Automatic Generation of 3D GIFs

Jee Ian Tam
Stanford University
Stanford, CA 94305
jeetam@stanford.edu

Abstract

We combine monocular depth estimation with motion segmentation techniques to produce split-depth GIFs (GIF is a common animated image format, and split-depth is a technique to enhance the 3-D effect of the GIF by inserting white bars that split the depth of the scene). We use a deep convolutional neural field (DCNF) model to perform monocular depth estimation, and also perform motion segmentation using Gaussian Mixture Model for color cues and Optical Flow for motion cues into a Markov Random Field. We test the DCNF model on the NYU v2 dataset and find an average relative error of 0.24 and a RMS error of 0.825. We also compare the produced split-depth GIFs with a dataset of manually-annotated ones to find an average recall of 0.94 and average precision of 0.83 in computing the foreground/background pixels relevant to the split-depth level. We find that combining motion segmentation with monocular depth estimation results in an increase in precision of roughly 20% compared to solely using monocular depth estimation to generate the split-depth GIFs.

1. Introduction

GIF (Graphics Interchange Format) is a widespread file format used to display and share video animations on the internet, and easily lend themselves to modifications and enhancements. One type of modification is to make a GIF look more 3-dimensional by editing the GIF to include white vertical reference bars that split the depth of the scene in the GIF. We henceforth refer to such GIFs as "split depth GIFs". Some (static) examples are shown below.

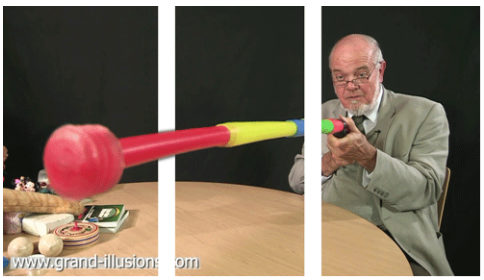

Figure 1: Split-depth GIF example 1, from `http://i.imgur.com/ZaskDW7.gif`

Figure 2: Split-depth GIF example 2, from `http://gfycat.com/AdeptSorrowfulIrishsetter`

The effect is most pronounced when a main object is approaching the viewer. The effect also seems to partially stem from the "window pane effect" that arises from the combination of the white vertical lines with a white background. The community that is interested in making split-depth GIFs currently does so by manually creating frame-by-frame masks using image editing software, which is particularly tedious. The main motivation for this project would be to use computer vision techniques to produce a pipeline/tool that can automatically generate convincing split-depth GIFs instead.

The input to our system is a GIF of arbitrary resolution and duration, and the output is the same GIF with added vertical white bars which produce the split-depth effect. We initially approached the problem as one of monocular (single-image) depth estimation, but later realized much better results by segmenting the main moving objects in the GIF as well.

2. Related Work

2.1. Review of previous work

The two main computer vision methods used in our work are monocular depth estimation and motion segmentation.

The classic work on monocular depth estimation was done by Saxena [1], where a depth map is learned via supervised training from a set of ground-truth RGB-D images based on summary statistics of local image patches, and uses a hierarchical, multiscale Markov Random Field to account for local depth consistency. [2] showed that incorporating information from semantic segmentation can help with depth map prediction. Eigen et. al [3] used a multi-scale deep convolutional network to predict a depth map from a single image instead. [4] trained an end-to-end deep CNN solution for converting 2D videos to 3D videos based on a training set of 3D movies on a frame-by-frame basis without incorporating temporal constraints. State-of-the-art is [5] which combined a deep convolutional neural network with a conditional random field to learn a depth mapping for single images, which is what we use for this work. [6] pursues an alternate approach of recovering depth from an estimation of the defocus map of a single image.

A good review of motion segmentation techniques can be found in [7]. A basic approach is to segment the moving object via a Markov Random Field on the Optical Flow of the video frames, as was done in [8]. [9] additionally includes estimated depth information as well. [10] extends this work by also learning a Gaussian Mixture Model of color cues for the moving objects to provide better segmentation performance, and is the main model that we use in this work. [11] also improves on this by presenting a fast online algorithm for motion segmentation. State-of-the-art is given in [12] where a convolutional network is trained on image and motion fields to detect moving objects and discard over/under segmentations or background parts of a scene.

