Combining Geometry and Learning for Scene Understanding

by

Arun Kumar Chockalingam Santha Kumar

(Under the Direction of Suchendra M. Bhandarkar)

Abstract

When an image is captured, the 3D Euclidean space describing its world is projected onto a 2D plane, effectively losing most pertinent underlying 3D Euclidean geometry information. Consequently, the ultimate goal of any 3D scene understanding system is to recover the lost 3D geometry while deconstructing the semantics of the scene, to perform perceptual decision making tasks such as object detection, pose estimation, shape recovery, *etc*.

Furthermore, the ill-posed nature of the 3D scene recovery problem, where multiple shapes can generate the same image, adds further complexity to an already challenging problem. Significant research has been devoted toward solving the 3D scene recovery problem over the past few decades, with approaches ranging from triangulation and space carving using multiple views of the scene, to using learning-based models to learn semantic priors, to reason and reconstruct the scene.

An alternative view of 3D scene understanding is that, given large amounts of data, it is possible to design machines that can automatically learn relevant relationships to perform various vision tasks such as reconstruction, pose prediction, *etc*. with minimal human supervision and without resorting to complex, manually-designed objective functions. The recent upsurge in deep learning techniques and abundance of data accompanied by the availability of annotations, has resulted in several state-of-the-art learning-based 3D reconstruction models that regress the underlying information in a purely data-driven manner. However, the success of deep learning has come at a hefty price, from the cost of gathering training data to the cost of painstaking labor involved in manual annotation of these data.

In light of the above, the goal of this dissertation is to explore both learning-based and geometry-based approaches to 3D scene reconstruction, more specifically, equip learning-based models with geometric reasoning to enable joint scene understanding. This dissertation aims to move away from annotation-intensive learning-based techniques to develop 3D scene reconstruction models that harness the power of geometry and learn from arbitrary data instead of from manually curated 3D datasets, exploit class priors, and most importantly, address the learning and geometric reasoning tasks holistically, to more effectively combat ambiguities in reconstruction and recognition.

INDEX WORDS: 3D Object Detection, Single View Scene Reconstruction, Pose Estimation, Multiple View Geometry, Scene Understanding, Deep Learning.

Combining Geometry and Learning for Scene Understanding

by

Arun Kumar Chockalingam Santha Kumar

A Dissertation Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2018

Combining Geometry and Learning for Scene Understanding

by

Arun Kumar Chockalingam Santha Kumar

Major Professor: Suchendra M. Bhandarkar

Committee: Mukta Prasad
Tianming Liu
Khaled Rasheed

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2018

Combining Geometry and Learning for Scene Understanding

Arun Kumar Chockalingam Santha Kumar

December 2018

To Dad, Mom and Sister

Acknowledgments

First and foremost, I owe a debt of gratitude to Dr. Suchendra Bhandarkar (my major professor) and Dr. Mukta Prasad (my unofficial co-advisor,) for their motivation, guidance and support over these years. Thank you Dr. Bhandarkar, for providing me the freedom to pursue my research interests and a positive learning environment, for pushing the boundaries every time it was possible for me to do better, and most importantly for teaching me to multitask. And thank you Mukta, for all your motivation during challenging times, for pushing me often out of my comfort zone, and above all, for the endless hours you spent on teaching me and for the several one-on-one discussions, that made learning intricate concepts far easier. Both of you have inspired me and played a significant role in instilling in me great research skills, and helped me at every possible level: from brainstorming ideas, implementing them, to writing them up. I also thank both of you for always finding times at the eleventh hour when faced with deadlines that would have been otherwise impossible to meet. This dissertation would not have been possible without both of you.

I would also like to extend my sincere thanks to the other members of my advisory committee, Dr. Tianming Liu and Dr. Khaled Rasheed for helping me with the dissertation planning, for taking their time to review my dissertation, and for their questions and valuable suggestions during my Ph.D. exam presentations.

I sincerely thank Anirban Mukhopadhyay and Karan Sharma for being great collaborators; a special thanks to both of you for introducing me to the field of computer vision. I am also grateful to all my lab mates Somenath, Manu, Srijita, Kyle along with Anirban and Karan, for making the stressful Ph.D. voyage much more enjoyable. I would like to express my heartfelt thanks to my friends especially Ajay Chandran, Johnson Philip, Mohan Kumar, Sindhuri, among many others, for being supportive of me.

