Learning Based 2D to 3D Conversion with Input Image Denoising

Divya K.P.[1], Sneha K.[2], Nafla C.N.[3]
[1](Department of CSE, RCET, Akkikkvu, Thrissur)
[2] (Asst. Professor, Department of CSE, RCET, Akkikkvu, Thrissur)
[3] (Department of CSE, RCET, Akkikkvu, Thrissur)

ABSTRACT

The availability of 3D content to that of the 2D counterpart is still dwarfed despite of the significant growth in last few years. To close this gap, many 2D-to-3D image and video conversion methods have been proposed. Specifically, the sense of depth in a scene captured by conventional cameras, where virtually no depth was ever provided, can now be experienced. A sense of depth can be used in many applications, like training simulations, gaming, scientific model exploration, and in cinemas. All these applications can use a sense of depth to get a sense of realism, and thus the users can truly appreciate the content that they are viewing. The method proposed here is a global method of estimating the entire depth of an input query image from a 3D repository(3D images+depth) by using a nearest neighbour regression type idea.

Keywords *- 3D images, depth, disparity, stereoscopic images.*

1. INTRODUCTION

Viewing 3D content, whether it be in the home, at the cinema, or in scientific environments, has become popular over the last few years. In the past, much research focused on obtaining a sense of depth of a scene by using only two views, corresponding to the left and right eye viewpoints, known as stereo correspondence. Using a reference viewpoint, whether it is the left or right eye, a sense of depth is determined for each point in the reference, where points in one image are matched with their corresponding points in the other image. The amount of shift that the corresponding points undergo is known as disparity. Disparity and depth have an inverse relationship, whereby higher the horizontal shift, or disparity, the smaller the depth, and the closer the point is to the viewer.

By presenting these two views of a scene to a human observer, where one is slightly offset from the other, this produces the illusion of depth in an image, thus perceiving an image as 3D. The work is primarily done by the visual cortex in the brain, which processes the left and right views presented to the respective eyes. In the past, the disparities of each point between both viewpoints are collected into one coherent result, commonly known as a disparity map. These are of same size as either of the left and right images. The image is monochromatic (grayscale) in nature, and its brightness at each point is directly proportional to the disparity between the corresponding points. When a point is lighter, this corresponds to a higher disparity, and will be closer to the viewer. Similarly, when a point is darker, this corresponds to a lower disparity, and will be farther from the viewer. For scenes that have smaller disparities in overall, the disparity maps are scaled accordingly, so that white corresponds to the highest disparity.

However, filming directly in 3D can be rather expensive and difficult to set up. So the one we are ultimately interested in, is to take already existing 2D content, and produce the left and right views artificially, or other views for multiview applications. The amount of materials is endless, but the methods required to convert monocular video to 3D is a very difficult problem to solve, and still one of the most open-ended problems in computer vision that exist today. When a camera captures a scene, it actually performs a transformation from 3D to 2D coordinates, which is well understood. However, while considering the inverse situation, which is the one we are interested in, our attempt is to produce information from a model that lost the very information we are trying to obtain. No matter the difficulties, the end result will be quite beneficial to anyone interested in the 3D experience, as conventional content can now be viewed in 3D, with a sense of depth that was never obtained before. The goal for the conversion of monocular video to its stereoscopic or multiview counterpart is to generate a set of disparity maps for each view that we wish to render. Each disparity map will determine the shift in pixels we need from the reference view, or a frame from the video sequence. The caveat here is that we must generate disparity maps for each frame of each view in the video sequence. In addition, most of the 2D to 3D conversion algorithms actually calculate a depth map, rather than the disparity map, but a simple conversion between these two can be achieved once a couple of the parameters of the camera are known. Regardless, 2D to 3D conversion algorithms have been the subject of many useful applications which are targeted for the industries, as well as the end user. Specifically, 2D to 3D conversion techniques are of primary consideration in the latest 3DTVs. This is due to the fact that 3D technology is now widely available for the home, and they can now experience their existing 2D content in 3D.

