Volume 5, Issue 8, August 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Semi-Supervised 2D to 3D Conversion

Kochurani Sabu, Dona Jose
CSE Department, VJCET, MG University
India

Abstract— The three-dimensional (3D) processing or 3D display plays a great role in the image or visual processing field. But the lack of 3D content is becoming an important problem for the 3D development. The 3D displays provide more realism in visual quality than the 2D displays do. The depth information for the existed 2D contents are not recorded and it is naturally lost when we are taken an image or video using the 2D camera. To solve the 2D to 3D conversion problem first we must extract the depth information from the existing 2Dimage or video and by using some methods, automatically convert it to 3D. Here a new 2D to 3D conversion method is proposed which is based on two existing segmentation algorithms namely, Seed growing based Graph cut and Random Walk. The Graph cut and Random walk combination produces a result that is better than the result these obtained separately. Also this requires user−defined seed points on the objects and regions in the image and uses this seed information to determine the rest of the depths in the image. After, the rest of the depths are determined, creating depth maps to generate the 3D content. Also region filling is used here to avoid the holes present in the depth map. And this work was to adaptively combine the Graph Cuts and Random Walks algorithms in order to produce good quality depth maps. Then the final depth map is used for Depth Image Based Rendering to generate the 3D views.

Keywords— Depth maps, graph cuts, random walks, semi-automatic, 2D to 3D conversion

I. INTRODUCTION

Stereoscopy, sometimes called stereoscopic imaging, is a technique used to enable a three-dimensional effect. Here depth illusion is added to the flat image. Creating 3D display plays a great role in the image or visual processing field and it is from the existing 2D content. The 2D to 3D conversion gives a sense of realism in the 3D image by adding the artificial depth to the existing 2D content. However, the current accepted method for high quality conversion is a manually labour intensive process, commonly known as rotoscoping. Specifically, two novel views must be generated for each single frame or image, using information from the frame, or some frames before or after the current frame in the case of video, if required. An animator extracts objects from the frame, and manually manipulates them to create the left and right eye views. While producing very convincing results, it is difficult and time consuming, and will inevitably be quite expensive, due to the large requirement of manual operators. This is very prohibitive to all but the largest of studios, and thus makes conversion difficult for smaller studios, amateur film makers, and even consumers. Despite these problems, 2D to 3D Conversion is very important to the stereoscopic post processing process, and should not be dismissed.

Natural stereoscopic filming is an option, but can become difficult and expensive, and converting single-view footage into stereoscopic 3D can become useful in cases where filming directly in 3D is too costly, or difficult. Research into conversion techniques is on-going in order to minimize this labour-intensive process. Most methods focus on the automatic viewpoint to extract depth information from an image or frame. However, even when minimizing the amount of user intervention for faster conversion, these can become extremely difficult to control the results of the conversion. Also, any errors that occur in the process cannot be easily corrected. No provision is in place to correct objects that appear at the wrong depths while converting, and may require extensive pre-processing or post-processing to correct. Therefore, there are advantages to pursuing a user-guided, semi-automatic approach to 2D to 3D conversion.

II. STATE OF THE ART

Three approaches for 2D to 3D conversion are: semiautomatic conversion, automatic conversion and manual conversion. In semiautomatic approach, for a single image or frame, the user simply marks objects and regions in an image. The close and far objects are denoted by lighter and darker intensities respectively. The depth labelling does not need to be accurate; it is just considered as an additional user input data. The end result is that the final depth for the image. In the case of automatic methods, no user interactions are needed and a computer algorithm automatically estimates the depth for a single image or video. Most methods concentrate on automatic methods. But the main problems of automatic methods are, difficult in error detection and correction. It may also require extensive pre/post processing. In the case of manual conversion, an trained person extracts objects from the image and manually manipulates or process them to create the left and right views. While producing good results, it is difficult and time consuming. Also it is very expensive because of the large requirement of manual operators.

