AUTOMATIC 2D TO STEREOSCOPIC VIDEO CONVERSION FOR 3D TVS

Xichen Zhou†, Bipin C. Desai, Charalambos Poullis†

Immersive and Creative Technologies Lab†
Department of Computer Science and Software Engineering
Concordia University

ABSTRACT

In this paper we present a novel technique for automatically converting 2D videos to stereoscopic. Uniquely, the proposed approach leverages the strengths of Deep Learning to address the complex problem of depth estimation from a single image. A Convolutional Neural Network is trained on input RGB images and their corresponding depths maps. We reformulate and simplify the process of generating the second camera's depth map and present how this can be used to render an anaglyph image. The anaglyph image was used for demonstration only because of the easy and wide availability of red/cyan glasses however, this does not limit the applicability of the proposed technique to other stereo forms. Finally, we present preliminary results and discuss the challenges.

Index Terms — stereoscopic video, 2d-to-3d conversion, 3D TV content

1. INTRODUCTION

Videos have been around for more than a century; the earliest surviving film recording is from 1888 and it shows traffic crossing the Leeds bridge in England. Since then a vast amount of videos have been recorded, including amateur recordings as well as professional films, the majority of which were filmed using a single camera.

Recently the enormous progress in the field of virtual reality and in particular the release of affordable VR headsets e.g. Google cardboard, Facebook Occulus, HTC Vice, Samsung Gear, etc has made it possible for the general public to directly experience and interact with 3D content. Stereo video cameras are starting to become popular for recording 3D videos and many recent films have already used them. However, videos captured with a single camera cannot be easily converted to 3D. A limited number of human-in-the-loop tools [evil twin, etc] are being used for converting a 2D video into 3D but require extensive human interaction which makes the process time-consuming and expensive. An attempt in automating this process was done by Google's Youtube with the "3D converter" which briefly appeared as new functionality but has now been removed probably because of the poor performance. A handful of other solutions have been proposed which we will review in section 2 which rely on depth, shading and lighting and cues in the video's frames.

In this paper we address the complex problem of automatically converting a video filmed with a single camera to stereoscopic content tailored for viewing in a VR environment.

Similar to other techniques we focus on creating a plausible depth interpretation of the scene that is used to generate the stereoscopic video which does not cause depth perception inconsistencies to the viewer; as opposed to first calculating an accurate 3D reconstruction of the original and then using this to generate the stereoscopic video. The proposed approach employs a Convolutional Neural Network which given an RGB image generates a depth map. A parallel-optical axes stereo setup is assumed and the depth map is used to generate the depth map for the second camera. Using the two depth maps and the disparities between the pixels, an anaglyph image is then rendered. Common artifacts due to rendering from depth are addressed using in-painting. The proposed technique has been tested on several videos and the results are reported.

The paper is organised as follows: Section 2 provides a brief overview of the state-of-the-art in the area. A technical overview of the proposed approach is presented in Section 3. In Section 4 we present in detail the architecture of the CNN used to generate a depth map from RGB images, and Section 5 describes how the anaglyph images are generated based on a depth map. Finally, in Section 7 we discuss the strengths and weaknesses of this approach and identify possible future work.

2. RELATED WORK

Below we provide a brief overview of the state-of-the-art in the area of depth estimation from images and/or videos.

Rendering from a novel viewpoint given a single image is a hot topic and an open research problem in computer vision. The inherent difficulty lies in the fact that recovering the depth from a single image/video and then computing disparities and depth map for a second camera is an ill-conditioned problem. One of the earlier works in the area was in [1] where it was shown how given depth information for a single view a second view can be generated. Since then, depth based rendering became popular. However, existing established methods for extracting depth information require the use of binocular cues i.e. multiple views. In particular, within the context of accurate 3D reconstructions, a plethora of different methods have already been proposed.

A popular methodology for dealing with multiple views of a scene is Structure-from-Motion (SfM). SfM has been successfully applied usually as a first step in the 3D reconstruction of large-scale areas [2] from a sequence of images or videos. In SfM, the inherent assumptions are that the scene remains static and the motion between the cameras capturing the scene is sufficiently large. Therefore, videos or films containing dynamic scenes/changes, or in which the motion between the cameras is minimal cannot be reconstructed.