A typical 2D-to-3D conversion process consists of two steps: depth estimation for a given 2D image and depth-based rendering of a new image in order to form a stereopair. While the rendering step is well understood and algorithms exist that produce good quality images, the challenge is in estimating depth from a single image (video). Therefore, throughout this paper the focus is on depth recovery and not on depth-based rendering, although we will briefly discuss our approach to this problem later.

There are two basic approaches to 2D-to-3D conversion: one that requires a human operator's intervention and one that does not. In the former case, the so-called semi-automatic methods have been proposed where a skilled operator assigns depth to various parts of an image/video. Based on this sparse depth assignment, a dense depth over the entire image or video sequence is estimated by the underlying computer algorithm. The involvement of a human operator may vary from just a few scribbles to assign depth to various locations in an image to a precise delineation of objects and subsequent depth assignment to the delineated regions.

In the case of automatic methods, no operator intervention is needed and a computer algorithm automatically estimates the depth for a single image (or video). To this effect, methods have been developed that estimate shape from shading, structure from motion or depth from defocus. Although these methods have been shown to work in some restricted scenarios they do not work well for arbitrary scenes.

The paper is organized as follows. In Section II, a review of the state of the art in 2D-to-3D image conversion is done. In Section III, we provide details of the global approach to the conversion. In Section IV, we show numerous experimental results and we conclude the paper in Section V.

2. STATE OF THE ART

There are two types of 2D-to-3D image conversion methods: semi-automatic methods, that require human operator intervention, and automatic methods, that require no such help.

2.1 Semi Automatic Method

Most of the semiautomatic methods of stereo conversion use depth maps and depth-image-based rendering.

The idea is that a separate auxiliary picture known as the "depth map" is created for each frame or for a series of homogenous frames to indicate depths of objects present in the scene. The depth map is a grayscale image having the same dimensions as the original 2D image, with different shades of gray to indicate the depth of every part of the frame. Although depth mapping can produce a fairly potent illusion of 3D objects in the video, it does not support semi-transparent objects or areas, does not allow explicit use of occlusion etc., so these issues should be dealt with a separate method.

A development on depth mapping, multi-layering works around the limitations of depth mapping by introducing several layers of grayscale depth masks to implement limited semi-transparency. Similar to a simple technique, multi-layering involves applying a depth map to more than one "slice" of the flat image, which can result in a much better approximation of depth and protrusion. The greater the number of layers is processed separately per frame, the greater the quality of 3D illusion tends to be.

3D reconstruction and re-projection may be used for stereo conversion. It involves 3D scene model creation, extracting original image surfaces as textures for 3D objects and, following it, rendering the 3D scene from two virtual cameras to acquire stereo video. The approach works well in case of scenes with static rigid objects like urban shots with buildings; interior shots etc., but have problems with non-rigid bodies and soft fuzzy edges.

Another method is to set up both left and right virtual cameras, both offset from the original camera with splitting the offset difference, and then inpainting occlusion edges of isolated objects and characters. Essentially clean plating several backgrounds, mid ground and foreground elements. Binocular disparity can also be derived from simple geometry.

2.2 Automatic Method

It is possible to automatically estimate depth using different types of motion. In case of motion of camera depth map of the entire scene can be calculated. Also, motion of objects can be detected and moving areas can be assigned with smaller depth values than the background. Moreover, occlusions provide information on relative position of moving surfaces.

On "depth from defocus" (DFD) approaches, the depth estimation is done based on the amount of blur of the object considered, whereas "depth from focus" (DFF) approaches tend to compare the sharpness of an object over a range of images taken with different focus distances in order to find out its distance to the camera.

DFD needs only 2 to 3 images at different focus to properly work, whereas DFF needs a minimum of 10 to15 images but is more accurate than the previous method.

The idea of depth from perspective is based on the fact that parallel lines, such as railroad tracks, roadsides etc., appear to converge with distance, eventually reaching a vanishing point at the horizon. Finding out this vanishing point gives the farthest point of the whole image.

