2D TO 3D GRAPHIC OBJECT CONVERSION

Abhishek Dhilpe , Pooja Kangane, Kaushal Nile, Saurav Deshmukh
DEPARTMENT OF INFORMATION TECHNOLOGY,
Sandip Institute of Technology and Research Centre, Nashik, Maharashtra, India.

Abstract : *The technique of converting 2D to 3D is critical in the creation and promotion of three-dimensional television(3DTV) since it has a sufficient supply of high-quality 3D programme content. A novel automated 2D to 3D conversion approach based on multi-depth cues is described in this paper. Perspective geometry, defocus, visual saliency, and adaptive depth models are among the depth cues used in our scheme, which will be combined into a single depth map based on the types of 2D scenes. The initial 2D image or video is transformed to stereoscopic for viewing on 3D display devices after the depth maps are removed.*

IndexTerms

1. INTRODUCTION

In image/object generation problems, convolutional neural networks (ConvNets)-based models have achieved state-of-the-art results. Variational autoencoders and generative adversarial networks are two of the class's most well-known works, both of which have had a lot of success in a variety of applications. The study of generative modelling on 3D data using similar frameworks has gained growing interest in light of the recent launch of wide publicly accessible 3D model repositories.

3D object models can be represented in a variety of ways in computer vision and graphics. As a result, triangular meshes and point clouds are common due to their vectorized (and thus scalable) data representations and compact encoding of shape information, which can also include texture. However, since the dimensionality per 3D shape sample can differ, this efficient representation has an inherent downside, making the application of learning methods problematic. Furthermore, since Euclidean convolutional operations cannot be directly implemented, such data representations do not elegantly fit within traditional ConvNets. Until now, most 3D model generation methods have relied on volumetric representations, which enable 3D Euclidean convolution to operate on standard discretized voxel grids. For both discriminative and generative problems, 3D ConvNets (as opposed to the traditional 2D form) have been successfully applied to 3D volumetric representations.

Despite their recent popularity, 3D ConvNets have a flaw when modelling shapes that have volumetric representations. Unlike 2D images, which contain meaningful spatial and texture information in every pixel, volumetric representations are sparse in information. To put it another way, the most detailed knowledge about shape representations is found on the surface of a 3D structure, which accounts for just a small portion of all voxels in an occupancy grid. As a result, trying to predict the huge quantifiable volume of useless 3D ConvNets data wastes many high-complexity 3D computation and memory turns, severely reducing the granularity of 3D volumetric forms that can even be modelled upon high-end GPU nodes widely used in research.

We propose an efficient framework for representing and generating 3D object shapes with dense point clouds in this paper. This is accomplished by learning to predict 3D structures from different perspectives, which is then combined and optimised using 3D geometric reasoning. We use 2D convolutional operations to predict points clouds that form the surface of 3D objects, as opposed to prior art that uses 3D ConvNets to work on volumetric data. Our findings show that we are able to produce much denser and more precise shapes than current 3D prediction methods. Our contributions are summarized as follows:

1. We suggest that 2D ConvNets can generate dense point clouds that form the surface of 3D objects in a 3D space.
2. To function as a differentiable approximation of true rendering, we add a pseudo-rendering pipeline. The pseudo-rendered depth images are often used for 2D projection optimization when studying dense 3D shapes.
3. On single-image 3D reconstruction problems, we show that our approach outperforms state-of-the-art methods by a large margin.

We suggest using a structure generator based on 2D convolutional operations to predict the 3D structure at N viewpoints from an encoded latent representation. The point clouds are joined by converting each viewpoint's 3D structure to canonical coordinates. The pseudo-renderer creates depth images from new perspectives, which are then used to refine joint 2D projections. There are no learnable parameters in this, and the reasons are solely dependent on 3D geometry.

2. LITERATURE SURVEY

2.1 Automatic 2D to 3D Conversion

2.1.1 Learning-Based, Automatic 2D-to-3D Image and Video Conversion

Despite tremendous development in recent years, 3D material is still overshadowed by its 2D counterpart in terms of availability. Many 2D-to-3D image and video conversion technologies have been developed to bridge this gap. Human-operated solutions have shown to be the most effective, but they are also the most time consuming and expensive. Automatic approaches, which often use a deterministic 3D scene model, have yet to achieve the same degree of quality because they are based on assumptions that are frequently broken in practice.

