Stereo Image Generation using Neural Networks

Siddhant Prakash, *1211092724* and Anurag Solanki, *1211084183*

Abstract—Using stereo image to generate 3D views is a challenging problem to solve, especially due to increase in demand for 3D content with the advent of virtual reality systems like Microsoft Hololens and Oculus Rift. With big companies such as Google, Facebook, Amazon and a lot of start-ups investing heavily in virtual and augmented reality system, this demand is set to increase exponentially with time. In our project, we will try to train a neural network to come up with a model to estimate the stereo pair given a single RGB image for 3D scene reconstruction.

Index Terms—Netural Networks, Autoencoder, VAE, GANs, Stereo, generative models.

1 Introduction

Finally, 3D images and videos are in the mainstream media after being ignored for so long. Since the advent of multi-view geometry, the potentials of stereo image pair in various computer vision problems, such as, depth estimation, object recognition, segmentation, simultaneous localization and mapping (SLAM) etc., has been exploited comprehensively. Stereo image pairs are essentially images of the same scene from two different view. The image pair differ from each other by a projective transformation. Stereo images has been studied extensively in 2-view geometry and geometrical constraints have been established between the pairs which can be exploited to efficiently generate depth map of the scene given the two views. The depth map along with the stereo pairs are all that is need to project a 3D scene on a display. The use of these stereo pairs for scene understanding has been the motivation behind such varied applications. Thus, the importance of having the stereo pair of an image increases manifold.

Given an image of a scene, can we generate its stereo pair image by training a neural networks end-to-end, is what we want to explore through our project. This can very well lead us to a way of estimating a better depth map than with previous methods. Thus the problem becomes that one of estimating a depth map by generating a stereo pair, rather than the traditional other way round as has been since so long.

In this project we present ways to generate stereo image pair of any given image for converting a scene from 2D to 3D. We have explored two generative models, viz. Variational Autoencoders and GANs, and propose a new auto-encoder architecture as an extension of VAEs crafted for this particular problem. The rest of the report is structured in the following way. Section 2 gives a thorough literature research on all the topics we have explored for our task. Following the related works we explain our interpretation of the task and methodology along with the datasets we considered and used in Section 3. The implementation details and results comes next in Section 4 which is followed by a discussion on the results obtained in Section 5. We conclude with Section 6 listing our achievements and the future works we have planned for the project. A representation of our system can be seen in Figure 1.

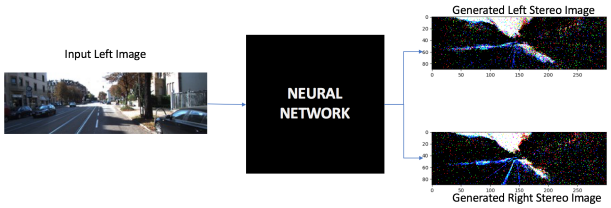


Fig. 1: Input image transformed to stereo-pairs

2 Related Works

The problem of obtaining stereo pair from a single image directly has not been explored much in the academia. Stereo image pairs have been used to deal with a number of computer vision problems. In depth estimation, number of algorithms [15] [16] [17] have been developed to optimally utilize the 3D scene information captured by the pairs. Stereo matching has been explored in many conventional computer vision algorithms [22] [23] problem which tries to estimate the cost of matching stereo image pairs. These algorithms act as the first stage of many stereo algorithms. Recently, neural network has been employed to compare these patches [24] in a fast as well as efficient manner, studying the trade offs between the two.

Although in past years, 3D view generation from single image gained momentum using learning based methods like Im2depth [30] and Make3D [31], which employed an MRF based algorithm to capture the 3D location and orientation of patches in an image. Eigen et. al. [32] was one of the first neural network based method to deal with depth estimation from single image which employed two deep network stacks for prediction. Recently published, competing directly with the work we are trying to do is Deep3D [25], which addresses the problem of generating stereo pairs using single image. Although, the accuracy

• *S. Prakash and A. Solanki is with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, 85281.*
E-mail: {sprakas9, asolank1}@asu.edu

achieved by the work is best, we tried to explore different approaches mentioned in Section 3.2 and 3.3 like Variational Autoecoders and GANs, to better the accuracy. Also, due to the availability of new and larger datasets like ScanNet, we were optimistic about our work.