Last but not least, I am profoundly grateful to my parents and my sister, for all their encouragement and unconditional support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

Scene understanding is one of the most fundamental problems in computer vision, that aims to obtain a comprehensive understanding of the semantics and recover the 3D structure of a visual scene. It encompasses a variety of sub-problems ranging from detecting objects, recovering their 3D shape and pose for reconstruction, to inferring semantic cues by deconstructing the scene into objects and layout in 3D to further model object-object and/or object-scene interactions. Despite significant advances in recent times, the problem of 3D scene understanding is still far from being solved.

Based on the nature of reconstruction, most traditional 3D scene understanding techniques can be categorized into two major classes: *reconstruction using semantic priors* [2, 4, 14] where we localize, estimate the pose of, and reconstruct an object of a known class, or *geometry driven reconstruction* [8, 7, 9] where the entire scene is reconstructed in a manner which may or may not be object-aware.

Alternatively, based on the nature of inputs used for reconstruction, 3D scene reconstruction techniques can also be categorized as *single view* [14, 1] or *multi-view* [15, 17, 16] reconstruction, where traditionally the former class of techniques attempts to reconstruct the 3D scene/object from a single 2D image mostly by exploiting learned object or scene priors, whereas the latter relies on techniques such as triangulation and space carving to geometrically reason about the scene. At the same time, it has to be noted that there have been a few attempts on using priors for multi-view reconstruction, as well as using geometric cues such as vanishing points for single view reconstruction techniques.

The research in this dissertation is primarily focused on equipping data-driven learning-based models with the ability to geometrically reason about an object or a scene. The goal is to enable these techniques to learn from arbitrary data with minimal need for human annotation.

This dissertation is categorized into three main sections, where the *first* section (comprising of Chapters 2 and 3) focuses on joint class-specific object detection, pose estimation and reconstruction through near automatically learned models, wherein we reconstruct objects in 3D from a single view in an automated pipeline [5, 19] using a 3D deformable parts model. The goal is to move away from using plain regression-based models for 3D object reconstruction (including deep learning methods) by estimating the continuous pose and shape, to geometrically reason in 3D about a scene or an object (modeled as a constellation of parts). The *second* section (Chapters 4 and 5) focuses on full scene 3D reconstruction from monocular video sequences, especially *monocular depth prediction* on video

sequences of street scenes, using geometry-aware deep learning models that are trained end-to-end. We propose the Long-Short Term Memory (LSTM) [20] and Generative Adversarial Network (GAN) [21] based deep learning models, that can exploit the sequential nature of the data to perform geometric reasoning about the pose, thereby obtaining state-of-the art results for the 3D scene reconstruction problem. Finally in Chapter 5 [22], we present a practical application of deep learning involving understanding of natural scenes that are deemed challenging due to the absence of textural and shape cues.

1.1 Motivation

When a 2D image is captured, the 3D Euclidean geometry of the world is lost and so is the 3D geometry of the scene and objects present in it, making it hard to reason about depth, plan navigation tasks *etc*. Thus the ultimate goal of a 3D scene reconstruction system is to recover the lost 3D Euclidean space in which the scene is embedded. One of the major challenges in doing so is that, the problem is fundamentally ill-posed. An infinite combinations of 3D object positions can project onto the same 2D image. As categorized previously, the 3D scene reconstruction problem has been tackled in several ways, with methods ranging from using multiple views [15, 17, 16] of the scene to triangulation or computing a space carving, or the use of semantic reasoning for object detection and object pose estimation [2, 4, 14] .

For the problem of class-specific object reconstruction, our goal is to detect, estimate the pose of the object, and reconstruct the 3D structure that is consistent with the 2D image. The challenges primarily stem from the problem's ill-posed nature and lack of knowledge of the shape of the object to be reconstructed. Thus, most techniques in recent times try to solve object pose estimation and 3D object reconstruction jointly, so that one can be used to resolve the ambiguities that arise due to the other [5]. For this purpose, we depart from the beaten path of using plain regression-based models (including deep learning methods) for estimating object pose/shape, to geometrically reason about an object (modeled as a constellation of parts) in 3D, while simultaneously detecting an object, estimating its continuous pose and recovering its underlying 3D shape.