In this, we propose a new class of methods that are based on the radically different approach of learning the 2D-to-3D conversion from examples. We develop two types of methods. The first is based on learning a point mapping from local image/video characteristics such as colour, spatial position, and, in the case of video, motion at each pixel to scene-depth at that pixel using a regression technique. The second technique uses a nearest-neighbor regression to estimate the whole depth map of a query image straight from a repository of 3D pictures (image + depth pairs or stereo pairs). On a variety of 2D photos, we

illustrate the efficacy and computational efficiency of our approaches, as well as analyse their flaws and benefits. Despite the fact that our results are far from ideal, they show that repositories of 3D content can be used to convert 2D to 3D images effectively. By guaranteeing temporal continuity of computed depth maps, an instantaneous extension to video is possible.

The difficulty of predicting depth from a single 2D image, which is the initial step in converting 2D to 3D, can be modelled as a shape-from-shading problem. On the other hand, this issue is quite limited; quality depth estimations can only be obtained in uncommon circumstances. Other techniques, such as multi-view stereo, try to recover depth by estimating scene geometry from numerous images that were not acquired at the same time. A moving camera, for example, can estimate structure from motion, but a fixed camera with variable focal length can estimate depth from defocus.Both are examples of the use of multiple images of the same scene captured at different times or under different exposure conditions (e.g., all images of the Statue of Liberty). Although such approaches are similar in spirit to the methods provided here, the fundamental distinction is that we use all photographs available in a huge library and automatically choose acceptable ones for depth recovery, whereas these approaches employ photos known to portray the same scene as the query image.

2.1.2 Stereo Correspondence and Disparity Maps

A review of the literature on transforming traditional monocular video sequences, or those captured by a single camera, into two or more views of the same scene that can be viewed using end-to-end 3D technology. This type of conversion can be broken down into many groups, and we'll go through the most applicable and up-to-date methods for each of them. We'll go over these approaches in-depth, citing works from the last few years as examples. Color information, edge information, techniques from depth-based image rendering (DIBR), motion, and analyzing scene features are the categories that 2D to 3D conversion can be broken down into. However, the two most popular methods are motion estimation and scene feature analysis.

2.1.3 Methods for Converting 2D to 3D Using Motion

The use of motion estimation to calculate the depth or disparity of the scene is the first of the most popular techniques for 2D to 3D conversion. The basic principle is that objects that are closer to the camera should move faster, while objects that are farther away should move slower. As a result, motion estimation can be used to evaluate the correspondence between two consecutive frames, as well as the necessary pixel shifts from the reference view (current frame) to the target view (next frame). However, the approaches differ in how they use motion vectors to create a stereoscopic or multiview video series, but the underlying mechanism is the same. We'll now start talking about how to use motion estimation for 2D to 3D conversion.

We begin with the method which focuses on real-time conversion from 2D to 3D. This work is arguably one of the methods that have proven that one of the main features to use when deciding the depth map is motion estimation. They primarily employ motion vectors from the MPEG4 decoder. They calculate the magnitude of the motion vector for each pixel by taking the Euclidean distance between the horizontal and vertical motion estimation components. Each frame in the video sequence in question is decomposed into its RGB components. The magnitude of the motion vector is calculated concurrently. Anaglyph images are generated for display by using the red channel of each frame and the motion vector magnitudes to move pixels to produce the correct image. Each frame's anaglyph image is created by combining the original image and the changed red channel.

Similarly, Huang at all. compute depth maps by combining motion and scene geometry. The motion vectors from the H.264 decoder are specifically used to produce a motion-based depth map. A moving object detection algorithm is also implemented to reduce the block effect induced by motion estimation in H.264. Following that, Gaussian mixture models are used to evaluate the objects in the foreground and to adjust the initial motion-based depth map with the H.264 decoder. They remove vanishing lines and vanishing points from the scene's geometry using edge detection and the Hough transform.

Finally, Yan et al. apply the scale-invariant feature transform (SIFT) between two consecutive frames before formulating a homography transformation. In addition, the mean-shift segmentation algorithm is used to over-segment the current frame. The over segmented result and homography transformation is then formulated into a graph cuts segmentation problem, and the depths are solved in this way Homography estimation is used to identify motion layers or layers in a scene that move in the same general directions. The depth of motion vectors in the same motion layer is most likely the same. As a result, the previous data is used to perform a region-based graph cuts segmentation. The end result is a depth assignment that is consistent across all motion layers.

2.1.4 Edge Information and Colour Cues

To segment objects in a scene, the next group of methods uses edge information and other colour cues. Color information is used for segmentation and as a depth cue on its own. Edge information is usually used to find complexity in items, while colour information is used for segmentation and as a depth cue on its own. To construct a depth map, Chang et al. combine edge knowledge with a depth gradient hypothesis. The depth map is then filtered with a cross bilateral filter to smooth the depth map's edge boundaries. DIBR is used to create stereoscopic views that are synthesised.