Many approaches has been explored for construction of a model to represent the distribution of data with the help of its latent information popularly known as representational learning. There are hierarchical model such as [4] which represent the probability distribution of different types of data whose models are generated. But in recent years, instead of graphical models, generative models which are pitted against a discriminative models have gained an edge over representational learning. These techniques do not include probabilistic distribution explicitly. There were other approaches like restricted Boltzmann machines [18] [19], Skip-gram model [20] [21] and Variational Auto-encoders, which we are exploring.

Kingma et al [14] introduced efficient inference and learning in directed probabilistic model whose latent variables have intractable posterior distributions models. Variational Autoencoder efficiently approximates posterior inference in almost any model with continuous latent variables. This learned posterior inference can be also used for a host of tasks such as recognition, denoising, representation, generation and visualization purposes. We are trying to exploit the fact that Variational autoencoders can be used as generative models. Can they generate image similar to the input image with shifted perspective.

Another approach very popular in recent years is that train a generative model from random data distribution and use a discriminator model to discriminate between real input data versus the fake generated data. Goodfellow et. al. [9] introduced this concept of Generative Adversarial Networks, and we exploit these networks as one of the methods to come up with a good representation of the images we are learning. Although, the nature of stereo-pairs is similar, they are not exactly same.c As the change between both the pairs is only of the viewpoint, there is very minimal change in depth which can be taken as almost same. We bank on the intuition that the GAN models can actually learn this minute affine transformations which differs the pair of images. We also know how difficult it is to train GANs. Thus, we explore different techniques introduced by [10] [11] which enables us to learn the model in a stable manner. Chen et al [12] and Shrivastava et al [13] introduces tricks to improve the real like appearance of the generated images, and we would like to explore these as they demonstrate very good results in understanding the latent semantics of the data distribution in representational learning.

3 METHODS

3.1 Dataset

The first and foremost requirement to learn good models using deep neural networks are their hunger for large datasets. Until recently, not many large datasets were available for the problem of RGB-D scene understanding. But now with the advent of datasets like KITTI [1] and Middleburry [2] stereo datasets, we can delve into the domain of learning models for 3D scene understanding. While the KITTI dataset has 400 dynamic scenes, along with Middleburry dataset, they do not contribute to more than a few thousand frames. Another recent addition to this class of dataset is the ScanNet [3], which provides us with 2.5M views, which should mostly satisfy our requirement. Although, in [25] the dataset used were the 3D movies downloaded from internet, which provided about 5M views. We tried to obtain the movie dataset although the ScanNet data and surprisingly even KITTI dataset were good enough for our purpose.

3.1.1 ScanNet Data set

ScanNet is a dataset of richly-annotated RGB-D images of real-world environments which contains 2.5M RGB-D images in 1513 scans acquired in 707 distinct spaces. [3] As this is RGB-D dataset stereo pair of the image has to be calculated from depth. We gained access to download this dataset but we have not use them in our experiments.

3.1.2 KITTI Stereo 2015 Data set

KITTI dataset consists of 200 training outdoor scene and 200 test outdoor scenes. In total, we have distributed 4200 stereo pairs of images for training and 4189 stereo pairs of images for testing of full dataset. Images comprise of dynamic scenes for which ground truth (stereo-pair) is known. The total dataset(scene flow multi view) is approx 13.5 GB including both testing and training images. We have used the KITTI dataset for all our experiments. One of the stereo pairs of image from the dataset can be seen in Figure 2.