2.2. Key contributions of current work

The current work is focused on applying existing computer vision work and tools to solve an existing problem (creation of split-depth GIFs), and so the current work does not extend the current state-of-the-art. Rather, our work is novel in the sense that there does not currently exist a tool to programatically create such split-depth GIFs, and so our current work is a proof-of-concept for such a solution. Our main contribution is to apply current state-of-the-art models for monocular depth estimation and motion segmentation to solve this problem, and investigate the limitations of those models in this context.

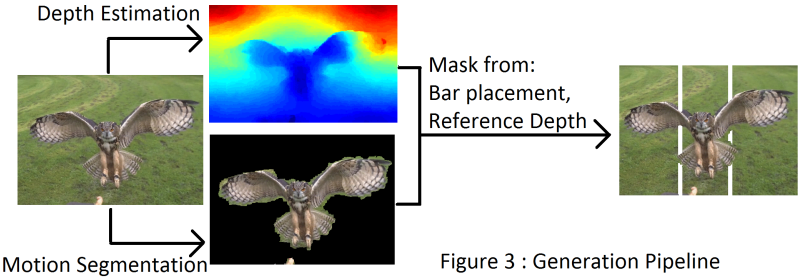

Figure 3 : Generation Pipeline

3. Technical Solution

3.1. Summary of technical solution

Our pipeline for producing split-depth GIFs is as follows :

1. Process the input GIF to obtain its constituent frames
2. For each frame, calculate the optical flow from the previous frame to obtain initial estimates of moving object segmentation using a Markov Random Field.
3. Learn a Gaussian Mixture Model from pixel colors of the initial estimates of the moving object, and use this to refine the moving object segmentation via another pass through the GIF frames.
4. For each frame, estimate a depth map for the frame using a pre-trained deep convolutional neural field model as given in [5]
5. Set a placement for the two vertical white bars, and a reference depth.
6. For each frame, create a mask that reveals the pixels on the white bars only if the pixel depth is below the reference depth and belongs to the moving object.

This is summarized in figure 3. There are currently several parameters that need to be tuned in the process to obtain a convincing split-depth GIF. These are :

- Optical Flow Smoothness and Threshold Parameters
- Motion Segmentation Markov Random Field Parameters
- Placement of vertical white bars
- Reference depth

3.2. Details of technical solution

3.2.1 Motion Segmentation

We implement the techniques presented in [10]. After extracting frames from the GIF, we calculate the an optical

flow field for each frame using the vision.OpticalFlow function in Matlab's vision toolbox. The optical flow field is computed using the Horn-Schunck method [13], with smoothness being the primary variable parameter.

We then threshold the optical flow field to obtain a binary mask that is an initial motion cue estimate of the moving object in the GIF. This motion cue is sparse is often only indicates the boundary of the moving object. To solve this issue we use a Markov Random Field (MRF) to take into account the fact that neighboring pixels are likely to have the same label, and neighboring pixels with similar colors are more likely to have the same label.

The pairwise potentials of the MRF are given as

$$\psi(l_i, l_j) = exp\Big(\frac{\lambda l_i l_j}{\alpha + d(i,j)}\Big) \quad (1)$$

where l_i is the label for pixel i with $l_i = 1$ if pixel i belongs to a moving object and $l_i = -1$ otherwise. λ and α are tunable parameters, and $d(i,j)$ is a color distance measure between pixels i and j, which we set to be the weighted euclidean color distance [14] $d(i,j) = \sqrt{3(R_i - R_j)^2 + 4(G_i - G_j)^2 + 2(B_i - B_j)^2}$. R_i is the red color value for pixel i, and similarly G_i for green and B_i for blue.

The unary potentials for the MRF are given as

$$p(f_i|l_i) = p_m(of_i|l_i) = exp\Big(l_i(of_i - \delta_m)\Big) \quad (2)$$

where f_i is the feature of pixel i (specifically the optical flow in this case), and of_i is the value of the optical flow field at pixel i. δ_m is a parameter.

Thus, given the above MRF potentials, moving object segmentation for an image frame I can be achieved by finding the labeling L that maximizes the posterior

$$P(L|I) \propto \prod_{i \in I} p(f_i|l_i) \prod_{i \in I} \prod_{j \in N_i} \psi(l_i, l_j) \quad (3)$$

where N_i is the set of 4-connected neighborhood pixels around pixel i. We solve the optimization via graph cut algorithms. Specifically, we use the GCMEX [15] Matlab wrapper, which implements graph cuts based energy minimization techniques described in [16] [17] [18].