In [1], the depth map is generated based on an image classification technique which classify the image either as outdoor or indoor one. Watershed and random walk [8] preserve sharpness and smoothness. But these can't be applicable to videos. Guttmann [2] introduce a new method for videos but it requires a large amount of memory and perception of depth in the middle of shot is lost. Bi-directional disparity propagation in [5] overcomes this. The depth map generated in [6] does not have sharp boundaries. In [7] sharpness is there, but the problem is depth values are constant inside the object. In [10] a method is discussed which is only applicable to videos and it must be applied to every video frame separately. In [11], the method is robust and it can be applicable to both images and videos. Here the depth map depends on a combination of two segmentation algorithms. It overcomes the problems in [8] but take more execution time.

III. SYSTEM ARCHITECTURE

The proposed system is mainly based on the two existing segmentation algorithms namely Random walk and Seed growing based Graph cut. The user has the option to provide initial depths and these initial depths are used for estimating the depth of the remaining pixels in the image. The marked image is segmented based on Random Walk segmentation algorithm. Again the same initial marked image is segmented based on Graph cut based seed growing method. The final depth map is the combined output of random walk output and graph cut output. If the final depth map is not an accurate one, then again mark the seeds on the image or in the first frame of video and re-run the algorithm.

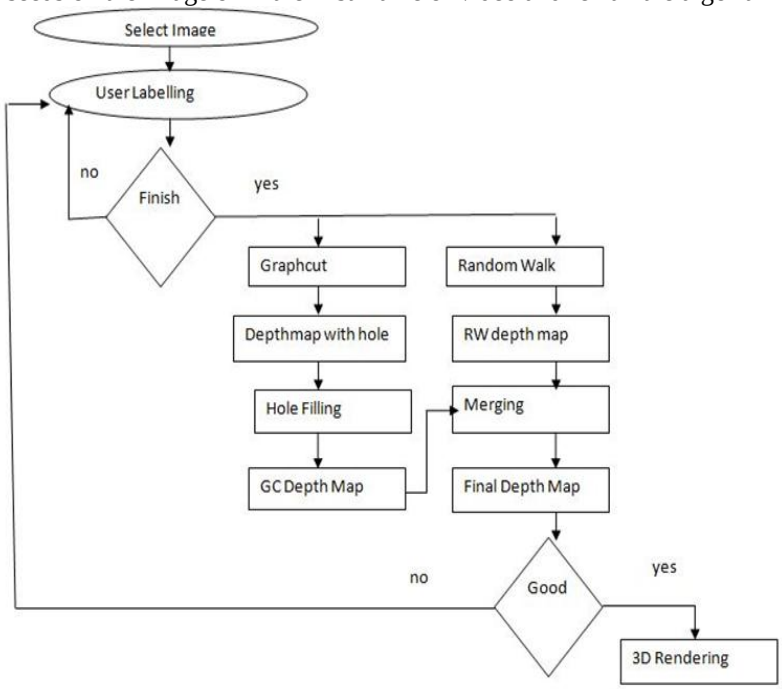

Fig 1: Pipeline of the system

The major modules of the proposed system are :

A. User Labelling

The system allows the user the choice of providing an initial estimate of the depth. That is the system allows, user can marks areas in an image with seed points on what the user think the best depth map should be and use this information to determine the rest of the depths. Also the user marks objects and regions to what he believes close or far from the camera. However, the user labelling need not be accurate, but it has to be perceptually consistent; because these initial depths are used for estimating the depth of the rest of the pixels in the image.

B. Graph Cut Segmentation

The Graph Cut automatic segmentation algorithm is a hard segmentation algorithm. The algorithm only creates result with depths/labels provided by user in the user labelling stage. Here the segmentation problem is making the depth maps for images and videos. Specifically, making the depth map is a multi-label segmentation problem.

As depth map generation can be considered as a multi-label classification problem, but Graph Cuts is actually a binary classification problem. Therefore, each unique user-defined depth value is assigned an integer label. Then a binary segmentation is performed separately for each label, i.e, the user defined labels having the user defined label are considered as foreground and the other portion of the image as background.