Zhang et al use two cues for depth map generation and fuse them together to create a final result, which is similar to this work. The dark-channel prior is the first signal, and it shows the distance between objects and the camera due to variations in ambient light. The second cue is dependent on the movement of the object. Both the dark channel and motion priors result in holes. Morphological opening and closing operations are used to address this.

To find the depth map of an image, Chang et al use edge detection and K-means segmentation. To find motion information in a scene, edge detection and registration are used. This data is combined with the effects of K-means segmentation and over-segmentation, which are performed using a variety of boundary conditions and thresholds.

2.1.5 Depth from Focus

Depth from focus is focused on the premise that objects in focus are foreground objects in certain scenes, while objects out of focus are background objects. These techniques work well for a specific subset of videos and photographs where this particular shooting technique is used, but this might not always be the case in a full-length film. To extract depth, they use focus information directly from an image. Gaussian filters with various parameters are used to remove objects of various focus.

To detect focus, these use information from the wavelet domain. The depth map is generated by looking at the high frequency components of an image's wavelet domain in YUV colour space. High frequency components reflect oriented objects and could thus be useful for locating foreground objects. The depth map that is generated is binary in nature.

2.1.6 Other Techniques of conversion

The remaining research on automated 2D to 3D conversion does not actually fall into one of the other categories, but rather covers a wide range of topics.

This paper presents a low-cost embedded auto-stereo game conversion device. Depth is achieved by removing secret information that was previously used to create the original scene. To speed up processing, software and hardware components are used.

Lie et al. suggest using motion compensation and a trilateral filter, as well as a feedback loop to optimise the results, to spread the depth map of a mainframe to other frames. Although it is believed that the original depth map was created by hand, the technique could be used in conjunction with other methods for creating depth maps automatically or semi-automatically.

Wu et al. extract corner points for monitoring using bidirectional optical flow, and then segment artefacts using the Mean Shift Algorithm. By creating a bounding rectangle around each object and comparing it to the bounding box of the same object in a previous frame, depth values are allocated.

2.2 Semi-Automatic 2D to 3D Conversion

The other subset of 2D to 3D conversion methods is known as Semi-Automatic conversion, which gets its name from the fact that the conversion process involves some user interactions. This interaction can take the form of output correction, prior knowledge, or something else where the algorithm needs more information from the user. Despite its limited size, this subset has recently sparked interest as researchers discovered that there are no completely automated conversion methods that can do a good enough job on large scale outputs.

B.Ward Depth Director, which uses traditional structure from motion and Graph Cuts combined with over-segmentation to produce an initial depth map, is an example of previous work in the field of semi-automatic 2D to 3D conversion. After that, the user can change the depth regions and/or add depth templates by interacting with them. Other approaches for user-assisted segmentation use Graph Cuts. The authors use over-segmentation to speed up the migration to the mobile space in this paper.

Many user-assisted conversion strategies include the concept of using user-provided strokes. User labelling allows users to communicate with images in a natural way; labels simply refer to things that users see and want to segment. The labelling can also be used to assign the depth value to the segmentation, thereby serving a dual purpose, and this study builds on these two approaches.

3. Proposed System

A system that converts 2d image into 3d object/point cloud with removal of colliding points in point cloud and reconstructing 3d models.

Firstly, structure generator viewpoints are produced in vast quantities then all these viewpoints are then given as an input to point cloud fusion to form a basic 3d diagram which may not be accurate. So to overcome this hurdle an image rendering mechanism comes into place.

Then the 3d model produced is given as an input to image rendering model where all the novel viewpoints are filtered further for better 3d object generation.

Fig.1. input and desired output.

4. Approach

We'll illustrate how to use standard 2D ConvNet to learn prior shape knowledge while combining the benefits of Point Cloud compact representation. Our aim is to create 3D predictions that use dense point clouds to compactly form the surface geometry.This model's clever trick is to combine the fusion and pseudo-rendering modules. Combining the 3 modules together, we obtained an end-to-end model that learns to generate a compact point cloud representation from one single 2D image, using only a 2D convolution structure generator.