After using the graph cut algorithm to get a segmentation of the moving object, we train a Gaussian Mixture Model (GMM) G^f on the RGB pixel values of the moving object using an expectation-maximization algorithm with an agglomerative clustering strategy to estimate the optimal number of components. This is done using the CLUSTER package [19]. The affinity of a pixel with color c to the moving object G^f is estimated as

$$affi_c(c) = max_{g_j \in G^f}\big[g_j(c)\big] \quad (4)$$

$$g_j(c) = \frac{p_j}{\sqrt{|\Sigma_j|}} exp\Big(-\frac{1}{2}(c - \mu_j)^T \Sigma_j^{-1}(c - \mu_j)\Big) \quad (5)$$

where g_j is the j-th component of the Gaussian Mixture Model G^f. p_j is the corresponding prior probability, Σ_j is the covariance matrix and μ_j is the mean vector. We then define a likelihood model based on this affinity as

$$p_c(c_i|l_i) = exp\Big(l_i(log(affi_c(c_i)) - \delta_c)\Big) \quad (6)$$

where δ_c is a parameter.

Thus, we extend the unary potential presented in equation 2 to include the moving object color GMM as follows :

$$log(p(f_i|l_i)) = log(p_m(of_i|l_i)) + \lambda_c log(p_c(c_i|l_i)) \quad (7)$$

We then solve the MRF using the graph cut algorithm as was done previously to obtain the final segmentation of the moving object in the GIF.

3.2.2 Monocular Depth Estimation

We use the pre-trained model provided by [5] to perform monocular depth estimation on each frame of the GIF. The architecture of the model is shown in figure 3.

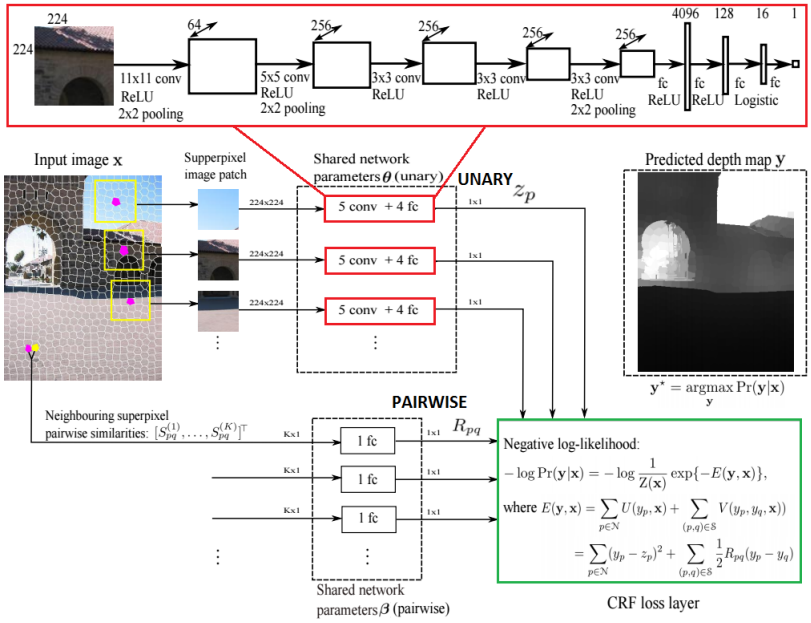

Figure 3: Deep Convolutional Neural Field architecture

The input image is first over-segmented into superpixels. To compute unary potentials z_p for each superpixel, an image patch centered around its centroid is cropped, resized, and fed into a CNN composed of 5 convolutional and 4 fully-connected layers. To compute pairwise potentials R_{pq}, K types of similarities are computed for each pair of neighboring superpixels p,q and fed into a fully-connected layer. Thus, given the unary and pairwise potentials, the loss for a CRF can be computed, and the whole network can be optimized/trained by minimizing the negative log-likelihood of the data. To predict the depth $\mathbf{y}$ of a new image $\mathbf{x}$, the conditional probability $P(\mathbf{y}|\mathbf{x})$ is maximized as follows :