Graph Cuts is run for a total of N times, where N represents the total number of unique depths, once for each label. Each time a binary classification is performed for each label or depth. Based on the spatial distance between each seed points with all other pixels and also the intensity difference of each seed points with all other pixels, a weight matrix is created. The weight matrix shows the affinity of a seed point towards all other pixels. Each pixel is assigned with the label of the seed point which has more affinity towards the pixel. If a pixel was only assigned to one label, then that is the label it is assigned. However, if a pixel was assigned multiple labels, assign the label of the seed point to which the pixel

has most affinity. The pixels having similar labels are grouped to form a cluster. This way by seed growing the seed points each segment will be generated.

The only thing to be noted in the graph cut segmentation is, it is a binary segmentation algorithm. So to convert it to multi-label, each unique user-defined label is given an integer label. Then binary segmentation is performed for each user-defined label. In some cases, even this will not always result in every pixel being classified, but region filling methods can be used to correct this.

C. Hole Filling

A hole is defined as an area of dark pixels surrounded by lighter pixels, in the case of gray scale images. The depth map is a gray scale map where the intensity represents depth. In certain cases, there are a number of holes present in the depth map output of graph cut. Holes are created in depth maps because of local intensity variations. The holes with area less than a previously specified threshold are marked as imperfections and are filled. Small non joint clusters in a segment are marked as noise and are removed from each segment.

D. Random Walk

Random Walk is a soft segmentation algorithm. It will start at some label, visiting all of the unlabelled nodes in the image and finds the similar ones. The walker is biased by the edge weights inorder to visit similar nodes than dissimilar ones. For the purpose of image segmentation, the goal is to classify every pixel in an image to belong to one of K possible labels.

Random Walks determine the probability of each pixel belonging to one of these labels, and the label that the pixel gets classified as is the one with the highest probability. This is performed by solving the combinatorial Dirichilet problem. i.e, the problem of finding the random walk probability has the same solution as the combinatorial Dirichilet problem which is the problem of finding a harmonic function subject to its boundary values.

Choosing the proper edge weighting is important. The edge weighting determines how similar, or dissimilar, two pixels are. The Eucledian distance represented how different two pixels were from one another. A Eucledian distance of zero indicates that the two nodes are identical while a great distance implies the two pixels are completely dissimilar. Consider some distance function, say d(i,j). Distance has the convenient property of being zero when two objects are ”close together” (the same) and being very large when they two objects are ”far apart” (very different). For pixels, these objects would be the colour vector, Ci. The distance function is written as

$$d(i, j) = II\ (Ci - Cj)II$$

works very well in most cases, where II II is the Euclidean norm.

However, the expression shows a similarity measure. Therefore, the weighting should describe similarity, not dissimilarity. The inverse of the distance cannot be used because the result will be infinity if the two data points are the same and therefore it results in computational problems. Grady’s choice of a weighting function is

$$Gi, j = exp{-}\beta(i, j)$$

The value of β is taken as 90. The two matrices, Gu and Gs, can be considered as ”sub-adjacency matrices”. The graph connections all of the neighbors that are unknown (Gu) and all of the neighbors that are seeds (Gs). The complete adjacency matrix is G = Gu + Gs. The matrix GE is the degree matrix of G. Each element along the main diagonal of GE is the sum of all of the edge weights connected to that particular node. The final form of the Random Walker algorithm so that

$$(GE - Gu)v = Gss$$

$$Lv = b$$

where L = GE − Gu is a Laplacian matrix and b = Gss is the boundary vector. Since it is not necessary to solve for the already known nodes, the final solution to system is simply

$$v = L^{-1}b$$

Thus the probabilities for the unknown pixels are identified. Then this probability is directly used as the depths.

E. Object Tracking in Video

In the proposed system object present in the frames are tracked by the optical flow method (Iterative Pyramidal Lucas-Kanade method). Iterative Pyramidal Lucas-Kanade method is a differential method for optical flow estimation. There is some assumption to consider when using this object tracking method.