3.1.3 Synthetic Matlab Data

For initial exploratory tasks, we simulated our own 3D dynamic scene using Matlab and captured the scene from two viewpoints corresponding to stereo pair of images. We explain the procedure we used to obtain the synthetic stereo pair of images.

The first step to obtaining data set was to create a 3D scene in Matlab. We used the in-built "importGeometry" function to import a 3D model in the plot. We then pass the axes of the window to a nothe function we created which estimates the current viewpoint and helps us obtain the stereo pairs given the distance between the two camera centers of the left and right view. This distance is equal to $b/2$ where b is the distance between the two views as shown in Figure 3 [27] .

To obtain the different viewpoints, we first inquire the view property of the figure using "get" function. We obtain the azimuth and the elevation of the camera of which we manipulate the azimuth from $[0°, 180°]$ to get the stereo view of the model from all angle around it. We can also manipulate the elevation from $[-90°, 90°]$, but as we approach the top view, the variation in views from around the model becomes very negligible. So we restrain the range of angles of elevation to manipulate within a small limit, ideally $[-15°, 15°]$. The stereo pair is generated, by taking another parameter corresponding to b and for best result we keep b to be small, ideally within $(0, 5]$.

This enables us to obtain a number of stereo synthetic pair of images for given one model. If say the view of an object is given by (a, e), where a is the azimuthal angle and

(a) Left Image

(b) Right Image

Fig. 2: Stereo Image pair from KITTI Dataset

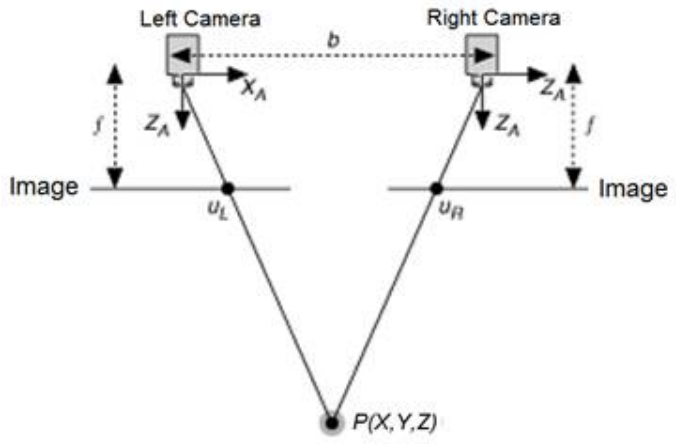


Fig. 3: Stereo Vision System

e is the elevation, we obtain stereo image pair with the given views,

$$Left\_view : (a - d/2, e)$$

$$Right\_view : (a + d/2, e)$$

, where d is the change in azimuthal angle due to the change in distance between center of baseline and the corresponding view, i.e

$$d \simeq b$$

for small angles ($d \leq 5°$).

We faced some challenges in obtaining the synthetic dataset using Matlab. First of all, the support for importing 3D model as a figure is very primitive in Matlab. Only '.stl' files are supported from import. To import more complex 3D data structures, such as '.obj' and '.off', we need to first load the object and then voxelize it using helper functions. Then, we need to figure out the co-ordinate system and map them to the plot point-by-point. We have restricted our modeling to only '.stl' objects.

One of the major issue with both the KITTI data set images and synthetic data set images are their size. The images are of very high resolution, which presents us with a memory bound while providing input to the neural network. Thus, we have reduced the resolution of our images before we provide them as input to our architecture. This, lowers our accuracy which we could have attained at full resolution, but since the project is exploratory by nature, we were willing to make the trade-off.