Let $\mathbf{A} = \mathbf{I}+\mathbf{D}-\mathbf{R}$, where $\mathbf{I}$ is the $n \times n$ identity matrix, $\mathbf{R}$ is the affinity (pairwise potential) matrix composed of R_{pq}, and $\mathbf{D}$ is the diagonal matrix with $D_{pp} = \sum_q R_{pq}$. With $\mathbf{z}$ as the unary potentials, the CRF energy function can thus be written as

$$E(\mathbf{y}, \mathbf{x}) = \mathbf{y}^T A\mathbf{y} - 2\mathbf{z}^T\mathbf{y} + \mathbf{z}^T\mathbf{z} \tag{8}$$

and we thus have the predicted depth map as

$$\mathbf{y}^* = argmax_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \tag{9}$$
$$= argmin_{\mathbf{y}} E(\mathbf{y}, \mathbf{x}) \tag{10}$$
$$= \mathbf{A}^{-1}\mathbf{z} \tag{11}$$

4. Experiments and Results

4.1. Unused Explorations

In this subsection we present some findings from techniques that we initially investigated but ultimately did not use in the final pipeline.

We downloaded various models from [3], [5], [6], [4] and [28], and also implemented a basic version of the algorithm as presented in [1]. One of the limitations of all of the models is that they are trained on a limited number of datasets - The most common datasets are Make3D[29] and NYU v2 Depth [30] datasets, where the former consist of outdoor college campus scenes and the latter consist of indoor house and office scenes. Since GIF scenes span a variety of situations, the less sophisiticated models used in [28], [1] and [3] did not display good generalization to scenes that were not similar to the datasets that they were trained on, and the depth maps generated were not accurate enough to isolate a general foreground object in an image frame from the background. [6] only works well with images that have a defocused background or foreground.

[4] shows decent results for predicting depth from a single image. The model is a CNN that predicts a right-eye image given an input (left-eye) image, with the model trained from a dataset of 3D movies. Initial attempts at estimating a disparity/depth map calculated from the left-eye and right-eye images yielded poor results. This is because the model constructs a right-eye image from multiple probabilistic depth representations from internal CNN layers. Thus, the internal depth representations give a more accurate representation of the depth map calculated by the model. I was able to extract the activations from the internal layers to obtain the probabilistic depth maps as shown in figure 4. However, we chose to not use this model in favor of [5] instead as we were able to obtain more accurate results with the latter. Part of this is the limited depth resolution of this model due to the limited number of depth levels.

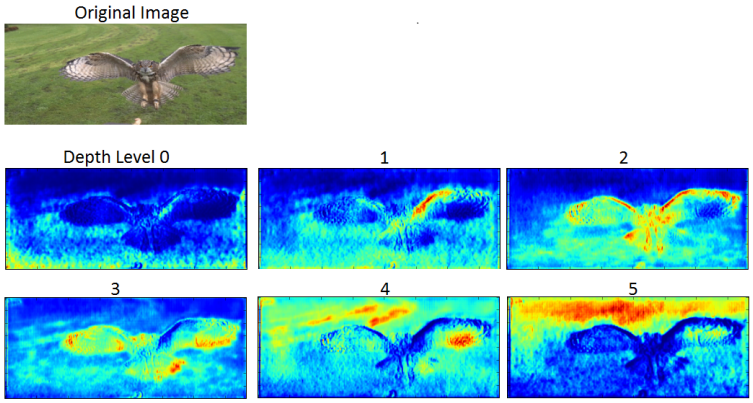

Figure 4: Internal CNN probabilistic depth representation of [4]. Heatmaps represent probability that pixel belongs to that level, lower level corresponds to distance being closer to camera

4.2. Motion Segmentation

Figure 5: Motion Segmentation Example. Left : Original Frame. Middle : Thresholded Optical Flow Field. Right : Final Segmentation Results

An example of the result of the motion segmentation algorithm is presented in figure 5. The corresponding GIFs can be found at `http://imgur.com/9xnvavQ` and `http://imgur.com/JnrJXHs`.

We can see that the motion segmentation manages to segment the moving object most of the time, but usually

over-segments the object and ends up segmenting some of the background as well. [10] addresses this by determining the component in the Gaussian Mixture Model that corresponds to the background and removing it from the model, but this was not implemented due to time constraints.

