2019

Ghost Towns: Semantically Labelled Object Removal From Video

William Clifford
National University of Ireland Maynooth

Charles Markham
National University of Ireland Maynooth

Recommended Citation

Clifford, W. & Markham, C. (2019). Ghost towns: semantically labelled object removal from video. *IMVIP 2019: Irish Machine Vision & Image Processing*, Technological University Dublin, Dublin, Ireland, August 28-30. doi:10.21427/1tag-k222

Ghost towns -Semantically Labelled Object Removal from Video

William Clifford* and Charles Markham

Department of Computer Science, Maynooth University, Maynooth, Co. Kildare, Ireland.

Abstract

This paper describes a method used to produce a video of a road in which the foreground items which obstruct the view of the road have been removed i.e. other vehicles. Once these regions have been identified they are replaced using suitable images that closely resemble the original background. The work considers an approach that uses multiple video sequences of the same road ($C_1 ... C_n$). One video is identified as video C_p, that requires the least repair. All instances of vehicles in each frame of video were identified using a Convolutional Neural Network (CNN). The regions associated with each vehicle were then filled using suitable regions from the frames from the remaining streams (assuming at least one of these streams has a background region visible which matches our query region in C_p). To match frames and locate suitable patches ideally the video sequences need to be aligned both temporally and structurally. To match the frames temporally a bag of visual words approach was taken. To align the frames structurally a template search was performed on regions surrounding the region to be replaced. Given the template matches, the region between these templates in the matching frame were used to fill where the vehicles were previously, leaving behind only the background.

Keywords: Imaging, Image Processing, Machine Vision, Driving Simulation, Video.

1 Introduction

The aim of this work is to extend on previous work into building a photo-realistic driving simulator [Brogan et al., 2013]. It is relatively straightforward to adjust speed using accelerator pedal information to control the apparent speed of a vehicle. A bigger challenge is the introduction of steering. This issue was resolved, in part, by implementing projective texture mapping approach in which the video is projected on to a 3D estimate of the road geometry [Clifford et al., 2017]. The limitation of this technique was that visual artefacts were produced when a foreground object not matching the assumed geometry of the scene entered a frame. In addition, vehicles moving past the camera car could never be overtaken on playback. The aim of this research is to remove vehicles from a frame in order to produce a video that will allow projective texture mapping to work more effectively in the driving simulator. 3D reconstruction of a scene is a popular topic in machine vision [Mur-Artal et al., 2015, Klein and Murray, 2007, Izadi et al., 2011]. Transient moving objects (pedestrians and vehicles) can be considered noise in the reconstruction data, for the work being done here. Another application of this work may be to provide video free from this noise prior to reconstruction.

Within the last ten years there has been advances in image labelling, scene perception, and inpainting. Advances in the area of machine learning (deep neural networks) have assisted in solving these problems. Often the limitation of using such systems lies in acquiring enough training data in order to produce the desired output. For example, foreground removal in the inpainting domain can be accomplished by providing training data that both has an image with and without bit masks covering images [Liu et al., 2018, Pathak et al., 2016, Vo et al., 2018]. This can either be accomplished by using real data collected or generative adversarial networks which hide parts of the image in one end and find the missing region in the other.

*William.Clifford@mu.ie

This paper proposes an alternative method which could provide a closer to ground truth solution, while also being able to pick out pixels for removal based on their semantic meaning.

2 State of the Art

2.1 Inpainting and Background Substraction

Inpainting techniques have come far from patch based recovery [Bertalmio et al., 2001, Telea, 2004]. It has been reported that it is possible to train CNN's to develop a contextual understanding of an image and inpaint based on that context, similar to humans [Pathak et al., 2016]. The results of this form of inpainting are impressive. However, in many cases, inpainted regions appear blurred. More recently improvements have been made by [Vo et al., 2018], these appear to provide more convincing inpainted images.