3.2 Variational Autoencoder

Variational autoencoders are encoders that learns the latent variable model for its input data. Instead of letting neural network learn any arbitrary function, Variational autoencoders learn the parameters of probability distribution which models the data. If we sample data from this distribution, we may generate data similar input data. Variational autoencoders can act as 'Generative Model'. [14]

In this paper, we have modified the above Variational autoencoder approach. We give left image as input, encoder

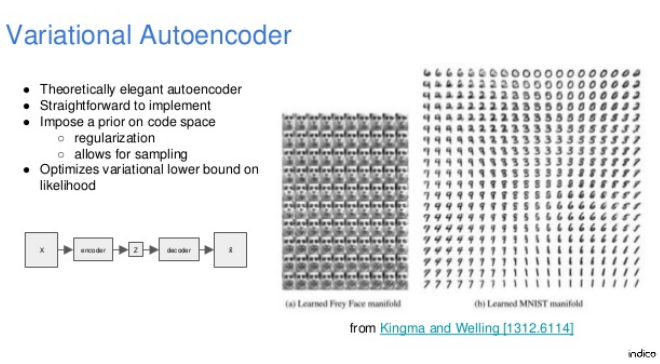


Fig. 4: Variational Auto-encoders

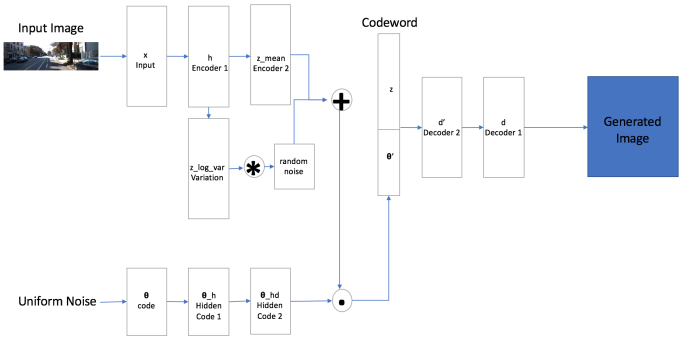


Fig. 5: Our Implemented Variational Auto-encoder

will encode it in latent space and decoder will then decode the image back with some reconstruction error. We call this reconstructed image. In our first approach, this reconstructed image could be compared with right image of the same view. As variational autoencoders may generate an image from latent space after learning the distribution, so we can also compare the generated image from the latent space with right view image.

3.3 Generative Adversarial Networks

GAN is a framework to train deep generative model using a mini-max game. There are two models, viz. a generator and a discriminator which play against each other to learn the data distribution. The goal of the generator is to learn the probability distribution $P_G(x)$ which is as close as to the real data distribution $P_{data}(x)$. The generator G instead of learning probability for each x learns the data distribution P_G by mapping a random noise variable $z \sim P_{noise}(z)$ into a sample $G(z)$.The generator is trained by playing a game against a discriminator D which distinguishes samples from real P_{data} versus fake P_G. Formally, the minimax game is given by Equation 1.

$$min_G max_D V(D,G) = \mathbb{E}_{x\sim P_{data}}[logD(x)] + \mathbb{E}_{z\sim P_{noise}}[log(1-D(G(z)))] \quad (1)$$

4 Experiments & Results

4.1 Modified VAE

Initially, we implemented a basic model for Variational Autoencoder. We tried with MNIST dataset to get deeper understanding of the implementation and how the distribution is modeled using the network. We got results similar to the one shown in Figure 4. We next are trying to modify it according to KITTI dataset which required image to be preprocessed.

The first step was to downsize the input images to feed it into the network. As the original KITTI images were of resolution 1310 X 369, we downsized it to 300 X 90 by scaling and cropping the images in proportion such that the height of the images remain 90. We were able to feed these RGB images in the network and got initial results with the normal VAEs.

Our basic architecture of VAE implementation for generating stereo pair of image is shown in Figure 5

Some of the experiments we did are as mentioned below.

- We have done experiments with modifying intermediate dimensions (256, 512, 1024, 2048) and latent dimensions (64, 128, 256, 512).
- We have also changed the hyper parameters with using optimizer Adam [33] and RMSProp [34].
- We experimented with loss functions, initially we took only the VAE loss [33]. After that we tried VAE loss with categorical cross entropy loss of generated image with right image from dataset.