In addition, the state-of-the-art regression models rely heavily on manually designed and annotated 3D CAD models and annotations [1, 2, 3, 14], which are very laborious to generate. Our goal instead is to design algorithms that use arbitrary images while employing geometric reasoning, to minimize the need for manual annotations. We employ Structure from Motion (SfM) [24] and Space Carving [23] techniques on image sequences taken around the object under consideration, to generate 3D meshes that allows us to reason about the shape of the object [5, 19]. Furthermore, we developed pipelines [5, 19] that combine data-driven deep semantic part learning accompanied by principled modeling and effective optimization of the underlying physics, object pose and shape deformation, and scene occlusion. Given a query image, the goal is to simultaneously detect and reason about the object in 3D, to recover its shape and estimate its continuous pose. For this pur-

pose, we propose a two-pronged model comprising of: (a) a *sparse* part model for reasoning about shape and continuous pose, and (2) a *dense* model for reasoning about occlusion and reconstructing comprehensive 3D shape of the object rather than retrieving the closest 3D CAD shape proxy [2].

Reasoning about the object in 3D amounts to only half of the scene understanding problem, whereas the rest is reliant upon reconstructing the whole scene in a manner which may or may not be object-aware. The rest of the research in this dissertation focuses on extending the geometry-driven reasoning for learning methods for holistic scene reconstruction, where we focus on monocular depth prediction on video sequences of street scenes. The aim is to develop geometry-aware end-to-end learning models that learn to predict pixel-wise depth estimates for the entire scene. In recent times, several attempts have been made to incorporate geometric reasoning within deep learning models via loss functions while preserving the end-to-end differentiability. Most importantly the use of using temporal image consistency based loss functions [6, 8], allows the training to be done in a nearly unsupervised setting. In other words, the network can learn to estimate the scene depth just by watching videos [25]. Inspired by [6], we propose Long-Short Term Memory (LSTM)-based and Generative Adversarial Network (GAN)-based deep learning models, that can exploit the sequential nature of the data to perform geometric reasoning about the pose, thereby obtaining state-of-the art results for the depth reconstruction problem.

1.2 Organization of the Dissertation

1.2.1 Class-specific Single View Object Detection, 3D Reconstruction and Pose Estimation

Chapter 2: Class-specific Object Pose Estimation and Reconstruction Using 3D Part Geometry

Formally stated, given an image containing an object of a specific class, the goal is to simultaneously detect and reason about the object in 3D, to recover its shape and estimate its continuous pose, from 2D appearance cues.

The key contributions of the method proposed in this chapter are:

1. Formulation of deformable part-based reasoning about an object. We model the object's shape as a set of 3D part positions instead of performing wholistic reasoning about the entire object. The goal is to better model intra-class appearance and shape variations.

2. Learning of SfM-based [24]class-specific shape and appearance models from real image sequences, instead of manually generating CAD models for objects. CAD models are computationally expensive, painstaking to design [1, 2], and are limited in their capability to capture accurately the 3D shape geometry of the object.

3. Representation of an object's shape by a subspace spanned by basis shapes.

4. Instead of regressing the camera viewpoint or object shape bins [3, 4, 1], the object is modeled geometrically by reasoning about the physical projection of the underlying shape, alongside reasoning about the visibility of 3D parts with respect to the estimated viewpoint, thus allowing for effective estimation of the continuous viewpoint of the object.

Chapter 3: Learning Hierarchical Models for Class-Specific Reconstruction from Natural Data

As an extension to [5] in the Chapter 2, we propose a method that can estimate continuous camera pose and perform deformable 3D object reconstruction using an effective, incremental optimization procedure. Furthermore, the proposed method also reconstructs a full free-form 3D shape rather than retrieving the closest 3D CAD shape proxy [2], as is typical in most state-of-the-art techniques.

The key contributions of the method proposed in this chapter are:

- Formulation of an automatic part discovery framework, that identifies repeatable and salient regions across image sequences as parts.
- Formulation of a two-pronged model comprising of: (a) a *sparse* part model that is computationally efficient for purposes like detection and reasoning about shape and pose, and (2) a *dense* model for reasoning about occlusion and reconstructing comprehensive 3D shape of the object.