Recently, machine-learning-inspired techniques employing image parsing have been used to estimate the depth map of a single monocular image [2], [3]. Such methods have the potential to automatically generate depth maps, but it currently work only on few types of images (mostly architectural scenes) using carefully-selected training data (precise, laser scanned depth estimates or manually-annotated semantic depth classes). In the quest to develop data-driven approaches to 2D-to-3D conversion we have also been inspired by the recent trend to use large image databases for various computer vision tasks, such as object recognition [4] and image saliency detection [5].

3. 2D TO 3DCONVERSION BASED ON GLOBAL METHOD OF DEPTH LEARNING WITH INPUT IMAGE DENOISING

In this section a method that estimates the *global* depth map of a query image or video frame directly from a repository of 3D images (image+depth pairs or stereopairs) using a nearest-neighbor regression type idea is developed.

The approach proposed here is built upon a key observation and an assumption. The observation is that among millions and billions of 3D images available on-line, there likely to exist many whose 3D content matches that of a 2D input (query) we wish to convert to 3D. We are also making an assumption that two images that are photometrically similar also have similar 3D structure (depth). This is not unreasonable since photometric properties are often correlated with 3D content (depth, disparity). For example, edges in a depth map almost always coincide with photometric edges. Given a monocular query image *Q*, assumed to be the left image of a stereopair that we wish to compute, we rely on the above observation and assumption to “learn” the entire depth field from a repository of 3D images *I* and render a stereopair in the following steps:

1) Image denoising: removing the noise of input query images using NL means filtering.

2) search for representative depth fields: find *k* 3D images in the repository *I* that have most similar depth to the query image, for example by performing a *k* nearest-neighbor (*k*NN) search using a metric based on photometric properties.

3) Depth fusion: combine the *k* representative depth fields, for example, by means of median filtering across depth fields.

4) Depth smoothing: process the fused depth field to remove spurious variations by preserving depth discontinuities, for example, by means of a cross-bilateral filter.

5) Stereo rendering: generate the right image of a fictitious stereopair using the monocular query image and the smoothed depth field followed by suitable processing of occlusions and newly-exposed areas.

Specific details of these steps depend on the type of 3D images contained in the repository. The above steps apply directly to 3D images represented as an image+depth pair. However, in the case of stereopairs a disparity field needs to be computed first for each left/right image pair. Then, each of these disparity field can be converted to a depth map, e.g., under a parallel camera geometry assumption, with fusion and then smoothing taking place in the space of depths. Alternatively, the fusion and smoothing can take place in the space of disparities (without converting to depth), and the final disparity is used for right-image rendering.

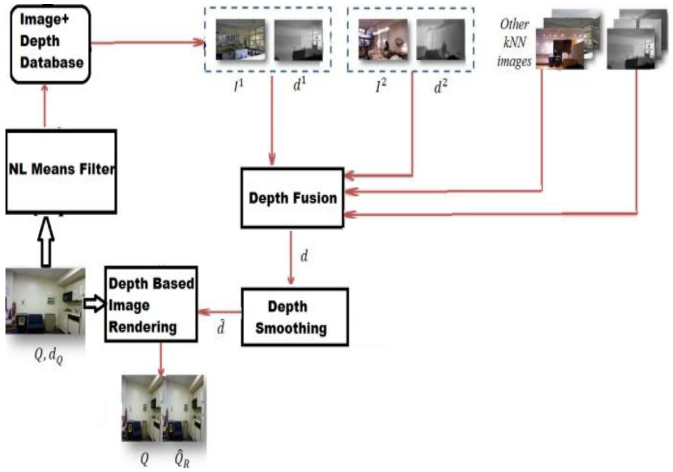

Fig 1: Block diagram of overall approach

Fig. 1 shows the block diagram of the approach. The sections below provide a description of each step and some high level mathematical detail. In these sections, Q_R is the rightimage which is being sought for each query image *Q*, while d_Q is the query depth (ground truth) needed to numerically evaluate the performance of a depth computation. Again, we assume that a 3D dataset *I* is available by means of laser range finding, in Make 3D dataset. The goal is to find a depth estimate $\widehat{d}$ and then a right-image estimate $\widehat{Q_R}$ given a 2D query image *Q* and the 3D dataset *I*.