Background modelling is typically used to inpaint or background subtract foreground items in video as opposed to a single frame. For a taxonomy on how these models are initialized see [Bouwmans et al., 2017]. Typically these methods assume a static camera. So the background remains static allowing a probabilistic model to be computed based on the assumption foreground objects will move and the background will not [Bloisi et al., 2015, Laugraud et al., 2016, Laugraud et al., 2017]. Best results among these have been seen in neural network approaches [Maddalena and Petrosino, 2012], as compared by [Bouwmans et al., 2017].

2.2 Neural Networks for Scene Labelling

Currently, image labelling appears to be a solved problem. Given a large enough dataset of labelled images deep neural networks can classify the presence of an object to high degrees of accuracy [Krizhevsky et al., 2012, Simonyan and Zisserman, 2014]. This has even been refined to a pixel level in semantic segmentation, and finally instance segmentation [Zhou et al., 2017, Zhou et al., 2018, Hariharan et al., 2015]. It is now possible to classify each pixel in an image semantically. There are many resources available for using these algorithms whether it be Pytorch or Keras for developing a neural network. There are many open sourced datasets for tackling particular problems aside from the ones in the papers mentioned. In this work driving datasets may be more relevant, including CamVid and KITTI database [Brostow et al., 2009, Alhaija et al., 2018].

2.3 Scene Reconstruction

In scene reconstruction many sensors can be used but often standard cameras are used. Point correspondences between frames are calculated usually over a set of features found by algorithms such as Scale Invariant Feature Transform (SIFT) or Oriented FAST and Rotated BRIEF (ORB) [Lowe, 2004, Rublee et al., 2011]. The presence of features can be used to find closely matching frames using an approach such as bag of binary words [Gálvez-López and Tardós, 2012]. While the disparity between features among the frames can be used to create 3D point clouds with the aim of constructing a model of the scene [Mur-Artal et al., 2015]. However there are other algorithms which use more sophisticated depth cameras such as the Kinect and do not need to make these considerations [Izadi et al., 2011].

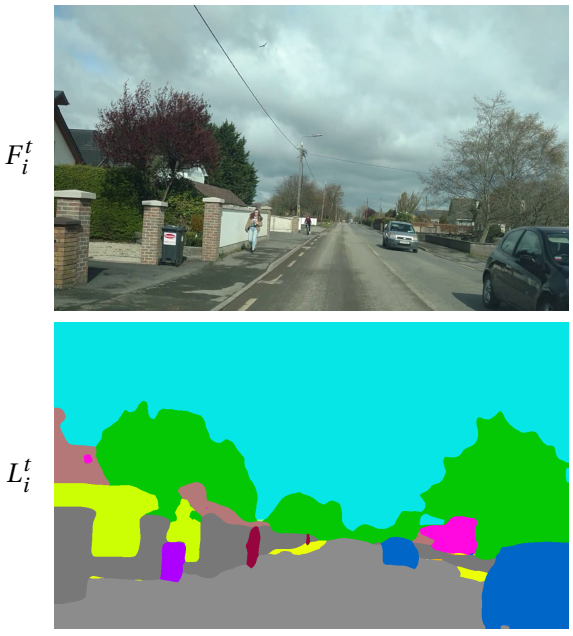

Figure 1: Top image is an example of a frame collected from the Moyglare Road in Maynooth. Bottom image is the semantically labelled version of that image (e.g. blue pixels are cars).