- Generation of a comprehensive 3D reconstruction of the object capturing the object's natural deformation alongside modeling of occlusion, as an alternative to retrieving the closest 3D CAD Model.

1.2.2 Monocular Depth Prediction

Predicting the depth map of a scene is often a vital component of monocular SLAM pipelines. Depth prediction is fundamentally ill-posed due to the inherent ambiguity in the scene formation process. Infinite possibilities of scene depth maps can yield a given image, and reconstructing the right depth map is an intractable problem. However, understanding the semantics of objects in an image can help narrow down the possibilities to realistic depth maps.

Chapter 4: A Recurrent Neural Network Architecture for Monocular Depth Prediction

In this chapter, we investigate the power of a recurrent neural network (RNN) architecture for depth prediction in real world scenes, from monocular video sequences as well as single images. The proposed method models the depth of the scene based on a sequence of input images and their corresponding depth maps, learning their spatio-temporal relationship, without requiring explicit definition of the inter-frame geometric consistency or pose-based supervision.

In this chapter,

- By using learning-based models to reason pose temporally, can we effectively model pose without the use of depth for verification?

- We propose a novel convolutional LSTM [26] (ConvLSTM)-based network architecture for depth prediction from a monocular video sequence [20].
- We harness the ability of long short-term memory (LSTM)-based RNNs to sequentially reason and predict the depth map for an image frame as a function of the appearances of scene objects in the image frame as well as image frames in its temporal neighborhood.
- In addition, the proposed ConvLSTM network is also shown to be able to make depth predictions for future or unseen image frame(s) as well.

Chapter 5: Monocular Depth Prediction using Generative Adversarial Networks

We extend current geometry-aware neural network architectures that learn from photoconsistency-based reconstruction loss functions defined over spatially and temporally adjacent images by leveraging recent advances in adversarial learning [21].

Our key contributions in this chapter are:

- Formulation of a generative adversarial network (GAN) that can learn improved reconstruction models, with flexible loss functions that are less susceptible to adversarial examples, using generic semi-supervised or unsupervised datasets.
- Formulation of a generator function in the proposed GAN that learns to synthesize neighbouring images to predict a depth map and relative object

pose, and a discriminator function that learns the distribution of monocular images to correctly classify the authenticity of the synthesized images.

- Formulation of a photoconsistency-based reconstruction loss function that is used to assist the generator function to train well and compete against the discriminator function.

1.2.3 Chapter 6: Practical Applications of Deep Learning for Environmental Monitoring

Object detection in and segmentation of images of natural scenes are challenging problems primarily due to the presence of textureless regions, or lack of shape cues, and appearances of objects with heterogeneous composition of textures.

The key contributions of the method in this chapter are:

- Formulation of a deep learning-based agglomerative clustering-based image segmentation framework that uses spatial and spectral cues alongside textural cues, to reliably segment regions from cluttered images.

- Application of the proposed agglomerative clustering based segmentation framework to a practical problem of reliably identifying and extracting regions containing algal blooms from images of inland water-bodies.

- Extension of the proposed method to detect Cyanobacterial Harmful Algal Blooms (CyanoHABs) from stock photos. CyanoHABs are a type of algal bloom that contaminates water bodies adversely affecting the ecosystem, and leaves the water unusable for human consumption.

- Design of a multi-modal fusion framework that combines multiple modalities (satellite imagery and stock photos) to jointly learn a better representation given the noisy ground truths from both image modalities.

Bibliography

[1] Pepik, B., Gehler, P., Stark, M., and Schiele, B. $3D^2$PM - 3D deformable part models. *Proc. ECCV*, 2012.

[2] Pepik, B., Stark, M., Gehler, P., and Schiele, B. Teaching 3D geometry to deformable part models. *Proc. IEEE CVPR*, 2012.

[3] Tulsiani, S., Malik, J. Viewpoints and keypoints. *Proc. IEEE CVPR*, 2015.

[4] Hejrati, M., and Ramanan, D. Analysis by synthesis: 3D object recognition by object reconstruction. *Proc. IEEE CVPR*, 2014.