The images generated from this modified VAE with loss 1, only VAE loss, is shown in Figure 6. For other images generated with loss 2 and loss 3, please check the supplementary material. In the image, the top image is the input left image, the middle image is the generated image from the VAE, and the bottom is the right stereo pair ground truth images.

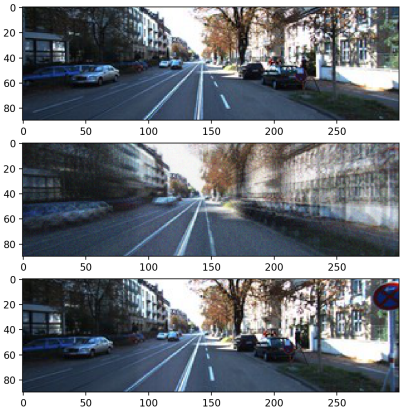


Fig. 6: Generated Image from Modified VAE

We have verified the 3D reconstruction of the stereo pair for visual testing from the following website [35]. The website gives a disparity map along with the oscillating the stereo pairs at a certain frequency in horizontal direction to give a 3D appearance. The disparity map can be seen in Figure 7 along with the screen shot of the stereo pair at an instant.

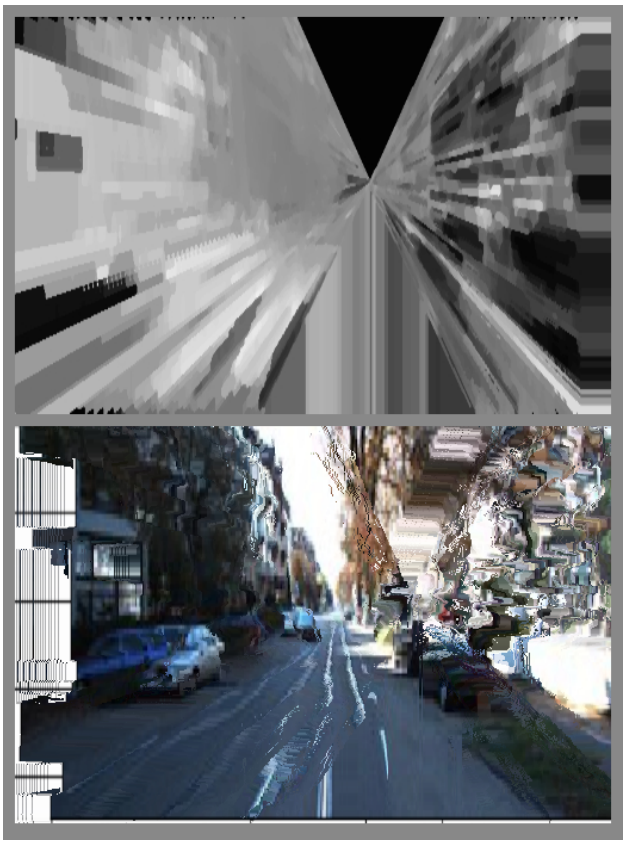


Fig. 7: Disparity Map of left (input) and right(generated) stereo image

4.2 Stereo encoder-decoder network

We tried the approach to transform the variational auto-encoder is such a way that after we input one image (i.e. Left image) then our decoder comes up with two images that are stereo pairs of each other. Our intuition was to use the left image pair as input image, which going through a normal de-noising auto-encoder outputs a representation of the left images. Meanwhile, the codeword augmented by the uniform noise input to the network as an extra parameter should be able to capture the transform parameters, ideally the projective transform parameters of the stereo pair of the input image. Thus, we generate a new image using the codeword, which gives us the right stereo pair image representation of the input left image. The structure of the encoder-decoder network is shown in Figure 8.

The network is inspired from the original variational encoder network we implemented, but we hypothesized that to capture the transformation between the image our generated image need to have a variational code in the latent space which we try to model as the right image pair space. Thus, the top part of the network is a normal auto-encoder while the bottom part captures the essence of the transformation, which is projective by nature, in the image.