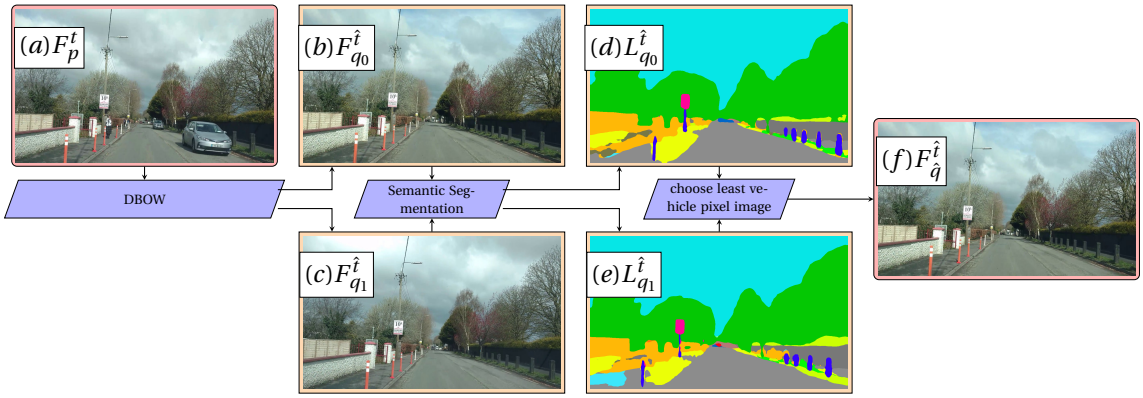

Figure 2: Workflow of how a frame is nominated to assist in the inpainting technique. The image to be filled is the input, $(a)F_p^t$. DBOW is used to find frames similar to the input frame from the other streams, $(b)F_{q_0}^{\hat{t}}$ and $(c)F_{q_1}^{\hat{t}}$. The two frames outputted are then semantically segmented, $(d)L_{q_0}^{\hat{t}}$ and $(e)L_{q_1}^{\hat{t}}$. The number of vehicle pixels in the segmented image are counted and the image with the least vehicle pixels is returned, $(f)F_{\hat{q}}^{\hat{t}}$.

3 Approach

A set of n cameras are used in this process $\{C_i\}$, where $i \in [1...n]$. Each camera was moving and recorded at different times. Camera's produced frames F_i^t, where i corresponds to the subscript of the camera it is from, and t is the timestamp (frame number). For each set of frames produced by the each camera C_i, a corresponding semantically labelled image is produced using a trained CNN [Zhou et al., 2017, Zhou et al., 2018]. The network used a pretrained architecture including Resnet-50 and a Pyramid Pooling Module [He et al., 2016, Zhao et al., 2017]. This produced corresponding labelled frames L_i^t, where i is the corresponding camera it came from and t is the timestamp, see Figure 1. Using the semantically labelled frames and their individual pixels $L_i^t(x,y)$, it was possible to identify the frames that had the fewest vehicle pixels, and thus which camera's stream had to undergo the least foreground removal. By converting the labelled images to bit masks, $m(L_i^t(x,y))$, for all labelled vehicle pixels within all given L_i^t, it is possible to calculate which camera viewed the fewest vehicles and requires the least foreground removal, see Equation 1. For example the mask may be formed by only looking for blue pixels which correspond to labelling cars.

$$C_p = \min_i \sum_0^t \sum_0^x \sum_0^y m(L_i^t(x,y)) \tag{1}$$

The camera selected from Equation 1 with the lowest count of vehicle pixels, C_p, where p is the selected camera. This was chosen as it required the least image recovery following vehicle removal.

In camera C_p the pixel locations that matched the same pixel locations marked as vehicles $m(L_p^t(x,y))$ were removed, for all frames, F_p^t. To fill the missing regions in F_p^t the other sequences were used. To find a suitable matching frame in another sequence the feature matching process bag of binary words was used [Gálvez-López and Tardós, 2012]. This used ORB features identified in each frame and found the top ten matching frames in the other streams C_{i-1}, Figure 3. The top ten frames were used to make it more likely that a match can be found for each camera used, as well as a frame temporally ahead of the previous frame ($F_q^{\hat{t}}$ where $\hat{t} >= \hat{t} - 1$). In this case there were only 3 streams used, including C_p.

