Automatic Image Conversion From 2D to 3D Using Support Vector Machine

Harini S
Dept.Of Computer Science
H.K.B.K College Of Engineering
Bangalore-45

***Abstract*—3D image and video applications are becoming popular in our daily life, especially at home entertainment. Although more and more 3D movies are being made, 3D image and video contents are still not rich enough to satisfy the future 3d image in market. There is a rising demand on new techniques for automatically converting 2d to 3d image. the most common method involves human operator and automatic conversion of image. The proposed system methods that are used to convert 2d to 3d image includes point mapping from local images which mainly includes attributes like color location and motion and global method which mainly estimate the depth of an image that are stored in repository(depth+image) using a nearest-neighbor regression type idea. We demonstrate both the efficacy and the computational efficiency of our methods on numerous 2D images and discuss their drawbacks and benefits. these method lack behind in time computation hence we present a new method support vector machine which increase the time efficiency and calculate the depth map of an image and also use median filter and cross bilateral filters to provide high quality images.**

Keywords—Images,Nearest neighbouring search,SVM,Filters.

I.INTRODUCTION

Rapid development of 3D displays technologies and image has brought 3D into our life. As more facilities and devices are 3D capable, the demand for 3D image and video contents is increasing sharply.However, the tremendous amount of current and past media data is in 2D format and 3D stereo contents are still not rich now.

The availability of 3D-capable hardware today, such as TVs, Blu-Ray players, gaming consoles, and smartphones, is not yet matched by 3D content production. constantly growing in numbers, 3D movies are still an exception rather than a rule, and 3D broadcasting (mostly sports) is still minuscule compared to 2D broadcasting. The gap between 3D hardware and 3D content availability is likely to close in the future, but today there exists an urgent need to convert the existing 2D content to 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 stereopairimages.

The methods we propose in this paper, carry the "big data"philosophy of machine learning. In consequence, they apply toarbitrary scenes and require no manual annotation. Our datadrivenapproach to 2D-to-3D conversion has been inspired by the recent trend to use large image databases for various computer vision tasks, such as object recognition [18] and image saliency detection [19]. In particular, we propose anew class of methods that are based on the radically different approach of learningthe 2D-to-3D conversion from examples.We develop two types of methods. The first one is based onlearning a point mapping from *local* image/video attributes, such as color, spatial position, and motion at each pixel, toscene-depth at that pixel using a regression type idea. Thesecond one is based on *globally* estimating the entire depthmap of a query image directly from a repository of 3D images(image+depth pairs or stereopairs) using a nearest-neighbor regression type idea. Early versions of our learning-based approach to 2D-to-3D image conversion, either suffered from high computational complexity [8] or were tested on only a single dataset [9]. Here, we introduce the local method and evaluate the qualitative performance and the computational efficiency of both the local and global methods against those of the Make3D algorithm [14] and a recent method proposed by Karsch[7]. We demonstrate the improved quality of the depth maps produced by our global method relative to stateof- the-art methods together with up to 4 orders of magnitude reduction in computational effort. We also discuss weaknesses of both proposed methods.

II.EXISTING SYSTEM

A. Semi-Automatic Method

To reduce operator involvement in the process and, therefore, lower the cost while speeding up the conversion,research effort has recently focused on the most labor-intensiv steps of the manual involvement, namely spatial depth assignment Guttman [6] have proposed a dense depth recoveryvia diffusion from sparse depth assigned by the operator.In the first step, the operator assigns relative depth to imagepatches in some frames by scribbling. In the second step, acombination of depth diffusion ,that accounts for local imagesaliency and local motion, and depth classification is applied.In the final step, disparity is computed from the depth field and two novel views are generated by applying half of the disparity amplitude. Phan [12] propose a simplified and more efficient version of the Guttmann et al. [6] method using scale-space random walks that they solve with the help of graph cuts. Liao [10] further simplify operator involvement by first computingoptical flow, then applying structure-from-motion estimationand finally extracting moving object boundaries. The role ofan operator is to correct errors in the automatically computeddepth of moving objects and assign depth in undefined areas.