It has been observed that given an image of a scene, the human visual system is able to extract both semantic and metric information, therefore it should be possible to estimate the depth from a single image. In [3] the authors present one of the first attempts in estimating 3D models from a single monocular image. They proposed an Markov Random Field model for inferring 'plane parameters', combined with supervised learning techniques in order to reconstruct 3D scenes from one image only. More recently, Karsch et al [4] extended this method to videos with no assumptions on the scene being static or the motion between the cameras. Instead, their 'depth-transfer' method recovers depth information from similar scenes contained in a database for which depth values have been previously recorded. They made the assumptions that

"the distribution of depth is comparable among similar scenes". Specifically, the process begins by finding candidate matches between the input image and images in the database using GIST descriptor [5] and k-nearest neighbours (kNN). SIFT-flow [6] was then used as a second step for transferring the depth, followed by an optimization to smooth over the transferred depth.

Several other depth-transfer variants have been proposed with the most recent proposed by Kong et al [7]. These methods share an almost identical pipeline and also proceeds with the candidate matching, the depth transfer, and the optimization for smoothing the results.

Recently, single-shot solutions have also been proposed. These rely on training Convolutional Neural Networks (CNNs) to generate the depth maps directly. One such approach is proposed in [8] where a CNN is trained to learn a potential function consisting of unary depth error and adjacency superpixel depth error. As it is common with neural network based approaches, the success directly depends on whether a similar image to the input image has been used during the training.

3. TECHNICAL OVERVIEW

Figure 1 shows an overview of the proposed system. The input is a frame from a video. We feed-forward the input to a CNN framework which produces a low resolution depth map. Next the depth map becomes the input to the stereo rendering pipeline where a second depth map is generated corresponding to the second camera's view. Finally, an in-painting procedure eliminates the artificats introduced during the rendering of the depth map and a perceptually plausible anaglyph image is created.

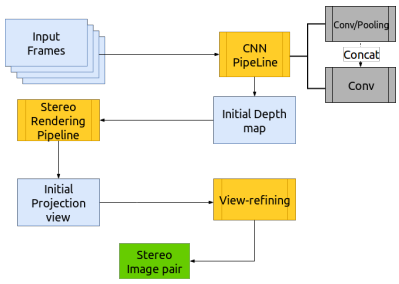

Figure 1: Pipeline of our work

4. CNN MODEL ARCHITECTURE

In this section we briefly describe the network architecture which was used to render the depth and normal maps. We used a network architecture proposed in [9], where multiple scales are concatenated in a coarse-to-fine fashion in order to archive better resolution. The network is trained using RGB and depth map pairs using a loss function for the depths given by

$$L_{depth}(D, D^{\star}) = \frac{1}{n}\sum d_i^2 + \frac{1}{2n}\left(\sum d_i^2\right) + \frac{1}{n}\sum |\bigtriangledown d_{x,y}| \tag{1}$$

which measures the difference between the generated and ground-truth depth, and also the gradient of the depth. The gradient of the depth encourages local structure similarities. Similarly the loss function for the normals is defined as,

$$L_{normal}(N, N^{\star}) = -\frac{1}{n} N \cdot N^{\star} \tag{2}$$

which is equivalent to the cosine proximity.

In summary, the network learns to predict coarse depth maps and then refine the prediction through multiple scales. At each scale there is a number of convolution layers and pooling layers. A scale refers to the resolution of the final output which is achieved either by using larger kernels or deeper layers. A crucial aspect of this network architecture is the concatenation of the output of previous scale to the input of the following scale in order to add additional depth of channels for convolution. The second and third row in Figure 3 show the output of this process, namely the depth and normals respectively.

5. STEREO VIEW RENDERING

The depth map generated using the aforementioned CNN is used to produce an anaglyph image. Below we describe the mathematical formulation for first creating a depth map for the second camera and then generating stereo from the pair of depth maps.

5.1. Stereo Camera Projections

Given a camera with internal parameters defined by camera matrix K and camera pose defined by matrix M, the projection $u = [x, y]$ of a 3D point $P = [X, Y, Z]$ onto the camera's image plane is given by,

$$\lambda_1 u = K \times M \times P \tag{3}$$

where λ is the normalized depth in homogeneous space.

The matrix K is a 3×3 upper-triangular matrix which describes the camera's internal parameters which include focal length f, principal point $[a, b]$, pixel aspect ratio [commonly set to $p = 1$], and skew [commonly set to $s = 0$. The matrix K is defined as,

$$K = \begin{bmatrix} p \times f & s & a \\ 0 & p \times f & b \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The above equation does not account for the effects of distortions occurring due to the optics of the camera.

The camera pose matrix M captures the position and orientation of the camera and is defined as,

$$M = \begin{bmatrix} R_{3x3} & -R_{3x3} \times t_{3x1} \\ 0 & 1 \end{bmatrix} \tag{5}$$

where R is a 3×3 rotation matrix and the vector t is the position of the world's origin with respect to the camera coordinates.