4.3. Monocular Depth Estimation

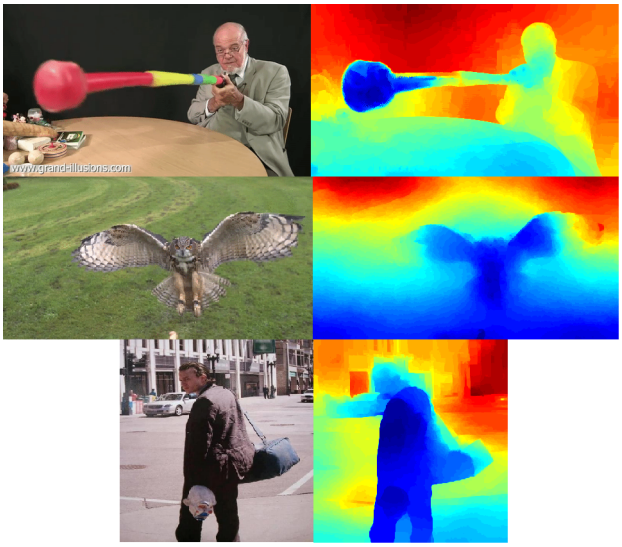

Figure 6: Example deep convolutional neural field depth maps. Heatmap represents distance from camera

We show the result of evaluating the pre-trained deep convolutional neural field model on several images in figure 6. The model is implemented using the MatConvNet [20] library. We see that the model is sophisticated enough to be able to predict depth maps across a variety of scenes. We also test the model on the NYU v2 [30] dataset, which contains a variety of indoor scenes annotated with depth data. We find that the model gives an average relative error of 0.24, and a RMS error of 0.825, where the average relative error is defined as $\frac{1}{T}\sum_p \frac{|d_p^{gt} - d_p|}{d_p^{gt}}$ and RMS error is defined as $\sqrt{\frac{1}{T}\sum_p (d_p^{gt} - d_p)^2}$. Here d_p^{gt} and d_p are the ground-truth and predicted depths respectively at the pixel p, and T is the total number of pixels in all evaluated images. This is similar the results presented in [5].

4.4. Generation of split-depth GIFs

We present key frames of several GIFs that we convert to split-depth GIFs in figure 7. The actual GIFs can be found at `http://imgur.com/a/fE7hs`. We see that for certain types of scenes, our pipeline is qualitatively able

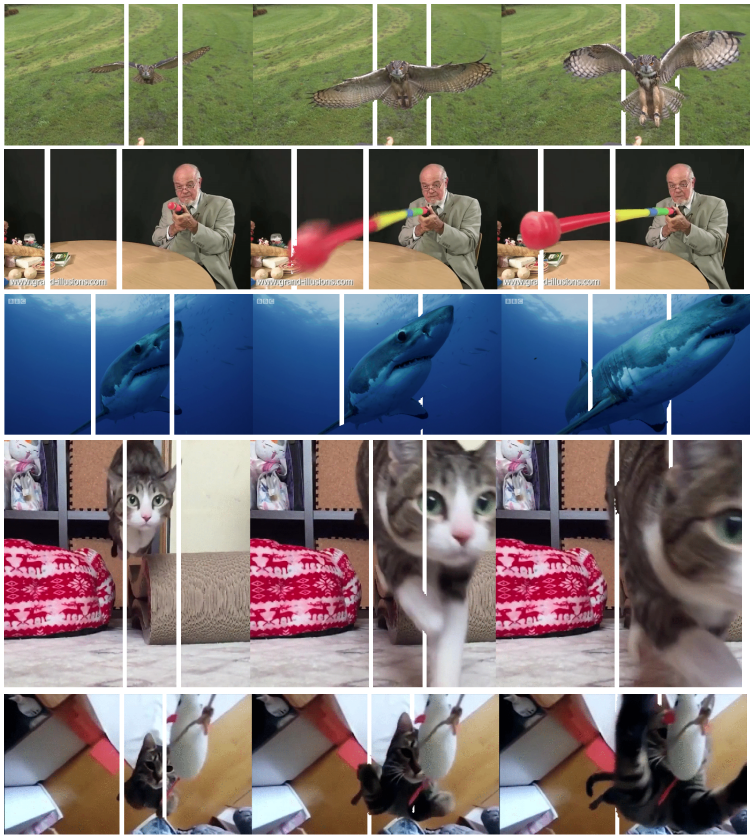

Figure 7: Sample key frames from generated split-depth GIFs.

to generate convincing split-depth GIFs that enhance the 3-D effect.