To pick which frame should be used to replace pixels in frame F_p^t, the nominated frames from each stream C_{i-1} have their semantically segmented corresponding frames compared, in a similar way to Equation 1, except the minimisation is done over a frame basis, Equation 2. The frame with the least vehicle labelled pixels is selected for pixel replacement Figure 2.

$$F_{\hat{q}}^{\hat{t}} = \min_{q,\hat{t}} \sum_0^y \sum_0^x m(L_q^{\hat{t}}(x,y)) \tag{2}$$

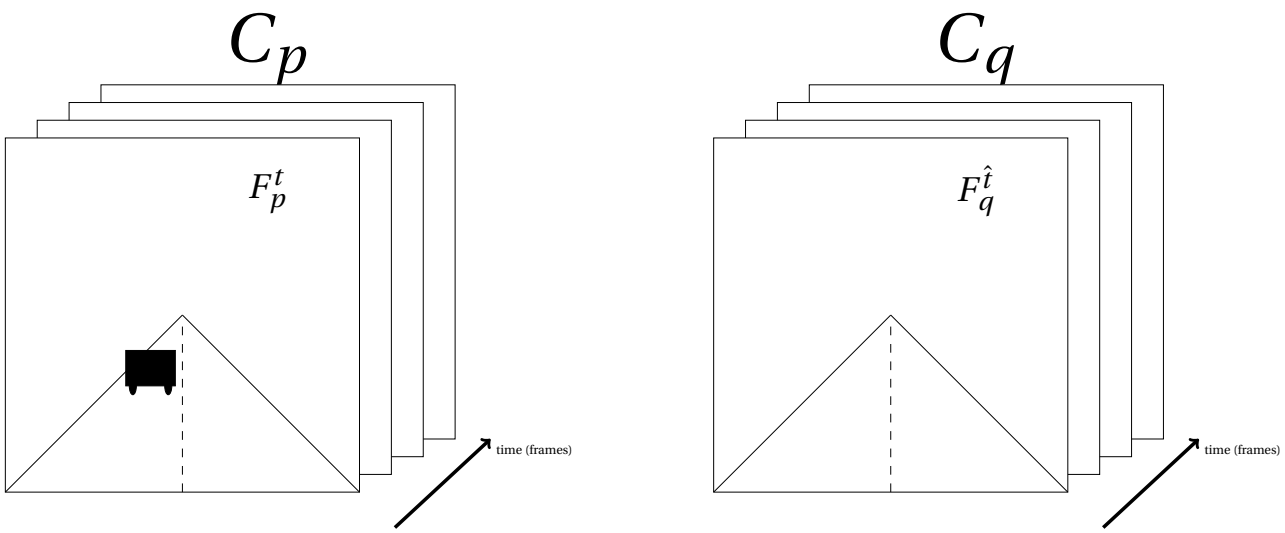

Figure 3: Two streams C_p and C_q. C_p's frame F_p^t has an obstruction in its foreground. The corresponding frame in C_q, $F_q^{\hat{t}}$, does not have this obstruction. The obstruction free area in $F_q^{\hat{t}}$'s frame is used for replacing the missing background in F_p^t following foreground (obstruction) removal.