We did the same experiments for this network as we did for the original VAE model. We further wanted to implement the network as fully convolutional network, by introducing convolutional encoder-decoder layers, which we are in the process of implementing at the time of our report generation.

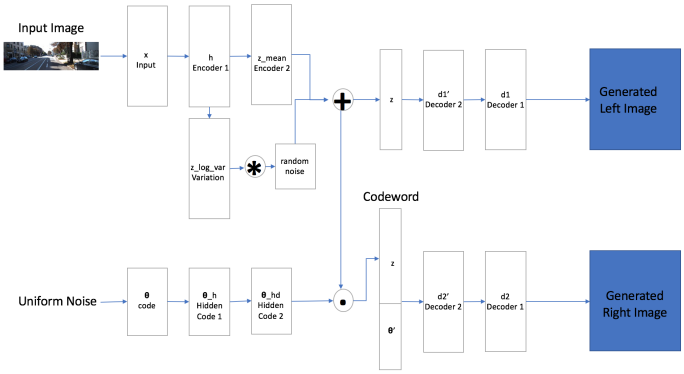
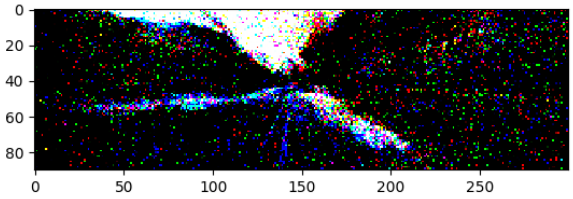
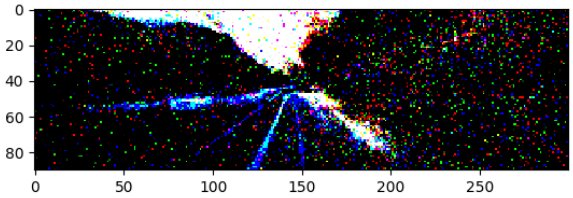


Fig. 8: Stereo encoder-decoder netwok

The image pair generated from one of the configuration we tested can be seen in Figure 9. We can see the generated images are like stereo pairs, but the images are still noisy. We attribute it to multiple causes, which we discuss in the discussion section.



(a) Left Generated Image



(b) Right Generated Image

Fig. 9: Stereo Image pair generated from stereo encoder-decoder network

4.3 GANs & Variations

We implemented a basic GAN architecture. The problem was to come up with network which we can formulate for our problem. As mentioned in related works, GANs are really tricky to train. We implemented DC-GAN network, introduced in [10] and [11], as our base network. We came up with a hypothesis of formulating our stereo pair problem in terms of InfoGAN [12].

InfoGAN is different form normal GAN such that, instead of taking only one incompressible noise vector z in its formulation, it takes an additional parameter c which are called the latent codes. These latent codes target the salient feature of a data distribution. Thus, while the GAN objective tries to model the input image distribution and generate similar images, the introduction of latent codes learns discrete salient feature of a given distribution, such as rotation, size, elevation etc. We hypothesize, that given a left generated image, if we try to learn the relation between this generated image coming from the input image distribution, and learn the latent codes for two categories, viz. first the properties which are similar to the right images, like color, texture, objects and secondly, the transformation, ideally the projective matrix parameters, we may be able to learn the latent distribution of right image. In which case, we can generate left image, from input distribution while the right stereo pair image can be generated from the estimated auxiliary distribution.

$$min_{G,Q} max_D V_{InfoGAN}(D,G,Q) = V(D,G) - \lambda L_I(G,Q) \tag{2}$$

Thus, the original InfoGAN formulation is given by Equation 2. Here, $V(D,G)$ is the original GAN objective function from Equation 1, with the modification being, that the generator network $G(z,c)$ takes input both, z and c. Q is the new auxiliary network that is being learned by maximizing the mutual information between the generated image distribution $P_{G(z,c)}$ and auxiliary distribution $P_{Q(c|x)}$. The mutual information term $\lambda L_I(G,Q)$ is given by Equation 3.