4.1 Structure Generator

We will build a standard 2D CNN Structure Generator that learns the prior shape knowledge of an object. With CNN, it is not possible to learn a point cloud directly. Therefore we will instead learn the mapping from a single image to multiple 2D projections of a point cloud. The structure generator predicts the object's 3D structure from N different perspectives. Due to their strong local spatial dependencies, pixel values in natural images can be synthesised using convolutional generative models; related phenomena can be observed when treating point clouds as (x, y, z) multi-channel images on a 2D grid. For volumetric predictions, this method eliminates the need for time-consuming and memory-intensive 3D convolutional operations.

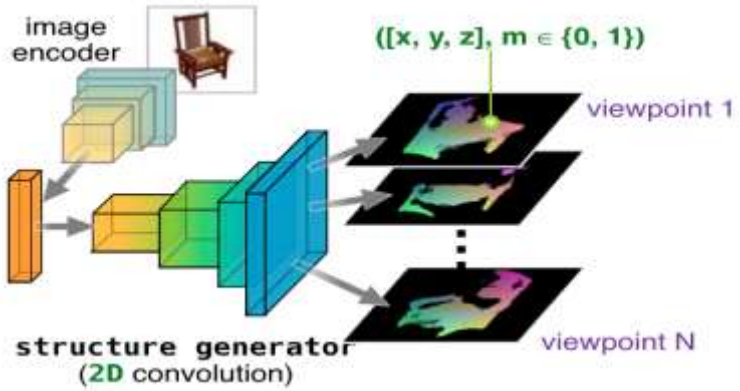

Fig.2. structure generator

1. Input: Single RGB image
2. Output: 2D projections at predetermined viewpoints.

4.2 Point Cloud Fusion

A point cloud is a collection of data points that are arranged in space. A 3D shape or object is represented by the points. Each point has its self x, y, and z coordinates. Volumetric data can also be represented using point clouds. Point clouds are an excellent representation for level-of-detail methods. Since they lack topology details, changing the detail level is as simple as adding or removing points. No expensive topology knowledge changes, such as those needed for progressive meshes, are required.

When you have scans of the real world (laser or photogrammetry), your algorithm first produces a bunch of points which are used for meshing. Nowadays, different point-based rendering approaches exist. The most prominent ones are based on locally approximating the surface by fitting planes or polynomial surfaces to a subset of neighboring points. Technically,these are still (locally) surface reconstructions.

Convert the expected 2D projections to 3D image space. This is possible because the viewpoints of these predictions are fixed and known beforehand.

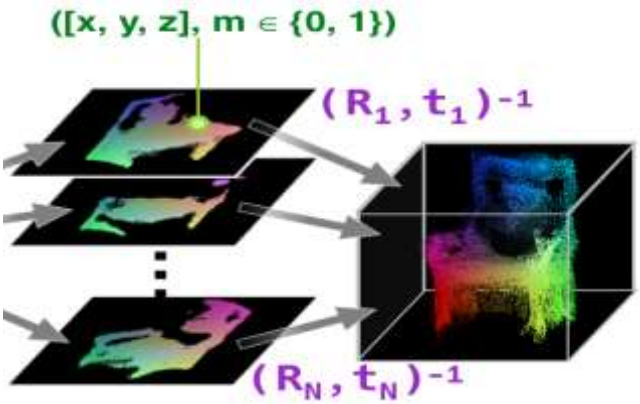

Fig.3. point cloud generation

1. Input: 2D projections at predetermined viewpoints.
2. Output: Point cloud

4.3 Image Renderer

If the Point Cloud fused from the predicted 2D projections are of any good, then if we rendered different 2D projections from new viewpoints, it should resemble the projections from the ground truth 3D model too.

This would mean that the 3D model generated after Image Renderer should resemble the 3D model generated from Point Cloud. This resemblance will give us the idea about the correctness of our predicted Point Cloud.

Multiple transformed 3D points in the image space can correspond to projection on the same pixels. If (x, y) were explicitly discretized, collision would be very likely. Increasing the accuracy of the projection positions by up-sampling the target image reduces the collision effect.

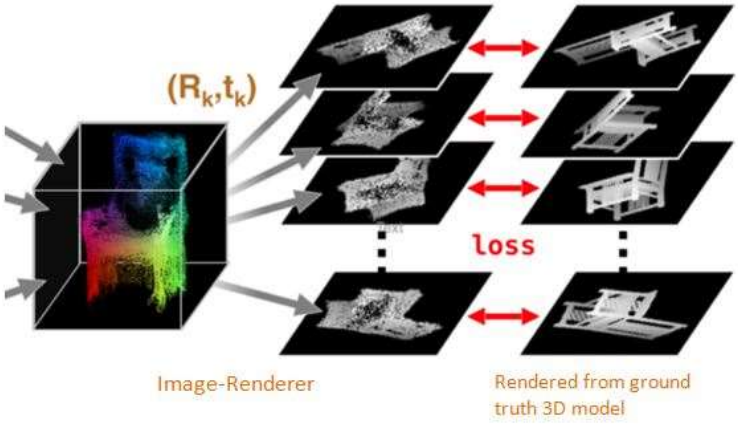

Fig.4. image rendering

- Input: Point cloud
- Output: depth images at novel viewpoints.