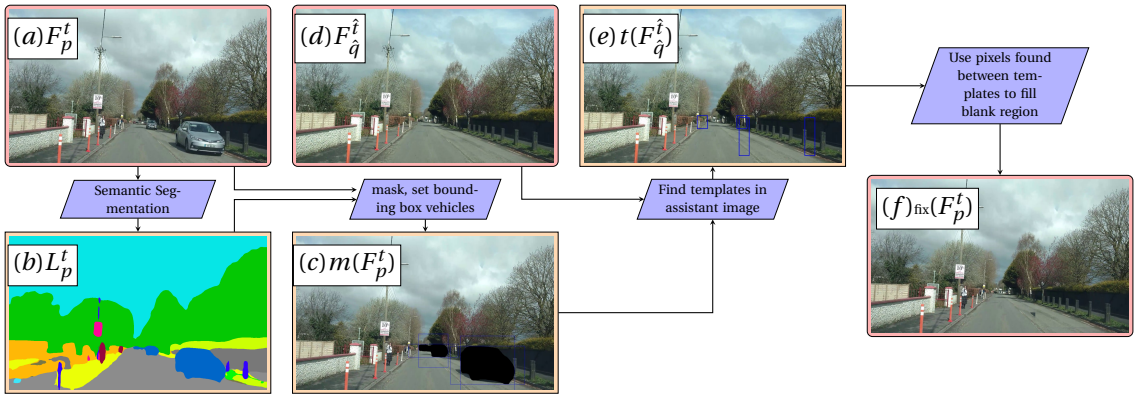

Figure 4: Given the query image, $(a)F_p^t$, the semantically segmented version of this image is found, $(b)L_p^t$. This labelled image is used to find vehicles in the image. Bounding boxes are set around the vehicles. Templates are chosen based off the adjacency to the bounding box, $(c)m(F_p^t)$. These templates are used to find where they lie in the other image from the previous workflow, $(d)F_{\hat{q}}^{\hat{t}}$, Figure 2. Those template positions are found, $(e)t(F_{\hat{q}}^{\hat{t}})$. The regions between the found template positions are used to fill the blank areas in the query image. Leaving the inpainted image behind, $(f)\text{fix}(F_p^t)$.

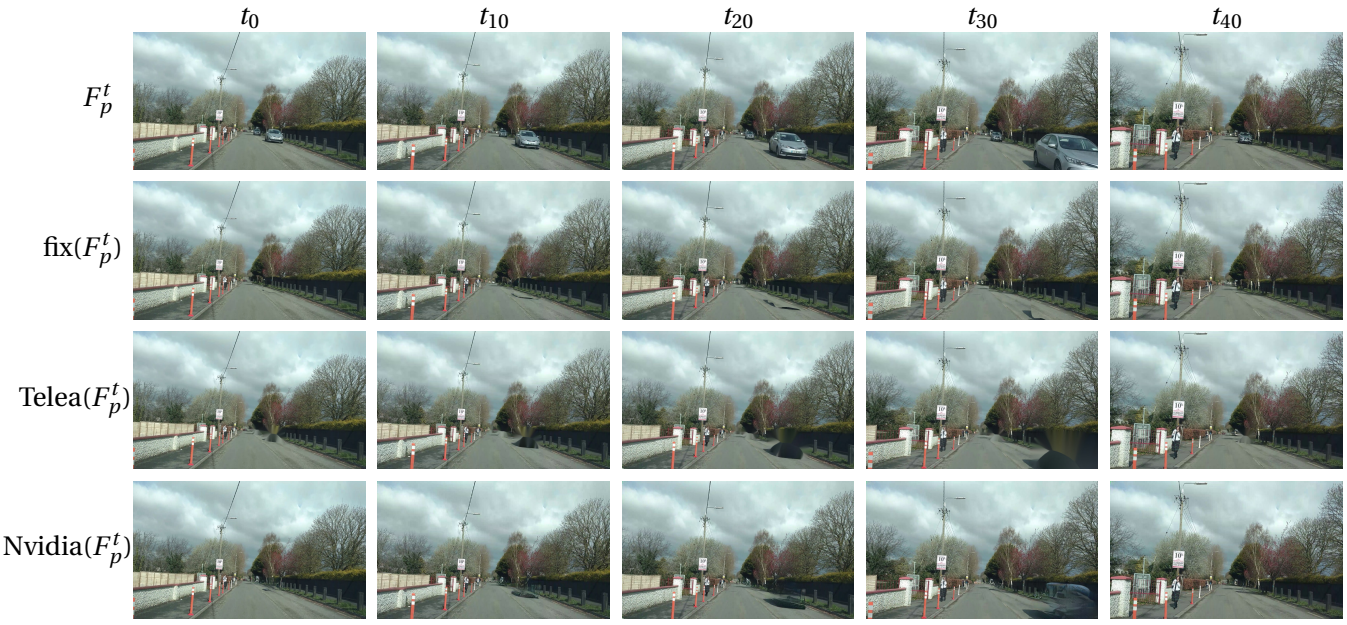

Figure 5: A comparison of inpainting algorithms, run over 50 frames, in intervals of 10 frames. The input is shown in F_p^t. The output of the workflow described in this paper is shown in the row labelled fix(F_p^t). The row labelled Tealea(F_p^t), is an example of a basic inpainting algorithm ran on the same masks provided to this papers algorithm. The final row labelled Nvidia(F_p^t) is an example of a more recent inpainting algorithm ran on the same masks provided to this papers method.