$$\lambda L_I(G,Q) = \mathbb{E}_{c\sim P(c), x\sim G(z,c)}[logQ(c|x)] \; + \; H(c) \tag{3}$$

Thus according to our hypothesis, $G(z,c)$ generates image similar to the left image distribution, but we introduce a mutual information term via the distribution $Q(c|x)$. If, instead of giving random latent codes from uniform or categorical distributions to $Q(c|x)$, we give the right image distribution $P(x')$ and try to map it as the auxiliary distribution $Q(c|x)$, it will learn the right distribution, and the mutual information term, will try to regularize the generated image which maximizes the mutual information between the left distribution and right distribution by learning the latent transformation codes by means of c. Thus, for our problem, the complete InfoGAN variant objective function can be represented by Equation 2 with the mutual information term $\lambda L_I(G,Q)$ given by Equation 4.

$$\lambda L_I(G,Q) = \mathbb{E}_{c\sim P(x'), x\sim G(z,c)}[logQ(c|x')] \; + \; H(c) \tag{4}$$

5 Discussion

We can see, the images generated by modified VAE is blurry, which we know is a disadvantage of generating images using variational auto-encoders. As we mentioned the images generated from stereo encoder-decoder network were noisy. This could be due to fault in our generator network, in ways in which we interpret the output pixels. We are trying to fix this issue too, because it seems that the images we are getting are true stereo pairs of each other.

We came up with an implementation of DC-GAN and using the DC-GAN a general implementation of InfoGAN, but we are yet to implement the modified InfoGAN implementation for our hypothesized InfoGAN. Thus, we are not submitting the code for InfoGAN with the supplementary material.

We also feel that since all auto-encoder network were completely fully-connected networks, using a convolutional auto-encoder network will significantly improve the quality of images that we are generating. Thus, our focus currently lies in implementation of convolutional stereo encoder-decoder network for stereo pair generation.

Looking at the time line we proposed initially, we have completed all the tasks. We have experimented with different architectures as described above to accumulate the results. We faced lots of difficulties such as limited computational resources, high image resolution, difficulty in obtaining 3D stereo pair datasets, outdated Keras toolbox etc.

6 Conclusion

We have come up with a solution to solve the stereo pair generation of image problem. We explored the various kinds of generative models studied in the literature. We came up with three different types of network architecture, which can be formulated for the aforementioned problem. Two of our proposed method uses an auto-encoder network while the third uses a variation of InfoGAN model. We backed our hypothesis by strong theoretical background of the methods proposed and showed some initial results in our investigation for the task, with the minimal implementation we could come up with of our hypothesis by the time of writing of this report.

Appendix A
Future work

We would like to extend this project and improve the results that we are getting now. We also thought to improve our results by making use of Convolutional Neural Network, but due to lack of time could not complete it on time.

Acknowledgments

The authors would like to thank Professor Baoxin Li and instructor Ragav Venkatesan for providing their valuable guidance. We also appreciate their considerable efforts which helped us to complete the project successfully.

REFERENCES

[1] Geiger, Andreas, et al. "Vision meets robotics: The KITTI dataset." The International Journal of Robotics Research 32.11 (2013): 1231-1237.

[2] D. Scharstein, H. Hirschmller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In German Conference on Pattern Recognition (GCPR 2014), Mnster, Germany, September 2014.

[3] A. Dai, A. Chang, M. Savva, M. Halber, T. Funkhouser, M. Niebner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. https://arxiv.org/pdf/1702.04405.pdf

[4] Yoshua Bengio (2009), "Learning Deep Architectures for AI", Foundations and Trends in Machine Learning: Vol. 2: No. 1, pp 1-127. http://dx.doi.org/10.1561/2200000006

[5] Hinton, Geoffrey E., Alex Krizhevsky, and Sida D. Wang. "Transforming auto-encoders." International Conference on Artificial Neural Networks. Springer Berlin Heidelberg, 2011.