- Neighbouring pixels of the pixel under consideration have similar motion or constant flow
- Displacement of the image contents or the pixel intensities of an object between two nearby frames is small

Let I, J be two, 2D (two dimensional) grayscale images. And Let x1= (x, y) be a pixel location in image plane. Then, I(x1)= I(x, y) and J (x1) =J (x, y) The image I(x1) will be referred to as the first image and J(x1) as the second image. Let upper left corner pixel coordinate vector is (0, 0). The nx and ny be the width and height of the images respectively. The lower right pixel coordinate vector is (nx-1, ny-1). Consider a specific image point u =(ux ,uy) on the first image I . The goal of feature tracking is to find the location v =u+ d, on the second image J, such that I(u) and J (v) are almost equal. If they are not similar find the updated velocity and apply it to the first image upto they are similar. In certain cases, it needs a number of velocity updating and similarity comparisons. So in the proposed system a threshold for the iteration is specified. Again once the tracking between two frames was over, then the current second frame become first, i.e, J(v) become I(u) and third frame become J(v) and so on. This process will continue on each specified pyramid level.

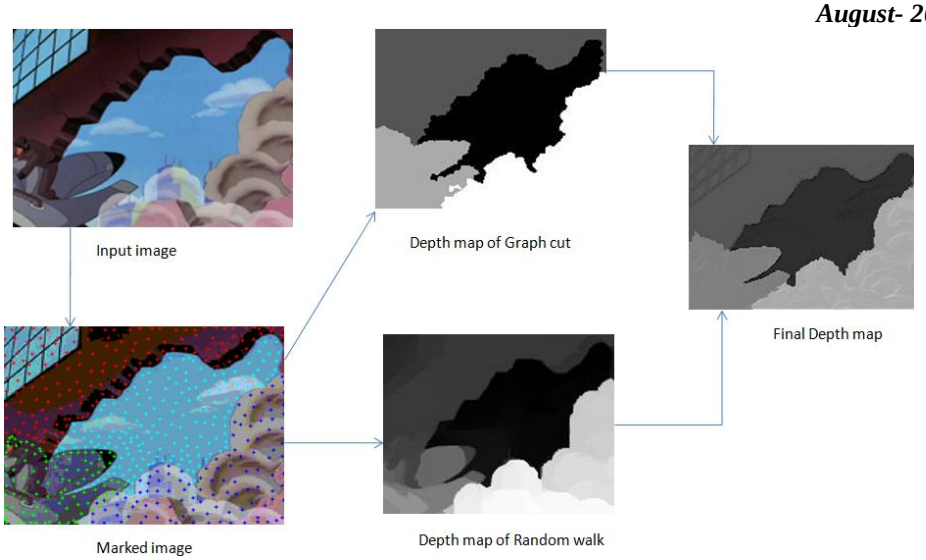

Fig 2 : Input image, Graph cut output, Random walk output and Final depth map

F. α Parameter Estimation

α is a scaling factor which determines how much effect the Graph Cuts result has on the final depth map. In the earlier work, this weighing scheme was static, and was manually determined depending on the image that was used by some methods and guesses. To overcome this situation, in this work, the system determines the weight of graph cut adaptively using edge information from the image. The objective is to allow the Graph Cuts result to dominate in areas of strong edges due to its edge preserving properties, while suppressing the depth map everywhere else. To do this, apply a Sobel edge detector to the grayscale counterpart of the image, and α is generated for each set pixel using the following formula:

$$\alpha = static{-}alpha + (normalized{-}contrast{-}measure * dynamic{-}alpha)$$

α will be high for high contrast regions like edges and α will be low for low contrast regions like surfaces. i.e, if the input image contains more surface areas then the high preferences is given to random walk. This is done adaptively decreasing the alpha value for graph cut.