4.4 Network Architecture

Combining the 3 modules together, we obtained an end-to-end model that learns to generate a compact point cloud representation from one single 2D image, using only a 2D convolution structure generator.

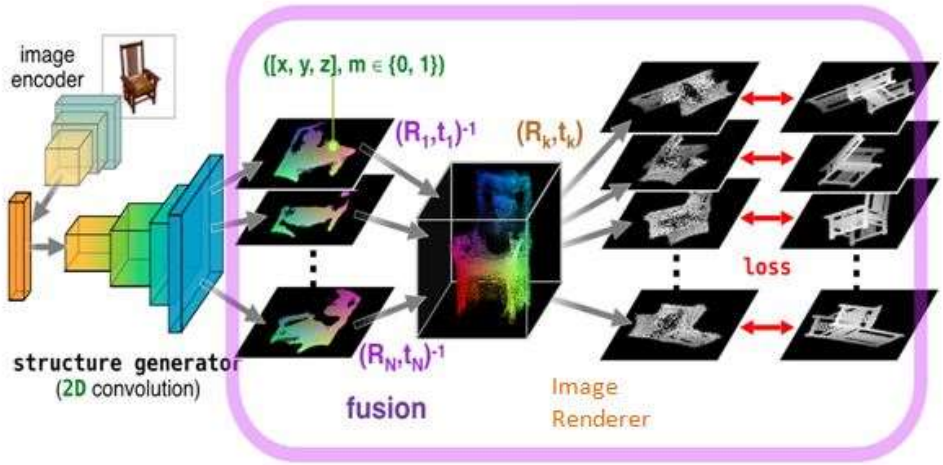

Fig.5. network architecture

4. Conclusion

In this paper, we introduced a technique which performs 2d to 3d conversion operation in an efficient and accurate way. We showed that collision of points in space does make a major difference which reduces the value of accuracy, removal of those colliding points increases the accuracy and makes it more similar to the object. Based on the technique we studied, we can conclude that conversion of 2D Planer image to 3D planer object is possible by comparing viewpoints of 2D objects and novel viewpoints of 3D objects Predict.

REFERENCES

[1] Qi, Charles R and Su, Hao and Mo, Kaichun and Guibas, Leonidas J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. arXiv preprint arXiv:1612.00593, 2016.

[2] Ludovic J. Angot, Wei-Jia Huang and Kai-Che Liu :A 2D to 3D video and image conversion technique based on a bilateral filter,2015.

[3] Xun Cao, Zheng Li, and Qionghai Dai: Semi-Automatic 2D-to-3D Conversion Using Disparity Propagation,IEEE TRANSACTIONS ON BROADCASTING, 2011.

[4] Fredo Durand and Julie Dorsey Laboratory for Computer Science, Massachusetts Institute of Technology: Fast Bilateral Filtering for the Display of High-Dynamic-Range Images, Association for Computing Machinery New York, NY, United States, 2002.

[5] 1Mujawar A. R., 2Nanaware J.D. : DEVELOPMENT AND IMPLEMENTATION OF AUTOMATIC 2D TO 3D IMAGE CONVERSION SYSTEM,International Journal of Advance Research in science and Engineering,2016.

[6] Moshe Guttmann, Lior Wolf, Daniel Cohen: Semi-automatic Stereo Extraction from Video Footage,International Conference on Computer Vision,978-1-4244-4419-9, 2009,

[7] Miao Liao, Jizhou Gao, Ruigang Yang: Video Stereolization: Combining Motion Analysis with User Interaction, IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, 0730-0301, 2012.

[8] Chen-Hsuan Lin, Chen Kong, Simon Lucey: Learning Efficient Point Cloud Generation for Dense 3D Object Reconstruction, The Robotics Institute Carnegie Mellon University, 2017.

[9] Harini S: Automatic Image Conversion From 2D to 3D Using Support Vector Machine, International Journal of Engineering Research & Technology,ISSN: 2278-0181 2016.

[10] Paul Rosenthal, Lars Linsen: Image-space Point Cloud Rendering, ResearchGate, 2015.