented a methodology which combines HD resolution images with the low resolution Kinect to produce high-resolution dense point cloud using graph cut. Firstly, Kinect and HD cameras are registered to transfer Kinect point cloud to the HD camera for obtaining high resolution point cloud space. Then, we discretize the point cloud in voxel space and formulate a graph cut formulation which take care of the neighbor smoothness factor. This methodology produces good high resolution image with the help of low resolution Kinect point cloud which could be useful in building high resolution model using Kinect.

## ACKNOWLEDGEMENTS

The authors gratefully acknowledge Dr. Subodh Kumar, Neeraj Kulkarni, Kinshuk Sarabhai and Shruti Agarwal for their constant help in providing several tools for Kinect data acquisition, module and error notification respectively.

Authors also acknowledge Department of Science and Technology, India for sponsoring the project on “Acquisition, representation, processing and display of digital heritage sites” with number “RP02362” under the India Digital Heritage programme which helped us in acquiring the images at Hampi in Karnataka, India.

## REFERENCES

Alexa, M. and Behr, J. (2003). Computing and rendering point set surfaces. In *IEEE Transactions on Visualization and Computer Graphics*.  
 Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for en-

ergy minimization in vision. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, pages 1124–1137.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pages 1222–1239.

Diebel, J. and Thrun, S. (2006). An application of markov random fields to range sensing. in advances in neural information processing. In *Advances in Neural Information Processing Systems*, page 291–298.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, 2nd edition.

Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized via graph cuts? In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 147–159.

Kutulakos, K. and Seitz, S. (1999). A theory of shape by space carving. In *7th IEEE International Conference on Computer Vision*, volume 1, page 307–314.

Morel, J.-M. and Yu, G. (2009). Asift: A new framework for fully affine invariant image comparison. In *SIAM Journal on Imaging Sciences*. Volume 2 Issue 2.

Qingxiong Yang, Ruigang Yang, J. D. and Nistr, D. (2007). Spatial-depth super resolution for range images. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Schuon, S., Theobalt, C., Davis, J., and Thrun, S. (2008). High-quality scanning using time-of-flight depth super-resolution. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*.

Schuon, S., Theobalt, C., Davis, J., and Thrun, S. (2009). Lidarboost depth super-resolution for tof 3d shape scanning. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Slabaugh, G. and Schafer, R. (2003). Methods for volumetric reconstruction of visual scenes. In *IJCV 2003*.

Smisek, J., Jancosek, M., and Pajdla, T. (2011). 3d with kinect. In *IEEE Workshop on Consumer Depth Cameras for Computer Vision*.

Zhang, Z. (2000). A flexible new technique for camera calibration. In *IEEE Transactions On Pattern Analysis And Machine Intelligence*, VOL. 22, NO. 11.