In a stereo setup projecting a 3D point $P = [X, Y, Z]$ results in two projections in each of the cameras $u_1 = [x_1, y_1]$ to $u_2 = [x_2, y_2]$ which using equation 3 give,

$$\lambda_2 u_2 = K \times R^{'} \times B \times R^{-1} \times K^{-1} \times \lambda_1 u_1 \tag{6}$$

where R' is the second camera's rotation expressed as a 3×3, B is the baseline i.e. the translation vector from the first camera to the second camera. Given the fact that the second camera has the same rotation matrix as the first camera [since it is a parallel optical-axes setup] the equation 6 can be further reduced to,

$$\begin{aligned} R_{3x3} \times B_{3x1} \times R_{3x3}^{-1} = \\ = \begin{bmatrix} R_{3x3} & 0_{3x1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_{3x3} & -t_{3x1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{3x3}^{-1} & 0_{3x1} \\ 0 & 1 \end{bmatrix} = \\ = \begin{bmatrix} I_{3x3} & -R_{3x3} \times t_{3x1} \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{7}$$

Substituting equation 7 into equation 6 gives,

$$\lambda_2 u_2 = \begin{bmatrix} f & 0 & a \\ 0 & f & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{3x3} & -R_{3x3} \times t_{3x1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{f} & 0 & -\frac{a}{f} \\ 0 & \frac{1}{f} & -\frac{b}{f} \\ 0 & 0 & 1 \end{bmatrix} \lambda_1 u_1 \tag{8}$$

Substituting λ_1 with the image depth value of each pixel and carrying all the multiplications results in equation 9 given by,

$$\lambda_2 u_2 = \begin{bmatrix} f & 0 & a \\ 0 & f & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \frac{x_1 - a}{f} - Rt_x \\ \lambda_1 \frac{y_1 - b}{f} - Rt_y \\ \lambda_1 - Rt_z \end{bmatrix} = \begin{bmatrix} \lambda_1(x_1 - a) - fRt_x + a \cdot (\lambda_1 - Rt_z) \\ \lambda_1(y_1 - b) - fRt_y + b \cdot (\lambda_1 - Rt_z) \\ \lambda_1 - Rt_z \end{bmatrix} \quad (9)$$

This equation provides a measure for the depth at a pixel in the second stereo camera given the depth value of the pixel in the first i.e. $\lambda + Rt$. In a parallel-optical axes stereo setup the second camera is positioned relative to the first camera's orientation which means that the baseline extends along the x axis in the first camera's coordinate system. Given the fact that the rotation matrix $R_{3\times3}$ forms the basis of the camera coordinate system, and since the translation $t_{3\times2}$ is orthogonal to the $Y - Z$ plane of the rotation matrix we get,

$$R_{3\times3}t_{3\times1} = \begin{bmatrix} \vec{Rx}^t \\ \vec{Ry}^t \\ \vec{Rz}^t \end{bmatrix} \times c\vec{x} = [c, 0, 0]^t \quad (10)$$

where $\vec{R.}^t$ is a transposed 3-vector for each of the axes and c is the camera offset i.e. baseline. Using equation 10 and substituting in 9 results in,

$$\lambda_2 u_2 = \begin{bmatrix} \lambda_1(x_1 - a) - fc + a \cdot (\lambda_1 - 0) \\ \lambda_1(y_1 - b) - f \cdot 0 + b \cdot (\lambda_1 - 0) \\ \lambda_1 + 0 \end{bmatrix} \quad (11)$$

which further implies,

$$u_2 = \begin{bmatrix} x_1 - \frac{fc}{\lambda_1} \\ y_1 \\ 1 \end{bmatrix} \quad (12)$$

The validity of equation 12 can be confirmed by verifying that the epipolar line appears horizontal when the camera motion is horizontal, which is indeed the case. This particular formulation proposed, of deriving the depth for the pixel in the second camera with respect to the depth in the first camera, eliminates the need for camera calibrations. In the following section we discuss the results produced with this technique.

5.2. Stereo Rendering

Equation 12 provides a convenient method of deriving the depth for the second camera given the focal length f and the camera offset c. This produces a depth map for the second camera which can be used to generate the anaglyph stereoscopic image. An important aspect of this process is that the drastically increasing or decreasing the focal length and camera offset results in small or large disparities between the depth maps which when used to generate a stereo image, the image causes discomfort. This is demonstrated in Figure 2 where the focal length and camera offset were intentionally large. As previously mentioned, the choice on the focal lens f and camera offset c is based on the variation in the disparity values which can be expressed as,

$$fc \geq \min(\max Disparity) \cdot \min(depth)$$

As the depth map is grayscale images that range from 0 to 255, a clear separation of foreground and background depends on disparity separation of foreground and background. Here we chose that the max disparity be 10 pixel given the input image size is around $500x500$. Note that if the depth map distributes in small range, foreground and background will not be separated well.