B. Automatic Method

Several electronics manufacturers have developed real-time 2D-to-3D converters that rely on stronger assumptions andsimpler processing than the methods discussed above, e.g.,-moving or larger objects are assumed to be closer to theviewer, higher frequency of texture is assumed to belong toobjects

located further away, etc. Although such methods maywork well in specific scenarios, in general it is very difficult,if not impossible, to construct heuristic assumptions that cover all possible background and foreground combinations.

The problem of depth estimation from a single 2D image,which is the main step in 2D-to-3D conversion, can beformulated in various ways, for example as a shape fromshadingproblem [20]. However, this problem is severelyunder-constrained; quality depth estimates can be found onlyfor special cases. Other methods, often called multi-viewstereo, attempt to recover depth by estimating scene geometryfrom multiple images not taken simultaneously. For example, amoving camera permits structure-from-motion estimation [17]while a fixed camera with varying focal length permits depthfrom-defocus estimation [16]. Both are examples of the useof multiple images of the same scene captured at differenttimes or under different exposure conditions.

III.PROPOSED SYSTEM

A.LOCAL POINT TRANSFORMATION

The first class of conversion methods we are presenting is based on learning a point transformation that relates local lowlevelimage or video attributes at a pixel to scene-depth at thatpixel. Once the point transformation is learned, it is appliedto a monocular image, i.e., depth is assigned to a pixel based on its attributes. This is in contrast to methods described where the entire depth map of a query is estimated directly from a repository of 3D images (image+depth pairs or stereopairs) using a nearest-neighbor regression type idea.

A pivotal element in this approach is a point transformation used to compute depth from image attributes. This transformationcan be estimated either by training on a groundtruth dataset, the approach we take in this paper, or defined heuristically.

Let $I = \left\{\left(\overrightarrow{I^1},\ d^1\right), \left(\overrightarrow{I^2},\ d^2\right), \ldots, \left(\overrightarrow{I^k},\ d^k\right)\right\}$denote a training dataset composed of K pairs $\left(\overrightarrow{I^k},\ d^k\right)$ where$\overrightarrow{I^1}$,is a color image (usually in *YUV* format) and $d_k$is this a color image (usually in YUV format) and $d_k$is the corresponding depth field. We assume that all images and depth fields have the same spatial dimensions. Such a dataset can be constructed in various ways. One example is the Make3D dataset [13], [14], [21] that consists of images and depth fields captured outdoors by a laser range finder. Another example is the NYU Kinect dataset [15], [22] containing over 100 k images and depth fields captured indoors using a Kinectcamera.

Given a training set I consisting of K image-depth pairs, one can, in principle, learn a general regression function that maps a tuple of local features such as (color,location,motion) to a depth value, i.e.

$$f : (\text{color, location,motion}) \rightarrow \text{depth.}$$

However, to ensure low run-time memory and processing costs, we learn a more restricted form of transformation:

$$f\,[\text{color, x,motion}] = w_c f_c\,[\text{color}] + w_i f_i\,[\text{x}] + w_m w_m\,[\text{motion}].$$

We now discuss how the individual color-depth, locationdepth, and motion-depth transformations as well as the weights are learned.