G. Depth Map Generation

The user gives the initial depth estimate, providing a rough sketch of the overall depth in the scene. To merge the two depth maps together to achieve greater accuracy, a depth map of graph cut is created. Graph Cuts depth map should help in maintaining the strong boundaries in the Random Walks depth map. Before merge the two depth maps together, modify the depth map of graph cut or adaptively set its priority or importance . The depth map of Random Walks is in the continuous range of [0, 1]. But the depth map for Graph Cuts is in the integer range of [1, N] based on the number of uer defined labels. Both depth maps correspond to each other, but one needs to be transformed into the other then only one can merge these two. The goal is to transform the integer labelling into a compatible labelling for Random Walks. This can be done by performing each labelling by graph cut, within the range of [0,1] for Random Walks. When Graph Cuts has completed, this one is used to map back to the continuous range. To finally merge the two depth maps together, use the following equation.

$$depthmap = \alpha * (OutputofGraphcut) + (1 - \alpha) * (OutputofRandomWalk)$$

H. Stereo Rendering

Most commonly the original 2D image or the input image is treated as the centre image. Here also the original image is used as one eye's image and so to generate the other eye's image the conversion cost is less. During stereo generation, pixels of the original image are shifted to the left to generate the other eye's image. Reconstruction and painting of any uncovered areas in the 3D image are not filled by the stereo generator.

Algorithm Steps

1. Select an image.
2. Mark depth labels by user.
3. Local contrast estimation on input image.
   (a) Set high for RW low contrast region
   (b) Set high for GC high contrast region
4. Perform seed growing GC on the input image
   (a) Generates a weight matrix which shows the affinity of a seed point towards all other pixels
   (b) Weight combines both the similarity measure in intensity and spatial distance between two points
   (c) Each pixel is assigned with the label of the seed point which has more affinity towards the pixel
   (d) Each segment is then assigned with the weight of the corresponding label
5. Apply Hole Filling on graph cut output
   (a) Specify a threshold

(b) Holes with area less than a threshold are marked as imperfections and are filled

6. Perform random walk on input image
   (a) Allocate region seeds for each region
   (b) Generate weights based on image intensities
   (c) Build laplacian matrix
   (d) Calculate the probability for each label
   (e) Each pixel is assigned with the label which has the highest probability
7. Generate final depth map
   (a) Combine output of GC ad RW

$$depthmap = \alpha * (OutputofGraphcut) + (1 - \alpha) * (OutputofRandomWalk)$$

8. 3D Rendering.

IV. PERFORMANCE EVALUATION

α is a scaling factor which determines how much effect the Graph Cuts result has on the weighting. In the previous works, this weight was static, and was empirically determined depending on the image that was used. To circumvent this situation, in this work, here determine α adaptively using edge information from the image. The objective is to allow the Graph Cuts result to dominate in areas of strong edges due to its edge preserving properties, while suppressing the depth map everywhere else. The resulting depth map with static alpha value of 0.5 and the depth map with dynamic alpha and the difference between these two in the figure are shown below.

Fig 3 : (a) depth map with static alpha value of 0.5 (b) depth map with dynamic alpha (c) difference between these two

The overall time taken to convert the images is dependent on the complexity of the scene, the size of the image, and the number of user-defined labels placed. In the proposed system, as the labelling count increases, the execution time also increases. On average, it takes between 30 seconds to 1 minute for labelling, while roughly a couple of seconds to generate the depth map. The accuracy depends on the labelling count. The accuracy of the system increases with labelling count.

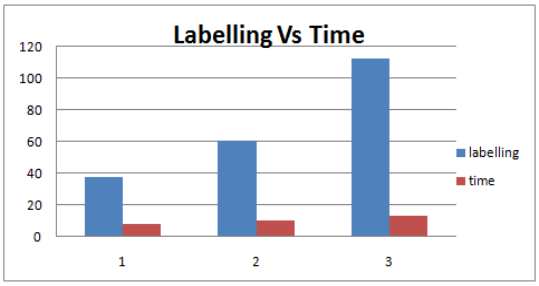

Fig 4 : Relationship between time and labelling

V. CONCLUSIONS

A single solution to convert the 2D images or videos to 3D does not exist. The conversion problem is yet a research area. Here presented a semi-automatic method for obtaining depth maps to convert a 2D image and video into stereoscopic 3D. A semi automated algorithm is preferable to an automated one as we can directly control the perceived depth for objects in the scene. Also here combined two existing segmentation algorithms namely Graph cut and Random Walk in a novel way to produce stereoscopic image pairs. Also used a hole filling method to correct the holes present in the depth map. The goal of this work was to adaptively combine the Seed growing based Graph Cuts and Random Walks algorithms in order to produce good quality depth maps. Sometimes the user must rerun the algorithm with new strokes on the image, to get a good quality depth map.