[5] Kumar, A.C.S., Bodis-Szomoru, A., Bhandarkar, S.M., & Prasad, M. Class-specific object pose estimation and reconstruction using 3D part geometry *Proc. ECCVW, Geometry Meets Deep Learning*, 2016.

[6] Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. arXiv preprint arXiv:1704.07813.

[7] Kuznietsov, Y., Stckler, J., & Leibe, B. (2017, February). Semi-supervised deep learning for monocular depth map prediction. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6647-6655).

[8] Garg, R., Carneiro, G., & Reid, I. (2016, October). Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proc. European Conference on Computer Vision* (pp. 740-756). Springer International Publishing.

[9] Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems* (pp. 2366-2374).

[10] Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., & Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4040-4048).

[11] DeTone, D., Malisiewicz, T., & Rabinovich, A. (2017). SuperPoint: Self-Supervised Interest Point Detection and Description. arXiv preprint arXiv:1712.07629.

[12] Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., & Rother, C. (2017, July). DSAC-differentiable RANSAC for camera localization. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Vol. 3).

[13] Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., & Fragkiadaki, K. (2017). Sfm-net: Learning of structure and motion from video. arXiv preprint arXiv:1704.07804.

[14] Kar, A., Tulsiani, S., Carreira, J., & Malik, J. (2015). Category-specific object reconstruction from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1966-1974)

[15] Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., & Szeliski, R. (2009, September). Building rome in a day. In Computer Vision, 2009 IEEE 12th International Conference on (pp. 72-79). IEEE.

[16] Furukawa, Y., Curless, B., Seitz, S. M., & Szeliski, R. (2010, June). Towards internet-scale multi-view stereo. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 1434-1441). IEEE.

[17] Agarwal, S., Snavely, N., Seitz, S. M., & Szeliski, R. (2010, September). Bundle adjustment in the large. In European conference on computer vision (pp. 29-42). Springer, Berlin, Heidelberg.

[18] Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics, 31(5), 1147-1163.

[19] Kumar, A, Bhandarkar SM, & Prasad M, "Learning Hierarchical Models for Class-Specific Reconstruction from Natural Data.", IEEE Computer Vision and Pattern Recognition (CVPR), 2018

[20] Kumar, A. C., Bhandarkar, S. M., & Mukta, P. (2018, June). Depthnet: A recurrent neural network architecture for monocular depth prediction. Deep Learning for Visual SLAM,(CVPR) (Vol. 2).

[21] Kumar, Arun CS, Suchendra M. Bhandarkar, and P. Mukta. "Monocular depth prediction using generative adversarial networks." Deep Learning for Visual SLAM,(CVPR), vol. 3, p. 7. 2018.

[22] Kumar, Arun CS, and Suchendra M. Bhandarkar. "A deep learning paradigm for detection of harmful algal blooms." In Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, pp. 743-751. IEEE, 2017.

[23] A. W. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3D model construction for turn-table sequences. In R. Koch and L. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments, LNCS 1506*, pages 155–170. Springer-Verlag, Jun 1998.

[24] Autodesk 123d catch. `https://en.wikipedia.org/wiki/Autodesk_123D`.

[25] Kendall, Alex Guy. "Geometry and Uncertainty in Deep Learning for Computer Vision."

[26] Xingjian, S. H. I., Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." In Advances in neural information processing systems, pp. 802-810. 2015.

Chapter 2

Class-Specific Object Pose Estimation and Reconstruction using 3D Part Geometry

Arun CS Kumar[1], Andras Bodis-Szomuru, Suchendra M. Bhandarkar, Mukta Prasad

[1]First Author. In the Proceedings of European Conference of Computer Vision 2016 (ECCV) workshops, Springer, Cham., Reprinted here with permission of publisher, September, 2016.

2.1 Abstract

We propose a novel approach for detecting and reconstructing class-specific objects from 2D images. Reconstruction and detection, despite major advances, are still wanting in performance. Hence, approaches that try to solve them jointly, so that one can be used to resolve the ambiguities of the other, especially while employing data-driven class-specific learning, are increasingly popular. In this paper, we learn a deformable, fine-grained, part-based model from real world, class-specific, image sequences, so that given a new image, we can simultaneously estimate the 3D shape, viewpoint and the subsequent 2D detection results. This is a step beyond existing approaches, which are usually limited to 3D CAD shapes, regression based pose estimation, template based deformation modelling etc. We employ Structure from Motion (SfM) and part based models in our learning process, and estimate a 3D deformable object instance and a projection matrix that explains the image information. We demonstrate our approach with high quality qualitative and quantitative results on our real world RealCar dataset, as well as the EPFL car dataset.