Fig. 1 shows a sample video frame with depth maps estimated from color, location and motion cues separately, as well as the final combined depth map. In order to obtain a color depth transformation fc.we first transform the YUV space, commonly used in compressed images and videos, to the HSV color space. We found out that the saturation component (S) provides little depth discrimination capacity and therefore we limit the transformation attributes to hue (H) and value (V). Let $[H^k[x], S^k[x], V^k[x]]T$ be the HSV components of a pixel at spatial location x quantized to L levels. The depth mapping fc[h, v], h, v = 1, ..., L is computed as the average of depths at all pixels in I with hue h and value v:

$$f_c[h, v] = \frac{\sum_{k=1}^{K}\sum_{x} 1\left(H^k[x]=h, V^k[x]=v\right) d^k[x]}{\sum_{k=1}^{K}\sum_{x} 1\left(H^k[x]=h, V^k[x]=v\right)} \qquad (1)$$

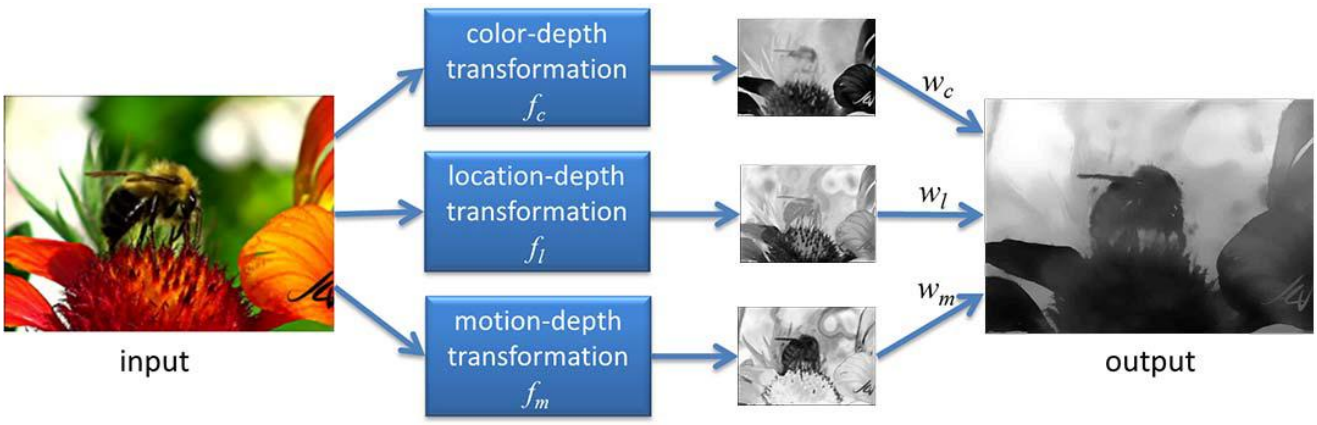


Fig.1. Example of depth estimation from color spatial and location and motion.

where 1*(A)* is the indicator function which equals one if *A* is true and equals zero otherwise.

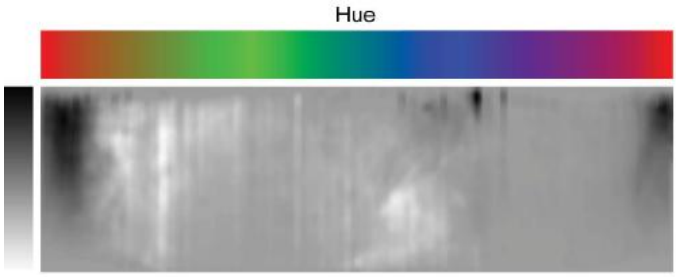


Fig.2. Color-depth Transformation

Fig. 2(a) shows the transformation f_c computed from a dataset $I$of, mostly, outdoor scenes. Note a large dark patch around reddish colors indicating that red elements of a scene are located closer to the camera. A large bright patch around bright-bluish colors is indicative of a far-away sky. The bright patch around yellow-orange colors is more difficult to classify but may be due to the distant sun as many videos have been captured outdoors.The location-depth transformation $f_l$is simply the average depth computed from all depth maps in Iat the same location:

$$f_l[x] = \frac{1}{K}\sum_{K=1}^{K} d^K[x] \qquad [2]$$

In addition to color and spatial attributes, video sequences may contain motion attributes relevant to depth recovery. In this case, local motion between consecutive video frames is of interest. The underlying assumption in the motion-depth transformation is that moving objects are closer to the viewer than the background. In order to estimate the motion-depth transformation f_m, the basic idea is to first compute local motion between consecutive video frames, then extract a moving object mask from this motion, and, finally, assign a distinct depth (smaller than that of the background) to this mask. This brings the moving objects closer to the viewer. The estimation of local motion may be accomplished by any optical flow method, e.g., [2], but may also require global motion compensation, e.g., [5], in order to account for camera movements. A simple thresholding of the magnitude of local motion produces a moving object's mask. However, since such masks are often noisy some form of smoothing may be needed. Cross-bilateral filtering [4] controlled by the luminance of the video frame, in which the estimated local motion is anchored, usually suffices.In the final step, the local transformation outputs are linearly combined to produce the final depth field.

B.GLOBALNEAREST-NEIGHBOR DEPTH LEARNING

While 2D-to-3D conversion based on learning a local point transformation has the undisputed advantage of computational efficiency – the point transformation can be learned off-lineand applied basically in real time – the same transformation isapplied to images with potentially different global 3D scenestructure. This is because this type of conversion, althoughlearning-based, is based on purely localimage/video attributes, such as color, spatial position, and motion at each pixel. To address this limitation, in this section we develop a second method that estimates the globaldepth 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.The approach we propose here is built upon a key observation and an assumption.
The following steps are:

- **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,
- **depth fusion**: combine the *k* representative depth fields,for example, by means of median filtering across depthfields

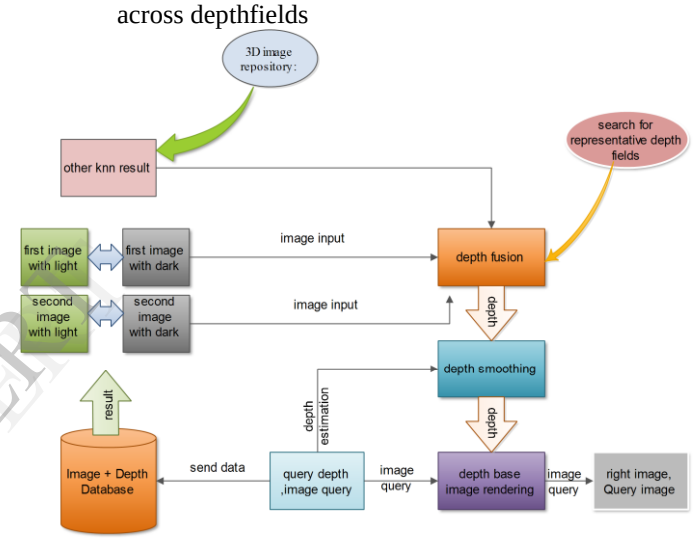


Fig.3. Block diagram of global method.

- **depth smoothing**: process the fused depth field to remove spurious variations, while preserving depth,for example, by means of cross-bilateralfiltering,
- **stereo rendering**: generate the right image of a fictitiousstereopair using the monocular query image and thesmoothed depth field followed by suitable processingof occlusions and newly-exposed areas.directly to 3D images represented as an image+depth pair.However, in the case of stereopairs a disparity field needsto be computed first for each left/right image pair. Then,each disparity field can be converted to a depth map.

1) 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 irrelevant. Images that are not photometrically similar to the 2D query need to berejected because they are not useful for estimating depth(as per our

assumption). Note that although we might misssome depth-relevant images, we are effectively limiting thenumber of irrelevant images that could potentially be moreharmful to the 2D-to-3D conversion process. The selection ofa smaller subset of images provides the added practical benefitof computational tractability when the size of the repository is very large. One method for selecting a useful subset of depth-relevant images from a large repository is to select only the k images that are closest to the query where closeness is measured by some distance function capturing global image properties such as color, texture, edges, etc. As this distance function, we use the Euclidean norm of the difference between histograms oforiented gradients (HOGs) [3] computed from two images.Each HOG consists of 144 real values (4×4 blocks with9 gradient direction bins) that can be efficiently computed.We perform a search for top matches to our monocular query Q among all images$\overrightarrow{I^k}$, $k = 1, ..., K$ in the 3D database$I$. The search returns an ordered list of image+depthpairs,from the most to the least photometrically similar *vis-à-vis* thequery. We discard all but the top k matches (kNNs) from thislist.