ACKNOWLEDGMENT

The authors express their sincere thanks to HOD, group tutor and staff in Computer Science department, Viswa Jyothi College of Engineering and Technology for many fruitful discussions and constructive suggestions during the implementation of this paper.

REFERENCES

[1] S. Battiatoa, S. Curtib, M. La Casciac, M. Tortorac, E. Scordatoca, "Depth-Map Generation by Image Classification," ACM International Conf. on Multimedia 2010.

[2] M. Guttmann, L. Wolf and Daniel Cohen-Or, "Semi-automatic Stereo Extraction from Video Footage," in Proc. IEEE Conf. Comput. Vis., Oct. 2009, pp. 136- 142.

[3] Richard Rzeszutek, Raymond Phan and Dimitrios Androutsos, ”Depth Estima tion for Semi-Automatic 2D to 3D Conversion,” ACM International Conferen ce on Multimedia Oct 2012.

[4] Xun Cao, Zheng Li and Qionghai Dai, ”Semi-Automatic 2D-to-3D Conversion Using Disparity Propagation,” IEEE Trans. on Broadcasting, vol. 57, no. 2, June 2011.

[5] Miao Liao, Jizhou Gao, Ruigang Yang and Minglun Gong, ” Video Stereoliza tion: Combining Motion Analysis with User Interaction,” in Proc. IEEE Trans. on Visualization nad Computer Graphics, vol. 18, no. 7, July 2012.

[6] Richard Rzeszutek, Raymond Phan and Dimitrios Androutsos, Semi-automatic Synthetic Depth Map Generation For Video Using Random Walks, IEEE International Conference on Multimedia and Expo (ICME) 2011, pp. 1-6.

[7] Pei-Jun Lee, Wen-Jay Yu and Chi-Feng Chuang, ”Watershed-based Stereo Match ing Algorithm for Depth Map Generation” IEEE Global Conference on Consumer Electronics 2012, pp. 411-412

[8] Xuyuan Xu, Lai-Man Po, Kwok-Wai Cheung, Ka-Ho Ng, Ka-Man Wong and Chi-Wang Ting, ”Watershed and Random Walks based Depth Estimation for Semi-Automatic 2D to 3D Image Conversion” IEEE International Conference 47 on Signal Processing, Communication and Computing (ICSPCC) 2012, pp. 84- 87.

[9] Ikuko Tsubaki, Atsutoshi Shimeno, Takeshi Tsukuba, Takeaki Suenaga and Masahiro Shioi ”2D to 3D conversion based on tracking both vanishing point and objects,” IEEE Global Conference on Consumer Electronics 2012, pp. 110- 14

[10] Tsung-Han Tsai, Chen-Shuo Fan and Chih-Chi Huang, ”Semi-automatic Depth Map Extraction Method for Stereo Video Conversion” IEEE International Con ference on Genetic and Evolutionary Computing 2012, pp. 340-343.

[11] Raymond Phan and Dimitrios Androutsos, ”Robust Semi-Automatic Depth Map Generation in Unconstrained Images and Video Sequences for 2D to Stereo scopic 3D Conversion”, IEEE Trans. on Multimedia, vol. 16, no. 1, Jan 2014

[12] Z. Kalal, K. Mikolajczyk, and J. Matas, Tracking-learning-detection,IEEE Trans. Pattern Anal. Mach. Intell., vol. 6, no. 1, pp. 114, 2010.

[13] C.Chahine and A.Nakib, On the random walks algorithms for image processing, LISSIE E.A 3956 Universite Paris, Creteil, France

[14] E.A. Zanaty and S. F. El-Zoghdy, New Region Growing based on Thresholding Technique Applied to MRI Data I. J. Computer Network and Information Sec urity, 2015, 7, 61-67

