Conversion of 2D Images to 3D Using Data Mining Algorithm

Zeal Ganatra  
Student  
D. J. Sanghvi College of Engineering  
Mumbai.

Riddhi Chavda  
Student  
D. J. Sanghvi College of Engineering  
Mumbai.

Prof. Lynette D’mello  
Asst. Professor  
D. J. Sanghvi College of Engineering  
Mumbai.

ABSTRACT
In this paper, we adopt the radically different approach to develop a simplified and computationally-efficient version of our recent 2D-to-3D image conversion algorithm. We consider two major techniques: calculating the depth using the edge information and second, depth map evaluation by kNN based learning. Using examples and with parameters, we compare these techniques to find out the optimum approach for conversion of 2D images to 3D images by introducing depth in the latter. While far from perfect, the presented results demonstrate that online repositories of 3D content can be used for effective 2D to 3D image conversion.

Keywords
2D images, 3D images, kNN Search, depth map, image processing, data mining.

INTRODUCTION
The advent of innovative 3D technology and accruing sales of 3D consumer electronics, has accompanied an increase in demands of more and more 3D technology. This has led to wide interest in conversion of the already existing two-dimensional (2D) contents to three-dimensional (3D) contents in the field of image processing. This is a great issue in emerging 3D applications because the conventional 2D content do not provide the depth information which is required for the 3D displays. So 3D displays enhance visual quality more than two-dimensional (2D) displays.

The 2D to 3D conversion adds the binocular disparity depth indication or cue to the digital images perceived by the human brain. Therefore, if it is done appropriately, it significantly improves the immersive effect while one is viewing this stereo video in comparison to the original 2D video. However, in order to be successful, this conversion needs to be done with sufficient accuracy and correctness, that is, the quality of the original 2D images should not deteriorate, and also the introduced disparity cue should not contradict to the other cues used for depth perception by the human brain. If this is done properly and thoroughly, the conversion produces stereo video of similar quality to “native” stereo video which is shot in stereo and accurately aligned and adjusted in post-production.

It has been found that the manual conversion of 2D – 3D image have been the most effective but then again are costly in terms of time. Many automatic conversion techniques have thus been proposed but each of these techniques consider assumptions that are not met in the real world.

The main difference between 2D and 3D images is clearly the presence of depth in 3D images which makes calculation of depth the most important factor during conversion of images from 2D to 3D. Several methods have been proposed for the same. Out of these we shall study mainly two methods. First, calculating the depth using the edge information of the existing 2D image and second, depth map evaluation by kNN based learning from training set of images. To compare these two methods, we use the generation of a depth map. A depth map is an image or image channel that contains information relating to the distance of the surfaces of scene objects from a viewpoint.

The above mentioned comparative study concludes the better of the methods to convert a 2D image to 3D is kNN based method. Hence in this paper, we will discuss both these methods and show this would result in 3D content being easily available from the existing 2D content. Imparting depth information to the existing pixels of the 2D image can give a better detail of the image. Hence, it should be used to generate various levels of information, like creating a 3D model of a region/space based on the existing 2D repository of images from the region for depth. Thus we would arrive at a conclusion that the kNN based method of depth evaluation is optimum.

DEPTH EVALUATION USING EDGE INFORMATION IN THE GIVEN 2D IMAGE
Depth generation algorithms for 2D to 3D conversions face two major challenges. First is the depth uniformity inside the same object in the image. Because the image consists of 2D pixel arrays, information about the object grouping relation of pixels is lacking. A better grouping of pixels implies a better outcome for the depth uniformity inside the object. An effective grouping method should consider both color similarity and spatial distance. The other challenge involves figuring out an appropriate depth relationship among all the objects in an image.