[6] Dong, Chao, et al. "Image super-resolution using deep convolutional networks." IEEE transactions on pattern analysis and machine intelligence 38.2 (2016): 295-307.

[7] Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." Proceedings of the 25th international conference on Machine learning. ACM, 2008.

[8] Mao, Xiaojiao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections." Advances in Neural Information Processing Systems. 2016.

[9] Reed, Scott, et al. "Generative adversarial text to image synthesis." Proceedings of The 33rd International Conference on Machine Learning. Vol. 3. 2016.

[10] Salimans, Tim, et al. "Improved techniques for training gans." Advances in Neural Information Processing Systems. 2016.

[11] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

[12] Chen, Xi, et al. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets." Advances in Neural Information Processing Systems. 2016.

[13] Shrivastava, Ashish, et al. "Learning from Simulated and Unsupervised Images through Adversarial Training." arXiv preprint arXiv:1612.07828 (2016).

[14] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

[15] Zhang, Liang, and Wa James Tam. "Stereoscopic image generation based on depth images for 3D TV." IEEE Transactions on broadcasting 51.2 (2005): 191-199.

[16] Eigen, David, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network." Advances in neural information processing systems. 2014.

[17] Garg, Ravi, Gustavo Carneiro, and Ian Reid. "Unsupervised CNN for single view depth estimation: Geometry to the rescue." European Conference on Computer Vision. Springer International Publishing, 2016.

[18] Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18.7 (2006): 1527-1554.

[19] Desjardins, Guillaume, Aaron Courville, and Yoshua Bengio. "Disentangling factors of variation via generative entangling." arXiv preprint arXiv:1210.5474 (2012).

[20] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[21] Kiros, Ryan, et al. "Skip-thought vectors." Advances in neural information processing systems. 2015.

[22] Kolmogorov, Vladimir, Pascal Monasse, and Pauline Tan. "Kolmogorov and Zabihs graph cuts stereo matching algorithm." Image Processing On Line 4 (2014): 220-251.

[23] Luo, Wenjie, Alexander G. Schwing, and Raquel Urtasun. "Efficient deep learning for stereo matching." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[24] Zbontar, Jure, and Yann LeCun. "Stereo matching by training a convolutional neural network to compare image patches." Journal of Machine Learning Research 17.1-32 (2016): 2.

[25] Xie, Junyuan, Ross Girshick, and Ali Farhadi. "Deep3d: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks." European Conference on Computer Vision. Springer International Publishing, 2016.

[26] Mathieu, Michael, Camille Couprie, and Yann LeCun. "Deep multi-scale video prediction beyond mean square error." arXiv preprint arXiv:1511.05440 (2015).

[27] 3D Imaging with NI LabVIEW. "How Stereo Vision Works", Publish Date Jan 04, 2017 http://www.ni.com/white-paper/14103/en/

[28] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks." Aistats. Vol. 15. No. 106. 2011.

[29] Goodfellow, Ian J., et al. "Maxout Networks." ICML (3) 28 (2013): 1319-1327.

[30] Baig, Mohammad Haris, et al. "Im2depth: Scalable exemplar based depth transfer." Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on. IEEE, 2014.

[31] Saxena, Ashutosh, Min Sun, and Andrew Y. Ng. "Make3d: Learning 3d scene structure from a single still image." IEEE transactions on pattern analysis and machine intelligence 31.5 (2009): 824-840.

[32] Eigen, David, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network." Advances in neural information processing systems. 2014.

[33] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[34] Tieleman, Tijmen, and Geoffrey Hinton. "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude." COURSERA: Neural networks for machine learning 4.2 (2012).

[35] "Online Computational Stereo Vision". last updated 18122015. http://www.ivs.auckland.ac.nz/quick_stereo/index.php