2.2 Introduction

Despite big advances, core computer vision problems in the area of detection and reconstruction are far from perfectly solved. It is increasingly recognized that to combat the problems faced by these areas of vision, effective solutions must tackle them jointly, modelling the physics of image formation, learn from data, expert-

knowledge and allow one problem to handle the ambiguities of the other. Although 3D geometric reasoning has become increasingly common in several computer vision applications, it is still some way off from becoming a standard consumer-level technique.

In this paper, we propose a framework that, given a 2D image, simultaneously detects an object category instance, estimates the object pose and shape in 3D, reasons about its part appearance and occlusion, thus performing object reconstruction in 3D and detection in 2D, jointly. The proposed framework learns a class-specific, deformable fine-grained, part-based model from image sequences, learning both appearance and geometry. Note that the ill-posed nature of the problem results in a complex solution landscape with several local minima. In order to enable reasonable solutions, we solve the problem by tackling the complexity in a gradual, incremental way. We start from a constrained setup for which the solution can be found reliably and then gradually increase the flexibility in the model to handle more variables in the problem.

The idea of tackling vision problems jointly, has been gaining traction recently [15, 10, 12, 13, 21, 22]. But the modern approaches, while making strides in tackling this problem, have often resorted to using high quality CAD models (which are expensive, painstaking to design, and/or limited in their capability to capture the object shape, appearance, especially the surface texture). Another tendency, is to model camera viewpoint using regression rather than modelling the physical projection process. Also, shape and view variation is often modelled using a bank of representations/templates and/or in a brute force approach. In our proposed

approach, we learn SfM based class-specific shape and appearance models from real image sequences as faithfully as possible. Some supervised input is acquired through minimal, intuitive input for fine-grained part understanding. At test time, we formulate the detection and reconstruction problem in terms of the actual re-projection error (this models the scene physics more accurate than regression) and use a variety of RANSAC-based techniques in order to make estimation efficient and effective. We will expand on the related work in the next section.

2.3 Related Work

As mentioned above, the problem of joint detection, reconstruction and pose estimation of object classes from images has received considerable attention within the computer vision research community in recent years [10, 22, 12, 12]. Existing approaches to solve this problem can be broadly categorized into two main subclasses, *i.e.*, distinctive view-based techniques and 3D geometry-based techniques. Distinctive view-based techniques exploit robust but less descriptive 2D features for view-specific models for detection and recognition [5, 2, 3]. The performance of statistical 2D feature based methods from the computer vision research literature inspired the development of most distinctive view-based techniques. Existing techniques [4, 6, 8] treat viewpoint estimation as a classification problem by dividing the viewpoint range into discrete bins. Ghodrati *et al.* [6] train multiple Support Vector Machine (SVM) classifiers, one for each discrete viewpoint, treating each classifier independently of the others. He *et al.* [8] use a two-step process, wherein

a viewpoint-parametrized classifier is first used to estimate a coarse viewpoint followed by fine-tuning step. Fenzi *et al.* [4] treat continuous viewpoint estimation as a regression problem which is solved using a Radial Basis Function Neural Network (RBF-NN). The RBF-NN is trained to predict the appearance features as a function of the viewpoint. Tulsiani *et al.* [15] train a Convolutional Neural Network (CNN) that can jointly predict the viewpoints for all classes using a shared feature representation. The CNN is used to estimate the coarse viewpoint which is subsequently leveraged for keypoint prediction. Though these view-based methods have been effective, one would expect that accurately modelling the physical projection process would be beneficial.