When a matching frame was found, $F_p^t \leftrightarrow F_{\hat{q}}^{\hat{t}}$, a template match over the regions surrounding where the obstruction in F_p^t was carried out over the matching frame in $F_{\hat{q}}^{\hat{t}}$. This was done for each instance of a labelled vehicle in F_p^t. With the resulting match, the pixels between the matching templates were used to fill the vehicles previous location, Figure 4.

Finally when the region was selected for replacement alpha blending was used to make the union of the two subsections follow the contour of the region rather than using a rectangular region. This was carried out based off the semantically labelled version for F_p^t, L_p^t. This allowed a pixel level of accuracy of where the replacement needed to occur, instead of just filling in a bounding box.

4 Results

The workflow described in Figure 4 works well for most frames. When working there are still minor issues including shadows from the inpainted object and the alignment of the patched regions with major projective lines in the scene. This is more obvious when the car gets closer to the camera, see t_{10} to t_{30} in Figure 5. The same issues can be seen when running the algorithms described in [Telea, 2004, Liu et al., 2018], see Figure 5 under labels Telea(F_p^t), and Nvidia(F_p^t).

5 Discussion

The method described works when template matching is viable, where the matched frames do not have significant scalar differences. This happens often as the same road was recorded multiple times, driving in the same direction, and on the same lane. Any large changes to the areas adjacent to the bounding boxes of the vehicles currently inhibit this algorithm from finding a match. For this reason, feature detection like SIFT or ORB around the vehicles should be used in future instead of an exact template match [Lowe, 2004, Rublee et al., 2011]. This will allow the features found while using DBOW to be used again. Saving on run time by only searching the images once. Future work will involve using the two images, for example F_p^t, the

input image, and $F_q^{\hat{I}}$, the feature matched image, to calculate a disparity map between the two. This can be done using stereo rectification based on their shared epipolar geometry, and calculating difference between pixel positions, see [Ma et al., 2012] for details on how this could be done.

In some sense the pixel level of accuracy works against this algorithm as it still exposes the shadows of the vehicles. However using techniques such as dilation or even a inpainting methods specifically aimed at shadow removal could solve this issue. It would be better to remove the shadows afterward instead of replacing larger regions of the original image.

The application for this work is to remove obstructions from video so the geometry of roads can be assumed and projective mapping of textures can be computed more easily [Clifford et al., 2017]. It is also planned to run this algorithm on a video for later use in a sparse SLAM system such as ORBSLAM [Mur-Artal et al., 2015], to see how it performs against an unedited video. A comparison of how the SLAM system works with this technique and another video inpainting algorithm would be informative.

Although this method may appear to only work just as well as inpainting algorithms which require less information (a single video sequence), see Figure 5, it does something the other inpainting algorithms does not. It can later retrieve what is actually behind the objects it has removed, instead of guessing. Although this could be done with optical flow, that would assume the object eventually moved out of the way of the item that was missed in previous frames. With the approach being developed the user can simply record the environment multiple times until their view is not obstructed.

The statistical based algorithms for background substraction perform well on static cameras [Laugraud et al., 2016, Laugraud et al., 2017, Maddalena and Petrosino, 2012, Bloisi et al., 2015]. The problem being worked on here involves moving cameras. Improvements to the prior knowledge of the scene could make these methods available here. Keeping track of the trajectory of the background using optical flow may allow for a model to be constructed. A problem with this may be examples where the foreground does not move (a parked car). Multiple recordings or scene labelling may be useful for this instance.