Fig. 4 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 is somewhat related to that expected in the query. In Fig. 5.we show search results for two indoor query images (office and dining room) performed on the NYU Kinect dataset. While some of the retained images share local 3D structures with the query image .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 good matches, for smaller databases this may not be true. Therefore, a judicious selection of k is important. We discuss the choice of k. We denote by K the set of indices i of image+depth pairs that are the top k photometrically-nearest neighbors of the query Q.

2D Query: Buildings

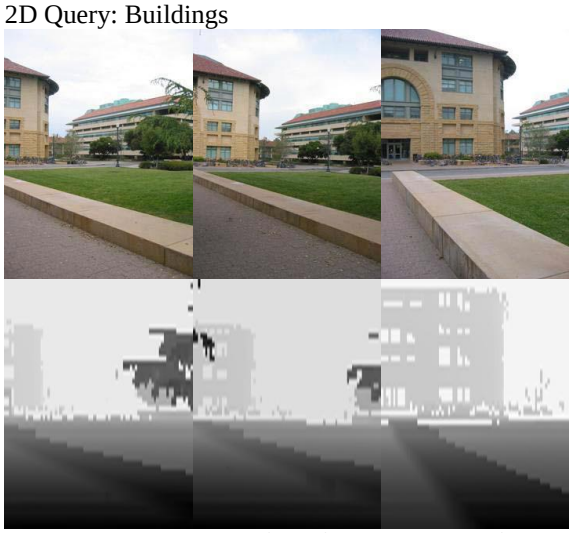


Fig.4. RGB image and depth field of two 2D queries (left column), and their four nearest neighbors (columns 2–5) retrieved using the Euclidean norm on the difference between histograms of gradients.

2D Query :Dining room

Fig. 5. RGB image and depth field of two 2D queries (left column), and their four nearest neighbors (columns 2-5) retrieved using the Euclidean norm on the difference between histograms of gradients.

2) Depth Fusion

In general, none of the NN image+depth pairs (I_i, d_i), $i \in K$ match the query Q accurately (Figs. 4 and 5). However,the location of some objects (e.g., furniture) and parts of the background (e.g., walls) is quite consistent with those intherespective query. If a similar object (e..g, building, table) appears at a similar location in several kNN images, it is likely that such an object also appears in the query, and the depth field being sought should reflect this. We compute this depth field by applying the median operator across the kNN depths at each spatial location $\mathbf{x}$ as follows:

$$d[\mathbf{x}] = \text{median}\{d_i[\mathbf{x}] \quad \forall i \in K\} \quad (3)$$

3) Cross-Bilateral Filtering (CBF) of Depth

While the median-based fusion helps make depth more consistentglobally, the fused depth is overly smooth and locallyinconsistent with the query image due to edge misalignmentbetween the depth fields of the kNNs and the query image.This, in turn, often results in the lack of edges in the fuseddepth where sharp object boundaries should occur and/orthe lack of fused-depth smoothness where smooth depth is expected.In order to correct this, similarly to Agnot[1],we apply cross-bilateral filtering (CBF). CBF is a variant of bilateral filtering, an edge-preserving image smoothing method that applies anisotropic diffusion controlled by the local content of the image itself [4]. In CBF, however, the diffusion is not controlled by the local content of the image under smoothing but by an external input. We apply CBF to the fused depth d using the query image Q to control diffusion. This allows us 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$d$. This is implemented as follows:

$$\hat{d}[\mathbf{x}] = \frac{1}{\gamma[\mathbf{x}]} \sum_{\mathbf{y}} d[\mathbf{y}] h_{\sigma_s}(\mathbf{x}-\mathbf{y}) h_{\sigma_r}(Y[\mathbf{x}]-Y[\mathbf{y}]),$$
$$\gamma[\mathbf{x}] = \sum_{\mathbf{y}} h_{\sigma_s}(\mathbf{x}-\mathbf{y}) h_{\sigma_r}(Y[\mathbf{x}]-Y[\mathbf{y}]), \quad (4)$$

Where $\hat{d}$the filtered depth field and $h_e(\mathbf{x})$=exp(−$\|x\|/2\sigma^2$)/$2\pi\sigma^2$is a Gaussian weighting function.The

directional smoothing of d is controlled by the query image *via* the weight $h_{\sigma_e}(Y\ [\mathbf{x}]-Y\ [\mathbf{y}])$. For largeluminance discontinuities, the weight $h_{\sigma_e}(Y\ [\mathbf{x}]-Y\ [\mathbf{y}])$ is small and thus the contribution of $d[\mathbf{y}]$ to the output is small. However, when $Y\ [\mathbf{y}]$ is similar to $Y\ [\mathbf{x}]$ then $h_{\sigma_e}(Y\ [\mathbf{x}]-Y\ [\mathbf{y}])$is relatively large and the contribution of $d[\mathbf{y}]$ to the output is larger. In essence, depth filtering (smoothing) is happening along (and not across) query edges.

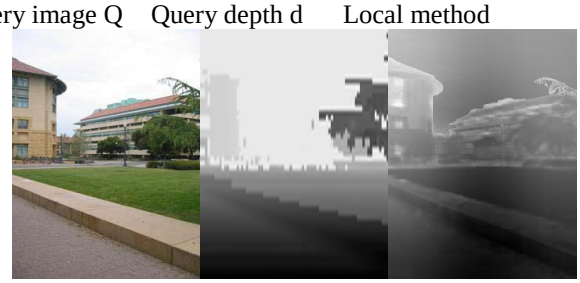
4) *Stereo Rendering*

In order to generate an estimate of the right image QR fromthe monocular query Q, we need to compute a disparity δ fromthe estimated depth $\hat{d}$. Assuming that the fictitious image pair(Q, $\widehat{Q_R}$) was captured by parallel cameras with baseline Band focal length f , the disparity is simply δ[x, y] = B f/ $\hat{d}[\mathbf{x}]$,where $\mathbf{x}=[x,y]^T$.We forward-project the 2D query Q toproduce the right image:

$$\widehat{Q_R}[x+\delta[x,y],y]=Q[x,y] \qquad (5)$$

while rounding the location coordinates $(x+\delta[x,y],y)$ to the nearest sampling grid point. We handle occlusions by depth ordering: if $(x_i+\delta[x_i,y_i],y_i)=(x_j+\delta[x_j,y_i],y_i)$ for some i, j , we assign to the location $(x_i+\delta[x_i,y_i],y_i)$ in Q_R 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 $(x_j,y_i)=(x_i+\delta[x_i,y_i],y_i)$,we apply simple inpainting using inpaint_nans from matlab Central.Applying a more advanced depth-based rendering method would only improve this step of the proposed 2D-to-3D conversion.

Query image Q Query depth d Local method



Global method Make3D

Fig.6.Query images from Fig. 5and depth fields: of the query, depth estimated by the local transformation method, depth estimated by the global transformation method (with CBF) and depth computed using the Make3D algorithm.

In Fig. 6, we show an example of median-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. Fig.7. compares the fused depth before cross-bilateral filtering and after. 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.