To overcome these two challenges, we present a novel algorithm that uses a simple depth hypothesis to assign the depth of each group rather than retrieving the depth value directly from the depth cue. Firstly, an effective grouping
method is chosen which involves grouping pixels that have similar colours and spatial locality. Now the depth values are assigned according to the hypothesis depth value. To enhance the visual comfort, we apply a cross bilateral filter. The following steps specify the process step-by-step:

**Step 1: Block-Based Region Grouping**

Figure 1 shows the flow of block grouping using edge information. The computational complexity is reduced by using a block-based algorithm. This algorithm considers that each pixel in the same block has the same depth value. For example, consider a 4x4 pixel block, and each node is four-connected. The value of each link connecting thee nodes is the absolute difference of the mean of neighbouring blocks:

\[ \text{Diff}(a, b) = | \text{Mean}(a) - \text{Mean}(b) | \]

Where ‘a’ and ‘b’ denote two neighbouring blocks, respectively, and Mean(a) and Mean(b) represents the mean colours of ‘a’ and ‘b’ respectively. This value gives measure of the similarity strength of neighbouring blocks. A smaller value implies a higher similarity between the two blocks [2].

![Figure 1: Flow of block based region Grouping](image)

**Step 2: Depth from Prior Hypothesis**

Now that we have generated the blocked groups, the corresponding depth for each block is assigned by the hypothesized depth gradient. The depth value of a given block group \( R \) is assigned by [2].

\[
\text{Depth}(R) = 128 + \frac{255 \cdot \left( \sum_{\text{pixel}(x,y) \in R} W_{rl} \cdot \left( x - \frac{\text{width}}{2} \right) + W_{ud} \cdot \left( y - \frac{\text{height}}{2} \right) \right)}{\text{pixel_num}(R)}
\]

where \(|W_{rl}| + |W_{ud}| = 1\)

A larger value of the assigned depth implies that the pixel represents a part closer to the user. The above equation suggests that the assigned depth value indicates the gravity centre of the block group, thus explaining why each block group belongs to the same depth. The absolute value and sign of \( W_{rl} \) and \( W_{ud} \) can adjust the left-to-right and top-to-bottom depth gradient weight [2].

**Step 3: 3D Image Visualization using Bilateral Filtering and Depth Image-Based Rendering**

The depth map generated by block-based region grouping contains undesirable blocky artifacts. Here, the blocky artifacts are removed by using the cross bilateral filter, as expressed in the following equation:

\[
\text{Depth}_f(x_i) = \frac{1}{N(x_i)} \sum_{x_j \in \Omega(x_i)} e^{-a \left( \frac{||x_i - x_j||^2}{\sigma^2} + \frac{||\text{Grad}(x_i) - \text{Grad}(x_j)||^2}{\sigma_f^2} \right)} \text{Depth}(x_i),
\]

\[
N(x_i) = \sum_{x_j \in \Omega(x_i)} e^{-a \left( \frac{||x_i - x_j||^2}{\sigma^2} + \frac{||\text{Grad}(x_i) - \text{Grad}(x_j)||^2}{\sigma_f^2} \right)}
\]

The cross bilateral filter smoothens the depth map properly while preserving the object boundaries. The blocky artifacts in the generated depth map are effectively removed while the sharp depth discontinuities along the object boundary are preserved [2].

**2D-TO-3D IMAGE CONVERSION BY LEARNING DEPTH FROM EXAMPLES**

Recently, mining techniques based on image parsing have been used for estimating the depth map from single
monocular images. Such methods can generate depth maps for any 2D visual material, but currently work on only few types of images using carefully selected training data is done.

The proposed approach is a simplified algorithm that "mines" the scene depth from a large repository of image and depth pairs and is computationally more efficient than the other algorithms.

Figure 3 shows the block diagram of our approach. The sections below provide a description of each step and some high-level mathematical detail. In these sections, Q is the query image for which a right image QR is being sought. We assume that a database I = {(I1, d1), (I2, d2), ...} of image + depth pairs (I, d) is available. Note that a database of stereoscopic videos, such as YouTube 3D, could be processed to extract image + depth pairs. The goal is to find a depth estimate ^d and then a right-image estimate ^QR given the 3D database I.