The idea of the proposed work is from Konrad et al[1] in which only gradients are used for feature extraction in kNN search. Here we propose image denoising for input images and wider feature selection for similarity search through GLCM features.

3.1 Input Image Denoising

Input image denoising is done using NL filter. Non local means is the algorithm used for image denoising. Unlike local means filters, which take the mean value of a group of pixels surrounding a target pixel to smooth the image, NL means filtering takes a mean of all pixels in the image, weighted by the similarity of these pixels to the target pixel. This result in much greater post-filtering clarity and lesser loss of detail in the image compared with local mean algorithms. NL means filter uses all the possible self-predictions and self-similarities the image can provide to determine pixel weights or filtering the noisy image, with the assumption that the input image contains an extensive amount of self-similarity. Since the pixels are highly correlated and the noise is typically independently and identically distributed, finding average of these pixels results in noise suppression there by yielding a pixel that is similar to its original value. The NL means filter removes the noise and cleans the edges without losing too many fine structures and details.The query image Q is directly given to NL means filter inorder to remove the noise. The resulting output image is then forwarded to kNN phase.

3.2 kNN Search

There exist two types of images in a large 3D image repository: those that are relevant for determining depth in a 2D query image, and those that are not relevant. Images that are not photometrically similar to the 2D query need to be rejected because they are not useful for estimating depth (as per our assumption). Note that although we might miss some depth-relevant images, we are effectively limiting the number of irrelevant images that could potentially be more harmful to the 2D-to-3D conversion process. The selection of a smaller subset of images provides the added practical benefit of computational tractability when the size of the repository is very large.

One method for selecting subset of depth-relevant images from a large repository is to select only the *k* images that are closest to the query where this closeness is measured by using some distance function capturing global image properties such as color, texture, edges, etc. As this distance function, the Euclidean norm of the difference between extracted feature properties of query image to that of the dataset images is used. This is implemented using GLCM features, Hierarchical Centroid and Color Statistics for texture, edge and color features respectively. This can earn better retrieval compared to HoG method. First of all the color, texture and shape features of the input query Q is calculated then that of the images in dataset is calculated one by one. With the results the difference of Q with those of the dataset images is calculated using Euclidean norm. We perform a search for top matches to our monocular query *Q* among all images _*I k* , *k* = 1, ..., *K* in the 3D database *I*. The search returns an ordered list of image+depth pairs, from the most to the least photometrically similar *vis-à-vis* the query. We discard all but the top *k* matches (*k*NNs) from this list.

Fig. 2 shows search results for two outdoor query images performed on the Make3D dataset #1. Although none of the four *k*NNs perfectly matches the corresponding 2D query, the general underlying depth will be somewhat related to that expected in the query. While some of the retained images share local 3D structures with the query image, other images do not.

The average photometric similarity between a query and its *k* th nearest neighbor usually decays with the increasing *k*. While for large databases, larger values of *k* may be appropriate, since there are many excellent matches, for smaller databases this may not be true. Therefore, a judicious selection of *k* is important. We denote by *K* the set of indices *i* of image+depth pairs that are the top *k* photometrically-nearest neighbours of the query *Q*.

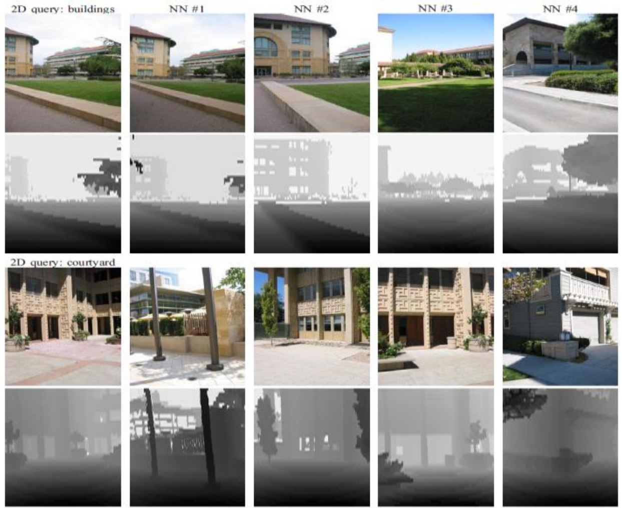

Fig 2: RGB image and depth field of two 2D queries (left column), and their four nearest neighbors (columns 2–5) retrieved