Query image Q Query depth d_Q Global(median)

Global(median+CBF) Make3D

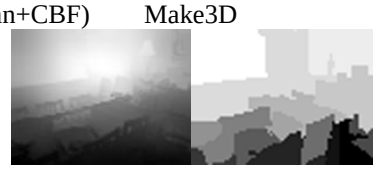


Fig. 7. Query images from Fig. 6 and depth fields: of the query, estimated depth by the global method after median-based fusion and after the same fusion and CBF, and depth computed using the Make3D algorithm.

In order to evaluate the performance of the proposedalgorithms quantitatively, we first applied leave-one-out cross-validation (LOOCV) as follows. We selected one image+depth pair from a database as the 2D query *(Q, dQ)*treating the remaining pairs as the 3D image repository *I*based on which a depth estimate d^ and a right-image estimate$Q^{\wedge}{}_R$are computed. As the quality metric, we used normalizedcross-covariance between the estimated depth $d^{\wedge}$and the ground-truth depth $d_Q$defined as follows:

$$C=\frac{1}{N\sigma_{\hat{d}}\sigma_{d_Q}}\sum(d^{[x]}-\mu_{\hat{d}})(d_Q[x]-\mu_{d_Q}) \qquad (6)$$

where$N$ is the number of pixels in $\hat{d}$and d_Q, $\mu_{d_Q}$and $\mu_{\hat{d}}$are the empirical means of $\hat{d}$and d_Q, respectively, while $\sigma_{\hat{d}}$ and $\sigma_{d_Q}$are the corresponding empirical standard deviations.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).

5) *Support Vector Machine*

The SVM method in general it is a set of labeled sample data in order to classify new sample data. To use SVM, you train the algorithm by providing it with example data that you have grouped into a series of categories. Then, when you provide the algorithm with new, unknown data, it assigns that data to one of your given categories based on its resemblance to the known training data.It mainly distinguish the objects in a given image using HOG and SVM uses a subset of training point also known as support vectors to classify different objects hence it is more efficient and which helps in conversion of 2D to 3D images in less time compare to proposed system i.e local and global methods. we can compute efficient time computation.

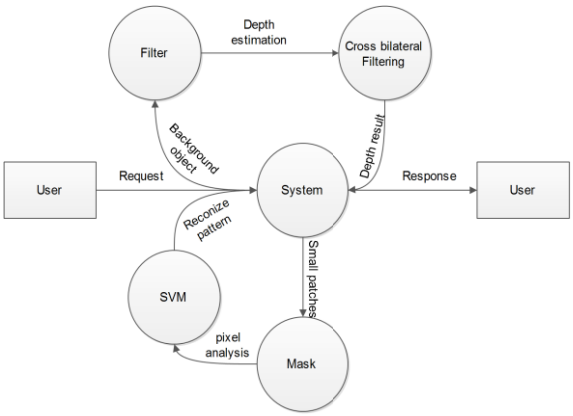


Fig.8. Block Diagram of SVM and filters used for conversion of 2d to 3D images.

The above fig.9.which uses the svm to convert 2D to 3D image using mask and cross bilateral filters.The advantage over local and global methods s during the conversion the time taken by the svm is very less i.e. about 5-6 seconds whereas the global and local takes 10-12 seconds.

IV. Conclusion

We have proposed a new class of methods 2D-to-3D image conversion that are based on the different approach of learning. One method is local point mapping from local image attributes to scene-depth. The second method is based on globally estimating the entire depth field of a query directly from a repository of image +depth pairs using nearest neighbor-based regression. These method overcome the disadvantage of existing system.While the local method perform extremely fast as it is, basically, based on table lookup. However, our global method performed better than the previous method in terms of cumulative performance across two datasets and two testing methods, and has done so at a fraction of CPU time.The support vector machine which provide better time computational efficiency.With the continuously increasing amount of 3D data on-line and with the rapidly growing computing power in the cloud, the proposed framework seems a promising alternative to operator-assisted 2D-to-3D image and video conversion.