This approach requires the entire scene to be semantically labelled, even when only a small part of the scene is dynamic. To reduce the computational load of this workflow entire image labelling of every few frames instead of every frame could be carried out. Meanwhile image labelling within a window surrounding any existing tracked objects would allow for a pixel level of accuracy and a smaller region to undergo labelling.

6 Conclusion

This workflow allows for a convincing level of inpainting to the degree that the viewer is no longer aware that there was a car within the scene. However there remains some significant visual artefacts. These are most apparent when the alignment of major projective lines in the scene are included in the patch (e.g. curbs along the side of the road). Further work is needed to rectify the alignment issues. Ultimately the method being developed offers a possible solution to background removal where the ground truth of the scene is important.

References

[Alhaija et al., 2018] Alhaija, H., Mustikovela, S., Mescheder, L., Geiger, A., and Rother, C. (2018). Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*.

[Bertalmio et al., 2001] Bertalmio, M., Bertozzi, A. L., and Sapiro, G. (2001). Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE.

[Bloisi et al., 2015] Bloisi, D. D., Grillo, A., Pennisi, A., Iocchi, L., and Passaretti, C. (2015). Multi-modal background model initialization. In Murino, V., Puppo, E., Sona, D., Cristani, M., and Sansone, C., editors, *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, pages 485–492, Cham. Springer International Publishing.

[Bouwmans et al., 2017] Bouwmans, T., Maddalena, L., and Petrosino, A. (2017). Scene background initialization: A taxonomy. *Pattern Recognition Letters*, 96:3 – 11. Scene Background Modeling and Initialization.

[Brogan et al., 2013] Brogan, Michael Kaneswaren, D., Commmins, S., Markham, C., and Deegan, C. (2013). Proc. layering reality: Realistic driving simulation. In *IT&T Conference*, AIT, Athlone.

[Brostow et al., 2009] Brostow, G. J., Fauqueur, J., and Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97.

[Clifford et al., 2017] Clifford, W., Deegan, C., and Markham, C. (2017). High speed reconstruction of a scene implemented through projective texture mapping. In McDonald, J., Markham, C., and Winstanley, A. C., editors, *Irish Machine Vision and Image Processing Conference Proceedings 2017*. Irish Pattern Recognition & Classification Society.

[Gálvez-López and Tardós, 2012] Gálvez-López, D. and Tardós, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197.

[Hariharan et al., 2015] Hariharan, B., Arbelaez, P., Girshick, R., and Malik, J. (2015). Hypercolumns for object segmentation and fine-grained localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Izadi et al., 2011] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al. (2011). Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM.

[Klein and Murray, 2007] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10. IEEE Computer Society.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Laugraud et al., 2017] Laugraud, B., Piérard, S., and Droogenbroeck, M. V. (2017). Labgen: A method based on motion detection for generating the background of a scene. *Pattern Recognition Letters*, 96:12 – 21. Scene Background Modeling and Initialization.

[Laugraud et al., 2016] Laugraud, B., Piérard, S., and Van Droogenbroeck, M. (2016). Labgen-p: A pixel-level stationary background generation method based on labgen. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 107–113.

[Liu et al., 2018] Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In *The European Conference on Computer Vision (ECCV)*.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Ma et al., 2012] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S. (2012). *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media.

[Maddalena and Petrosino, 2012] Maddalena, L. and Petrosino, A. (2012). The sobs algorithm: What are the limits? In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 21–26.

[Mur-Artal et al., 2015] Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163.

[Pathak et al., 2016] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544.

[Rublee et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. R. (2011). Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Telea, 2004] Telea, A. (2004). An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34.

[Vo et al., 2018] Vo, H. V., Duong, N. Q. K., and Pérez, P. (2018). Structural inpainting. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 1948–1956, New York, NY, USA. ACM.

[Zhao et al., 2017] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890.

[Zhou et al., 2017] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. (2017). Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Zhou et al., 2018] Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., and Torralba, A. (2018). Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision*.