In recent years, due to the wide availability of affordable depth sensors, 3D shape repositories and 3D CAD models, coupled with the fact that it makes more sense to reason in terms of the underlying 3D structure of the object, the research focus has shifted towards 3D geometry-based techniques for solving the 3D object pose estimation and reconstruction problem. With improved optimization techniques and processing power, we are able to learn these, more powerful models. Pepik *et al.* [12, 13] extended the Deformable Parts Model (DPM) [3] to represent the part locations and deformations in 3D. Yu *et al.* [17] on the other hand, propose an approach for learning a shape appearance and pose (SAP) model for both 2D and 3D cases, where the training instances with unknown pose are used to learn a probabilistic object part-based model. The class label and the pose of the object are inferred simultaneously by joint discovery of parts and alignment to a canonical pose. Xiao *et al.* [16] and Kim *et al.* [11], exploit synthetic 3D models

to incorporate 3D geometric information into the DPM framework [3] for pose estimation. More recently, Choy *et al.* [1] use Non-Zero Whitened Histogram-of-Gradients (NZ-WHO) features [7] to synthesize, on the fly, discriminative appearance templates from 3D CAD models, for several poses, scales on multiple CAD model instances of the object, to jointly estimate the viewpoint and the instance associated with the object. In particular, Pepik *et al.* [12, 13] rephrase the DPM framework [3] to formulate a structured learning output predictor to estimate the 2D bounding box of the object along with its viewpoint by enriching the object appearance model using 3D CAD data. The combination of robust DPM matching with the representational power of 3D CAD models is shown to result in a boost in performance across several datasets. We aim to extend this work by learning from real, SfM shapes and associated image appearance models and also treat viewpoint using a full projection model instead of regression.

There has been progress in this regard. Hejrati and Ramanan [9] learned the 3D geometry and shape of the object from 2D part annotations using a non-rigid SFM technique. In particular, Hejrati and Ramanan [10] represent 2D object part appearances using a Gaussian mixture model (GMM) that captures the appearance variations due to variations in the viewing angle. Zia *et al.* [18] use a 3D shape representation scheme to jointly model multiple objects allowing them to reason about inter-dependencies between the objects, such as occlusion, in a more deterministic and systematic manner.

Our proposed method departs from the beaten path described above, through the following means: (i) employing automatically estimated, real world 3D shapes

to learn deformable models (manually generated 3D CAD models are often lacking in appearance details (such as surface texture) and make simplifying approximations about the actual 3D geometry that undermine the challenges underlying the 3D object pose estimation and reconstruction problem), (ii) modelling the projection process for geometric reasoning instead of relying on regression models, (iii) solving the shape recovery and view estimation problems using an effective RANSAC based scheme (as opposed to the computationally intensive generative process of [10]) and (iv) using a fine-grained part representation, learnt from real data, to model the shape to a high resolution and accuracy for more complex analysis in the future. The pipeline of the proposed RANSAC based scheme for shape recovery and viewpoint estimation is shown in Fig. 2.1.

2.4 The Proposed Method

2.4.1 Problem Statement

Given a set of image sequences of the same object class, *e.g.* cars, each sequence being taken around a single object instance, our objective is to reconstruct the shape and pose of a new instance observed in a new input image. More precisely, we aim to learn a deformable shape model for the particular object class, which then allows us to estimate both the best deformed shape and the 3D object pose in a new image such that visible semantic, salient parts of the object project to their 2D observations in the image. The latter also involves an occlusion reasoning for the new viewpoint and object instance.