3.3 Depth Fusion

In general, none of the NN image+depth pairs (I_i, d_i), $i \in K$ match the query *Q* accurately. However, the location of some objects (e.g., furniture) and parts of the background (e.g., walls) is quite consistent with those in the respective query. If a similar object (e..g, building) appears at a similar location in many *k*NN images, it is likely that such an object also appears in the query image, and the depth field being sought should reflect this. We calculate this depth field by applying the mean operator across the *k*NN depths at each spatial location x as follows:

$$d[x] = \text{mean}\{di \quad [x] \quad \forall i \quad \in \quad K\}. \quad (1)$$

Although these depths are overly smooth, they provide a globally correct, although coarse, assignment of distances to various areas of the scene.

3.4 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 edge misalignment between the depth fields of the *k*NNs and the query image. This often results in the lack of edges in the fused depth where sharp object boundaries should occur and/or the lack of fused-depth smoothness where smooth depth is expected.

In order to correct this, similarly to Agnot *et al.* [6], a cross-bilateral filtering (CBF) is applied. CBF is a variant of bilateral filtering, which is an edge-preserving image smoothing method that applies anisotropic diffusion controlled by the local content of the image itself [7]. In CBF, however, the diffusion is not controlled by the local content of the image under smoothing but by an external input. CBF is applied to the fused depth *d* by using the query image *Q* to control diffusion. This allows the conversion to achieve two goals simultaneously: alignment of the depth edges with those of the luminance *Y* in the query image *Q* and local noise/granularity suppression in the fused depth *d*.

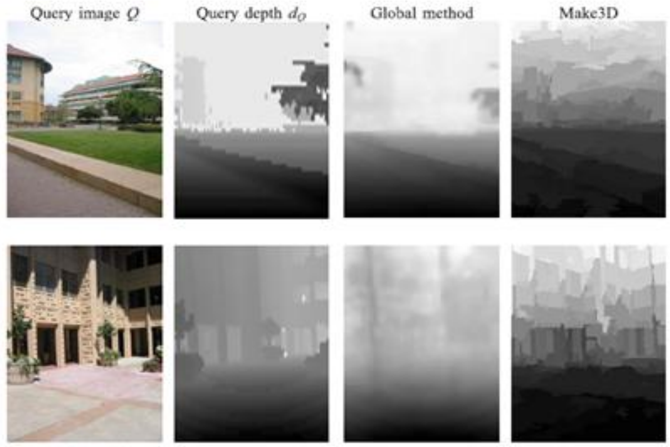

Fig 3: Query images and depth fields: of the query, depth estimated by the global transformation method (with CBF) and depth computed using the Make3D algorithm.

In Fig. 3, we show an example of mean-fused depth field after cross-bilateral filtering. Clearly, the depth field is overall smooth (slowly varying) while depth edges, if any, are aligned with features in the query image. The filtered depth preserves the global properties captured by the unfiltered depth field *d*, and is smooth within objects and in the background. Along with it keeps edges sharp and aligned with the query image structure.

3.5 Stereo Rendering

In order to generate the right image estimate $\widetilde{Q_R}$ from the monocular query image *Q*, we need to compute a disparity δ from the estimated depth $\widetilde{d}$. Simple inpainting using inpaint_nans from *MatlabCentral is applied* for handling newly exposed areas while rendering. Applying a more advanced depth-based rendering method would only improve this step of the proposed 2D-to-3D conversion.