Step 1: kNN Search

There are two types of images in a 3D image repository: those which are relevant for determining depth from a 2D query image, and those which are irrelevant. Images that are not photo metrically similar to the 2D query are rejected as they are not useful for estimating depth as per our assumption. Choosing a smaller subset of images gives an added advantage practically of computational tractability when the dictionary size is very large [1].

Our 2D query image Q is the left image from a stereo pair whose right image QR is unknown. We assume that a database of 3D images or videos I, such as the NYU depth database or YouTube 3D, is available, and that for each RGB image Ii in the database the corresponding depth field di is either known or can be computed from a stereo pair [1].

One method for selecting a useful subset of depth relevant images from a large image dictionary is to select only the 'k' images that are closest to the input where closeness is measured by some distance function which captures global image properties such as color, texture, edges, etc. For the distance function, we use the Euclidean norm of the difference between histograms of oriented gradients (HOGs) computed from two images. Each HOG comprises of 144 real values (4x4 blocks with 9 gradient direction bins) which can be efficiently computed. This image closeness measure is significantly less complex computationally as compared to the weighted Hamming distance between the binary hashes of features used originally [1].

We perform a search for the top matches to our 2D query among all the 3D images in the database I, which search returns an ordered list of image + depth pairs, from the most to the least photo metrically similar ones to the 2D query. We discard all matches except the top 'k' ones (kNNs) from this list.

Step 2: Depth Fusion

None of the NN image + depth pairs (Ii, di), i ∈ K may match a query Q accurately. If a similar object (e.g., table) appears at a similar location in several KNN images, then such an object can also appear in the query and the depth field being sought should reflect this. This depth field is computed by applying the median operator across the kNN depths at each spatial location x as follows:

d[x] = median{di[x], ∀i ∈ K}.

Examples of the fused depth fields d are shown in the central column of Figure 4. Although these depths are overly smooth, they provide a globally-correct, although coarse, assignment of distances to various areas of the scene.

Step 3: Cross Bilateral Filtering

While the median-based fusion helps make depth more consistent globally, the fused depth is overly smooth and locally inconsistent with the query image due to the following reasons:

1. Misalignment of edges between the fused depth field and query image,
2. Lack of fused depth edges where sharp object boundaries occur,
3. Lack of fused depth smoothness where smooth depth changes are expected.

We apply bilateral filtering to the fused depth with two goals: alignment of the depth edges with those of the query image Q and local noise/granularity suppression in the fused depth d. This is implemented as follows:
\[ d[x] = \frac{1}{y[x]} \sum_y d[y] h_{\sigma_e}(x - y) h_{\sigma_e}(Q[x] - Q[y]), \]
\[ y[x] = \sum_y h_{\sigma_e}(x - y) h_{\sigma_e}(Q[x] - Q[y]), \]

Where \( d \) is the filtered depth field and
\[ h_{\sigma_e}(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \]
is a Gaussian weighting function. Note that the directional smoothing of \( d \) is controlled by the query image via the weight \( h_{\sigma_e}(Q[x] - Q[y]) \). For large discontinuities in \( Q \), the weight \( h_{\sigma_e}(Q[x] - Q[y]) \) is small and thus the contribution of \( d[y] \) to the output is small. However, when \( Q[y] \) is similar to \( Q[x] \) then \( h_{\sigma_e}(Q[x] - Q[y]) \) is relatively large and the contribution of \( d[y] \) to the output is larger. In essence, depth filtering (smoothing) is happening along (and not across) query edges [1].

The filtered depth preserves the global properties captured by the unfiltered depth field \( d \), and is smooth within objects and in the background. At the same time it keeps edges sharp and aligned with the query image structure [1].