To quantify the quality of the generated split-depth GIFs, we compare a dataset of generated GIFs with hand-annotated versions by the community at the website `http://reddit.com/r/splitdepthgifs`. We use binary classification metrics to quantify the performance. For pixels of the vertical white bars, we define a true positive as a pixel being correctly classified as foreground (in front of the vertical white bars), and a true negative as a pixel being correctly classified as background (behind the vertical white bars). From this, we get a precision of 0.94 and recall of 0.83. In evaluating the metric, it must be kept in mind that the moving object spends a majority of the GIF behind the vertical white bars, and so we would expect to get a relatively high precision since most of the time the pixels are background pixels.

One key point is that motion segmentation greatly enhances the performance of the algorithm. For example, in figure 8 we see that just using the estimated depth map (which can also be seen in figure 6) results in placing a

Figure 8: Generated GIF frame without (Left) and with (Right) moving object segmentation.

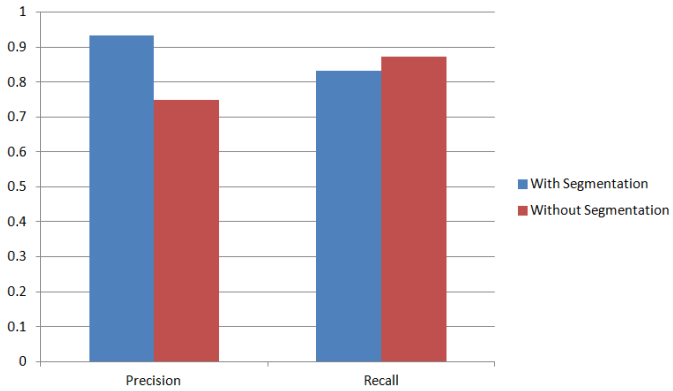

Figure 9: Precision and Recall with and without motion segmentation

portion of the background in front of the vertical white lines. The main insight is that the split-depth effect is most effective when only the depth of the moving object is split, but not the depth of the background. Thus, by performing motion segmentation, the quality of split-depth GIFs is greatly improved. This can also be seen in figure 9, where not using motion segmentation results in a noticeably lower precision due to the large increase in false positives (pixels that should be hidden, but are instead revealed).

5. Conclusions and Future Work

We have demonstrated a system to programatically generate split-depth GIFs. We use a deep convolutional neural field to perform monocular depth estimation on each frame of the GIF, and combine this motion segmentation via a Markov Random Field on optical flow motion cues and learned Gausian Mixture Model color cues. Our approach works best with scenes that have a static camera and moving object that approaches the camera sufficiently close to allow for a good ”splitting” of the depth of the scene. We find that segmentation of the moving object is crucial in producing a convincing split-depth GIF, as the main point is the split the depth of the moving foreground object, but not that of the background.

The motion segmentation method used in this work can be improved upon as in [10] where the background is additionally learned and subtracted from the Gaussian Mixture Model, or [12] where a CNN is used for motion segmentation instead. This would improve the quality of the generated GIFs.

Furthermore, the current pipeline involves manual tuning of several parameters such as the optical flow, graph cut and vertical white bar placement parameters in order to create convincing GIFs. Future work would also involve finding methods to reduce the necessity of manual parameter tuning, such as automating placement of the vertical white bars by maximizing the variance of the scene hidden by them while regularizing by temporal smoothness.

Lastly, our system involves combining data from two disparate systems for depth estimation and motion segmentation as can be seen in figure 3. An better system would be to train an end-to-end convolutional neural field across time to predict both depth and motion segmentation from an input video stream.

For code, please see `https://www.dropbox.com/sh/2qiiahwj4z75xvu/AABzmtv9qHhAVNfZSNaOFgnja?dl=0`.

References

[1] Saxena, Ashutosh, Sung H. Chung, and Andrew Y. Ng. "3-d depth reconstruction from a single still image." International journal of computer vision 76.1 (2008): 53-69.

[2] Liu, Beyang, Stephen Gould, and Daphne Koller. "Single image depth estimation from predicted semantic labels." Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010.

[3] Eigen, David, and Rob Fergus. "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[4] Xie, Junyuan, Ross Girshick, and Ali Farhadi. "Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks." arXiv preprint arXiv:1604.03650 (2016).

[5] Liu, Fayao, Chunhua Shen, and Guosheng Lin. "Deep convolutional neural fields for depth estimation from a single image." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[6] Zhuo, Shaojie, and Terence Sim. "Defocus map estimation from a single image." Pattern Recognition 44.9 (2011): 1852-1858.