4. EXPERIMENTAL RESUTS

We have tested our approach on Make3D outdoor dataset with depth fields captured by a laser range finder . Note that the Make3D[8] images are of 1704 × 2272 resolution but the corresponding depth fields are only of 55 × 305 spatial resolution and relatively coarse quantization. Therefore, for computational efficiency, we have re-sized the images to 240 × 320 resolution. We selected one image+depth pair from a database as the 2D query treating the remaining pairs as the 3D image repository *I* based on which a depth estimate $\widetilde{d}$ and a right-image estimate $\widetilde{Q_R}$ are computed. As the quality metric, we used normalized cross-covariance C between the estimated depth $\widetilde{d}$ and the ground-truth depth d_Q. The normalized cross-covariance *C* takes values between −1 and +1 (for values close to +1 the depths are very similar and for values close to −1 they are complementary).

In addition to this with the modification of image denoising for getting error free input images, the error rate of the output has also been reduced. This performance improvement of the algorithm is evaluated using the attributes Mean Squared Error (MSE) and its corresponding Peak Signal to Noise Ratio (PSNR). The performance graphs are shown in Fig.5 and Fig.6.

Table 1. Average and Normalized Cross-Covariance C Computed Across all Images in Make 3D for Proposed Method, Make 3D algorithm, Konrad et al[1]

	Global GLCM	Make 3D	Konrad et al
Average C	0.82	0.78	0.80
Median C	0.88	0.78	0.86

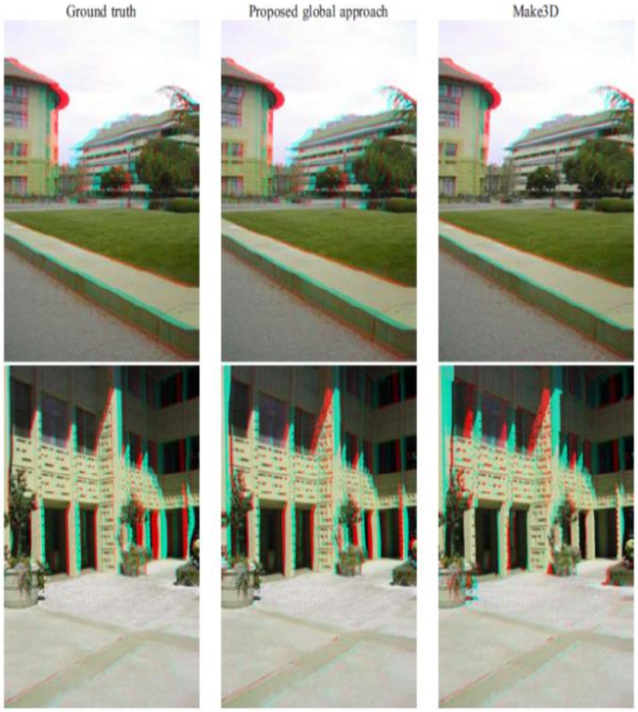

Fig 4: Anaglyph images generated using the ground-truth depth and depths estimated by the proposed global and Make3D algorithms.