Step 4: Stereo Rendering
In order to generate an estimate of the right image \( QR \) from the 2D query \( Q \), we need to compute the disparity \( \delta \) from the estimated depth \( d \). Assuming that the fictitious image pair \((Q, QR)\) was captured by parallel cameras with baseline \( B \) and focal length \( f \), the disparity is simply \( \delta(x, y) = B f / d(x) \), where \( x = (x, y)^T \). We forward-project the 2D query \( Q \) to produce the right image:
\[ Q_{\delta}[x + \delta(x, y), y] = Q[x, y] \]
while rounding the location coordinates \((x + \delta(x, y), y)\) to the nearest sampling grid point. We handle occlusions by depth ordering: if \((x_i + \delta(x_i, y_i), y_i) = (x_j + \delta(x_j, y_j), y_j)\) for some \( i, j \), we assign to the location \((x_i + \delta(x_i, y_i), y_i)\) in \( QR \) an RGB value from that location \((x_i, y_i)\) in \( Q \) whose disparity \( \delta(x_i, y_i) \) is the largest. In newly-exposed areas, i.e., for \( x_j \) such that no \( x_i \) satisfies \((x_j, y_j) = (x_i + \delta(x_i, y_i), y_i)\), we apply simple inpainting using inpaint nans from MatlabCentral [1].

**COMAPPATIVE STUDY**

The two methods of converting 2D images to 3D image have been discussed in the report. The first method generates depth map for 3D image from edge information and the other method which uses kNN based learning approach are compared. The different parameters judged from the results that are compared here are as follows:

1. Quality:
   In edge detection technique, we get a 3D image which is blocky. Considering, more the number of blocks more is the average runtime, we may need to compromise on quality.
   On the other hand, kNN based approach returns image with highly defined edges. This image is very less distorted as compared to the one obtained from the edge detection based technique.

2. Processing time:
   Both techniques use different methods for validating their result. The kNN based approach uses LOOCV based validation to compare the image to that of Make3D algorithm using their co-variance factors.
   The edge detection based technique varies block size against the average runtime to validate its result. Time taken to generate the output image depends on the block size, whereas in the kNN based approach it does not depend on any factors. Thus average runtime in the kNN approach is very less of merely 5 seconds as compared to the varying results in edge based method. The minimum time required in the edge detection technique is 500 milliseconds [2].

3. Parallax:
   Parallax is a displacement or difference in the apparent position of an object viewed along two different lines of sight, and is measured by the angle or semi-angle of inclination between those two lines.
   The parallax in case of edge detection technique is very large compared to the negligible value in kNN approach. This results in loss of image information for the 3D view.

**CONCLUSION AND FUTURE WORK**

Depth evaluation by learning from examples proposes a simplified data-driven 2D-to-3D conversion method and has objectively validated its performance against state-of-the-art Make3D algorithm. This algorithm compares favourably in terms of estimated depth quality as well as computational complexity. Admittedly, the validation was limited to a database of indoor scenes captured by Kinect camera. The generated anaglyph images produce a comfortable 3D perception but they are also not completely void of distortions. With the continuously accruing amount of 3D data online and rapidly growing computing power in the cloud, the proposed algorithm looks like a promising alternative to an operator assisted 2D to 3D conversion.

Depth evaluation using edge information has presented a novel 2D to 3D conversion algorithm. The proposed algorithm utilizes edge information to group the image into coherent regions. A simple depth hypothesis is adopted to assign the depth for each region and a cross bilateral filter is consequently applied to remove the blocky artifacts. This algorithm is quality-scalable depending on the block size. Smaller block size results in better depth detail whereas larger block size has lower computational complexity. Capable of generating a comfortable 3D effect, the proposed algorithm is highly promising for 2D-to-3D conversion in 3D applications.

The proposed approach when clubbed with efficient hardware and real-time systems can be used to convert 2D content to 3D in real-time and hence can be used in 3D modelling of otherwise 2D images captured from a normal camera.
REFERENCES