[7] Zappella, Luca, Xavier Llad, and Joaquim Salvi. "Motion segmentation: A review." Proceedings of the 2008 conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence. IOS Press, 2008.

[8] Jung, Cheolkon, and Joongkyu Kim. "Motion segmentation using Markov random field model for accurate moving object segmentation." Proceedings of the 2nd international conference on Ubiquitous information management and communication. ACM, 2008.

[9] Klappstein, Jens, et al. Moving object segmentation using optical flow and depth information. Springer Berlin Heidelberg, 2009.

[10] Liu, Feng, and Michael Gleicher. "Learning color and locality cues for moving object detection and segmentation." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[11] Ellis, Liam, and Vasileios Zografos. "Online learning for fast segmentation of moving objects." Computer VisionACCV 2012. Springer Berlin Heidelberg, 2012. 52-65.

[12] Fragkiadaki, Katerina, et al. "Learning to segment moving objects in videos." Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE, 2015.

[13] Horn, Berthold K., and Brian G. Schunck. "Determining optical flow." 1981 Technical symposium east. International Society for Optics and Photonics, 1981.

[14] "Colour Metric." Colour Metric. Web. 06 June 2016, http://www.compuphase.com/cmetric.htm

[15] Fulkerson, Brian, Andrea Vedaldi, and Stefano Soatto. "Class segmentation and object localization with superpixel neighborhoods." ICCV. Vol. 9. 2009.

[16] Boykov, Yuri, and Vladimir Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision." Pattern Analysis and Machine Intelligence, IEEE Transactions on 26.9 (2004): 1124-1137.

[17] Boykov, Yuri, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." Pattern Analysis and Machine Intelligence, IEEE Transactions on 23.11 (2001): 1222-1239.

[18] Kolmogorov, Vladimir, and Ramin Zabin. "What energy functions can be minimized via graph cuts?." Pattern Analysis and Machine Intelligence, IEEE Transactions on 26.2 (2004): 147-159.

[19] Bouman, Charles A., et al. "Cluster: An unsupervised algorithm for modeling Gaussian mixtures." (1997).

[20] Vedaldi, Andrea, and Karel Lenc. "MatConvNet: Convolutional neural networks for matlab." Proceedings of the 23rd Annual ACM Conference on Multimedia Conference. ACM, 2015.

[21] Liu, Miaomiao, Mathieu Salzmann, and Xuming He. "Discrete-continuous depth estimation from a single image." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[22] Ideses, Ianir, Leonid P. Yaroslavsky, and Barak Fishbain. "Real-time 2D to 3D video conversion." Journal of Real-Time Image Processing 2.1 (2007): 3-9.

[23] Varekamp, C., and B. Barenbrug. "Improved depth propagation for 2D to 3D video conversion using key-frames." Visual Media Production, 2007. IETCVMP. 4th European Conference on. IET, 2007.

[24] Po, Lai-Man, et al. "Automatic 2D-to-3D video conversion technique based on depth-from-motion and color segmentation." Signal Processing (ICSP), 2010 IEEE 10th International Conference on. IEEE, 2010.

[25] Wu, Chenglei, et al. "A novel method for semi-automatic 2D to 3D video conversion." 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2008. IEEE, 2008.

[26] Yan, Xi, et al. "Depth map generation for 2d-to-3d conversion by limited user inputs and depth propagation." 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011. IEEE, 2011.

[27] Cao, Xun, Zheng Li, and Qionghai Dai. "Semi-automatic 2D-to-3D conversion using disparity propagation." Broadcasting, IEEE Transactions on 57.2 (2011): 491-499.

[28] Jin, Ren "Structured Depth Detection Toolbox V1.0"

[29] Saxena, Ashutosh, Min Sun, and Andrew Y. Ng. "Make3d: Learning 3d scene structure from a single still image." Pattern Analysis and Machine Intelligence, IEEE Transactions on 31.5 (2009): 824-840.

[30] Silberman, Nathan, et al. "Indoor segmentation and support inference from RGBD images." Computer VisionECCV 2012. Springer Berlin Heidelberg, 2012. 746-760.

[31] Liu, Ce, Jenny Yuen, and Antonio Torralba. "Sift flow: Dense correspondence across scenes and its applications." Pattern Analysis and Machine Intelligence, IEEE Transactions on 33.5 (2011): 978-994.