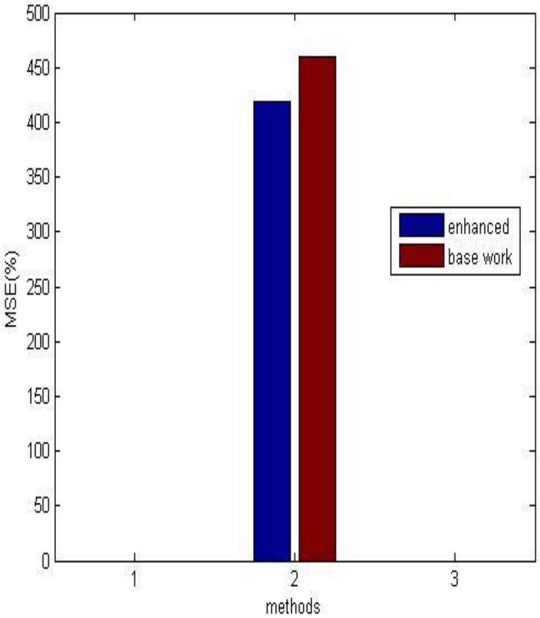

Fig. 5 Performance evaluation of the proposed method with Konrad[1] using MSE

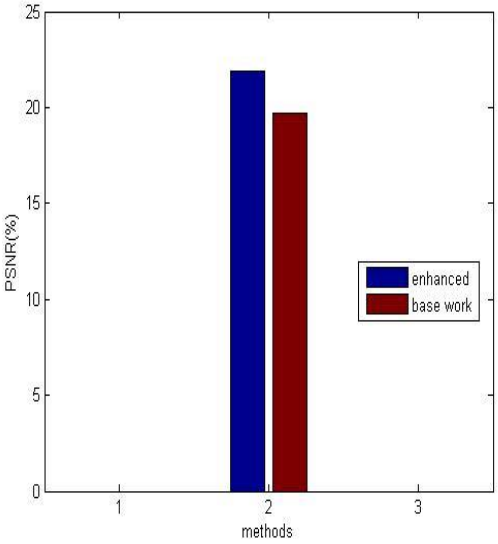

Fig. 6 Performance evaluation of the proposed method with Konrad[1] using PSNR

5. CONCLUSION

We have proposed a new class of methods aimed at 2D-to- 3D image conversion that are based on the radically different approach of learning from examples. The method we proposed is based on globally estimating the entire depth field of a query directly from a repository of image+depth pairs using nearest neighbour based regression using GLCM for feature extraction. We have objectively validated our algorithms' performance against state-of-the-art algorithms. Our global method performed better than the state-of-the-art algorithms in terms of cumulative performance across Make 3D dataset and has done so at a fraction of CPU execution time. Anaglyph images produced by our algorithms result in a comfortable 3D experience but are not completely void of distortions. Certainly, there is room for improvement in the future. With the continuous increase in amount of 3D data on-line and with the rapidly growing computing power in the cloud, the proposed work seems a promising alternative to operator-assisted 2D-to-3D image and video conversion.

ACKNOWLEDGEMNTS

We express our gratitude to the experts who have contributed towards development of the project. We acknowledge Saxena the Make 3D dataset developer. Every success stands as a testimony not only to the hardship but also to hearts behind it. Likewise, the present work has been undertaken and completed with direct and indirect help from many people and we would like to acknowledge all of them for the same.

REFERENCES

1. J.Konrad, M. Wang, P. Ishwar, C. Wu and D. Mukherjee, Learning-Based, Automatic 2D to3D Image and Video Conversion, *IEEE Trans. Image Processing*, Vol.22, No.9. pp. 3485-3496, Sep 2013
2. B. Liu, S. Gould, and D. Koller, “Single image depth estimation from predicted semantic labels,” in Proc. IEEE Conf. Comput. Vis. Pattern *Recognit.*, Jun. 2010, pp. 1253–1260.
3. Saxena, M. Sun, and A. Ng, “Make3D: Learning 3D scene structure from a single still image,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 824–840, May 2009.
4. Torralba, R. Fergus, and W. T. Freeman, “80 million tiny images: A large data set for nonparametric object and scene recognition,” *IEEE* Trans. Pattern Anal. Mach. Intell., vol. 30, no. 11, pp. 1958–1970, Nov. 2008.
5. M. Wang, J. Konrad, P. Ishwar, K. Jing, and H. Rowley, “Image saliency: From intrinsic to extrinsic context,” in *Proc. IEEE Conf. Comput. Vis.Pattern Recognit.*, Jun. 2011, pp. 417–424.
6. L. Angot, W.-J. Huang, and K.-C. Liu, “A 2D to 3D video and image conversion technique based on a bilateral filter,” *Proc. SPIE*, vol. 7526, p. 75260D, Feb. 2010.
7. F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images,” *ACM Trans. Graph.*, vol. 21, pp. 257–266, Jul. 2002.
8. (2012). *Make3D* [Online]. Available: http://make3d.cs.cornell.edu/data.html