References

[1] 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.

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracyoptical flow estimation based on a theory for warping," in Proc. Eur.Conf. Comput. Vis., 2004, pp. 25–36.

[3} C.J.C. Burges. Simplified support vector decision rules.In Lorenza Saitta, editor, Proceedings of the Thirteenth International Conference on Machine Learning, pages 71–77, Bari, Italy, 1996. Morgan Kaufman.

[4] 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.

[5] M.Grundmann,V.Kwatra, and I.Essa, "Auto-directed video stabilizationwith robust L1 optimal camera paths," in Proc. IEEE Conf. Comput.Vis. Pattern Recognit., Jun. 2011, pp. 225–232.

[6] M. Guttmann, L. Wolf, and D. Cohen-Or, "Semi-automatic stereoextraction from video footage," in Proc. IEEE Int. Conf. Comput. Vis.,Oct. 2009, pp. 136–142.

[7] K. Karsch, C. Liu, and S. B. Kang, "Depth extraction from videousing non-parametric sampling," in Proc. Eur. Conf. Comput. Vis., 2012,pp. 775–788.

[8] J. Konrad, G. Brown, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee,"Automatic 2D-to-3D image conversion using 3D examples from theInternet," Proc. SPIE, vol. 8288, p. 82880F, Jan. 2012.

[9] J. Konrad, M. Wang, and P. Ishwar, "2D-to-3D image conversion bylearning depth from examples," in Proc. IEEE Comput. Soc. CVPRW,Jun. 2012, pp. 16–22.

[10 M. Liao, J. Gao, R. Yang, and M. Gong, "Video stereolization:Combining motion analysis with user interaction," IEEE Trans.Visualizat. Comput. Graph., vol. 18, no. 7, pp. 1079–1088, Jul.2012.

[11] B. Liu, S. Gould, and D. Koller, "Single image depth estimation frompredicted semantic labels," in Proc. IEEE Conf. Comput. Vis. PatternRecognit., Jun. 2010, pp. 1253–1260.

[12] R. Phan, R. Rzeszutek, and D. Androutsos, "Semi-automatic 2D to3D image conversion using scale-space random walks and a graphcuts based depth prior," in Proc. 18th IEEE Int. Conf. Image Process.,Sep. 2011, pp. 865–868.

[13] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from singlemonocular images," in Advances in Neural Information ProcessingSystems. Cambridge, MA, USA: MIT Press, 2005.

[14] A. Saxena, M. Sun, and A. Ng, "Make3D: Learning 3D scene structurefrom a single still image," IEEE Trans. Pattern Anal. Mach. Intell.,vol. 31, no. 5, pp. 824–840, May 2009.

[15] N. Silberman and R. Fergus, "Indoor scene segmentation using a structuredlight sensor," in Proc. Int. Conf. Comput. Vis. Workshops,Nov. 2011, pp. 601–608.

[16] M. Subbarao and G. Surya, "Depth from defocus: A spatial domainapproach," Int. J. Comput. Vis., vol. 13, no. 3, pp. 271–294,1994.

[17] R. Szeliski and P. H. S. Torr, "Geometrically constrained structure frommotion: Points on planes," in Proc. Eur. Workshop 3D Struct. MultipleImages Large-Scale Environ., 1998, pp. 171–186.

[18] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images:A large data set for nonparametric object and scene recognition," IEEETrans. Pattern Anal. Mach. Intell., vol. 30, no. 11, pp. 1958–1970,Nov. 2008.

[19] 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.

[20] R. Zhang, P. S. Tsai, J. Cryer, and M. Shah, "Shape-from-shading:A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 8, pp. 690–706, Aug. 1999.

[21] (2012).Make3D[Online].Available: http://make3d.cs.cornell.edu/data.html

[22] (2012).NYUDepthV1[Online].Available:http://cs.nyu.edu/~silberman/datasets/nyu_depth_v1.html