Figure 2: Drastic changes to the focal length and/or camera offset lead to large disparities between the depth maps which when used to create an anaglyph image, the image causes discomfort.

In addition to using appropriate values for the focal length and camera offset, one has to address issues arising from the rendering. For example, depth buffering needs to be enabled in order to avoid overwriting pixel values. Another problem that arises is the presence of holes [or cracks] in the final rendering. This can be overcome by decreasing the focal length and camera offset, however this may lead to decrease in the disparities which in turn leads to the aforementioned problem with discomfort. This problem has been also reported by others such as in [10] and solutions have been proposed such as oversampling the image and enlarging the warp beam. Another possible solution to this problem is in-painting where the values of neighbouring pixels are used to fill in the missing values. In our work we use in-painting and in particular Navier-Stokes based inpainting method [11].

6. EXPERIMENTAL RESULTS

The proposed technique has been tested with images and videos downloaded from the web. Three example frames are shown in Figure 3. The top row shows the original frames. The second and third row show the depth and normal maps produced by the CNN given as input a single frame. The fourth row shows the anaglyph images which is rendered using the generated depth map and rendering process proposed. Note that in these cases there is no in-painting and artifacts appear throughout the images. This is primarily due to the fact that the size of the depth map computed by the CNN is smaller by a large factor of magnitudes i.e. 147×109 size, which causes misalignment with the original when scaled up. The bottom row shows the result of in-painting. All holes and cracks which were present are filled-in with neighbourhood information leading to perceptually plausible depth maps and anaglyph images. [1]

7. CONCLUSION AND FUTURE WORK

We have presented a novel method for automatic conversion of 2D images/videos to 3D. Our method leverages the strengths of Deep Learning to address the complex problem of depth estimation from a single image. The Convolutional Neural Network produces a depth map which is then used to render the anaglyph image. We use anaglyph images due to the easy and wide availability or red/cyan glasses however this does not limit our approach from being extended to other forms of stereo e.g. 3D TVs. Furthermore, we have presented a simplified formulation for computing the depth map of the second stereo camera given two parameters. The method has been tested with several videos and in the future we anticipate to evaluate the effectiveness of the approach with human participants.

8. REFERENCES

[1] Christoph Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv," in

[1] Example video result can be downloaded from here.

Figure 3: Top row: The original frames from videos depicting different scenes. Second, Third row: The depth and normal maps produced by the CNN using a single RGB image as input. Fourth row: The rendered anaglyph images [without in-painting]. Note the artifacts [cracks] appearing. Bottom row: The in-painted anaglyph image. All images are frames from the NYU2 RGBD dataset.

Electronic Imaging 2004. International Society for Optics and Photonics, 2004, pp. 93–104.

[2] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski, "Building rome in a day," in *2009 IEEE 12th ICCV*. IEEE, 2009, pp. 72–79.

[3] Ashutosh Saxena, Min Sun, and Andrew Y Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 824–840, 2009.

[4] Kevin Karsch, Ce Liu, and Sing Bing Kang, "Depth extraction from video using non-parametric sampling," in *Computer Vision–ECCV 2012*, pp. 775–788. Springer, 2012.

[5] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *IJCV*, vol. 42, no. 3, pp. 145–175, May 2001.

[6] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman, "Sift flow: Dense correspondence across different scenes," in *Proceedings of the 10th ECCV: Part III*, Berlin, Heidelberg, 2008, ECCV '08, pp. 28–42, Springer-Verlag.

[7] Naejin Kong and Michael J. Black, "Intrinsic depth: Improving depth transfer with intrinsic images," in *IEEE ICCV*, Dec. 2015, pp. 3514–3522.

[8] Fayao Liu, Chunhua Shen, and Guosheng Lin, "Deep convolutional neural fields for depth estimation from a single image," *CoRR*, vol. abs/1411.6387, 2014.

[9] David Eigen and Rob Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," *CoRR*, vol. abs/1411.4734, 2014.

[10] Sveta Zinger, Luat Do, and PHN de With, "Free-viewpoint depth image based rendering," *Journal of visual communication and image representation*, vol. 21, no. 5, pp. 533–541, 2010.

[11] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *IEEE CVPR 2001*, 2001, vol. 1, pp. I–355–I–362 vol.1.