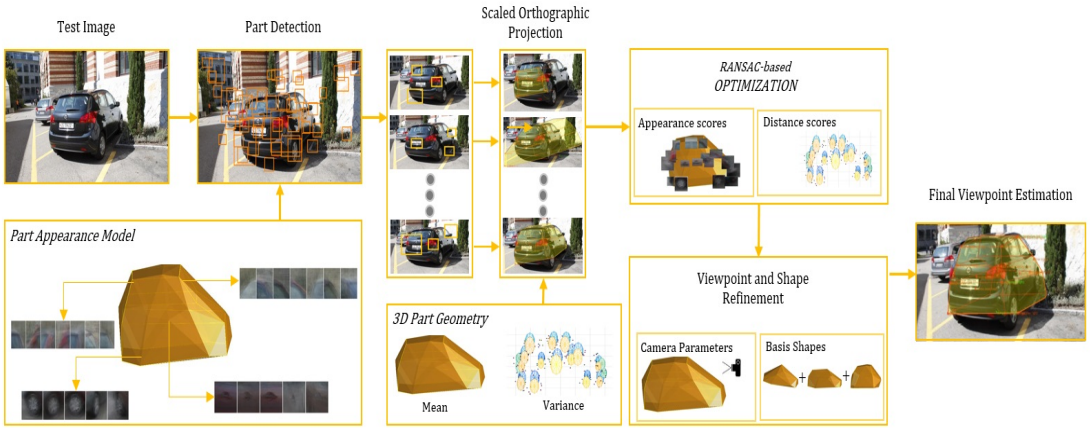

Figure 2.1: ***The proposed object shape and pose (or viewpoint) estimation pipeline.*** Given a test image, we perform part candidate detection using the learned mixture-model-based part appearance model, followed by viewpoint (scaled orthographic cameras) estimation using a RANSAC-based scheme. The optimization gradually fits more deformation to the shape to recover a realistic reconstruction with a refined camera estimate.

To formulate this, we will use the following notation. Each of the K uncalibrated image sequences given in the training set, indexed by $k \in \{1 \dots K\}$, contains N_k images taken around an object instance. Let us denote the n-th image in the k-th sequence by I_{nk}, and its associated 3×4 camera (projection) matrix by $\mathbf{C}_{nk}$, which encodes the relative object pose and the camera intrinsics. The estimation task is to predict the full projection model $\mathbf{C}$ and the 3D shape S of the object in a new uncalibrated input image I. For simplicity, we define the 2D object detection mask in I as projection of the fitted shape instance through the estimated camera.

2.4.2 A Class-Specific Deformable Model

There are different ways to represent the shape of an object instance that is of a particular object class. Due to its simplicity and elegance, we have chosen to use a semantic part-based construction in combination with a linear subspace deformation model.

We define the shape S of any object instance via the 3D positions of its P semantic parts in space. The set of parts is predefined per object class. If $\mathbf{s}_p$ is the position of the p-th part of an object instance, then the shape of this instance can be encoded by a $3 \times P$ matrix $\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \dots \mathbf{s}_P \end{bmatrix}$. The linear subspace model describes any shape as a linear combination of a set of L known basis shapes which capture the modes of variation in the training data. Thus, the shape matrix of a particular object instance is $\mathbf{S} = \sum_{l=1}^{L} \alpha_l \mathbf{B}_l$, where $\mathbf{B}_l$ is the $3 \times P$ matrix of a basis shape and α_l is the corresponding coefficient.

Assume that the basis shapes $\{\mathbf{B}_l\}_{l=1}^{L}$ are known from a training phase for an object class for now. Then given a new image I depicting an instance of the same object class, the objective is to compute the shape matrix $\mathbf{S}$ of the depicted instance, as well as the camera (projection) matrix $\mathbf{C}$ that maps 3D parts of the object to its observation in the new image I. The 2D projection $\hat{\mathbf{x}}_p$ of a 3D part location $\mathbf{s}_p$ can be formulated as

$$\hat{\mathbf{x}}_p = \rho(\mathbf{C} \cdot \mathbf{s}_p) = \rho\left(\mathbf{C} \sum_{l=1}^{L} \alpha_l . \mathbf{b}_{lp}\right) \tag{2.1}$$

where $\mathbf{b}_{lp}$ is the p-th column of basis shape matrix $\mathbf{B}_l$, $\rho(.)$ is a mapping that maps any vector (u, v, w) with $h \neq 0$ to $(u/w, v/w)$. The camera matrix $\mathbf{C}$ can describe a perspective or an orthographic projection. However, not all points on the surface of an object are visible in an image. The binary visibility state of a 3D point $\mathbf{s}$ in an image I of camera matrix $\mathbf{C}$ is modeled by a boolean variable $v(\mathbf{s}, \mathbf{C}) \in \{0, 1\}$, where 0 stands for *occluded* and 1 for *visible*.

Given the matrices of the basis shapes $\{\mathbf{B}_l\}$, the shape of an object instance is fully determined by its deformation parameters $\{\alpha_l\}$. The loss function for computing the shape matrix $\mathbf{S}$ and the camera matrix $\mathbf{C}$ of an object instance depicted in a query image I can be defined as the sum-of-squared Euclidean distances between the projections and the observations $\mathbf{x}_p$ of the visible object parts in image I:

$$L(\{\alpha_l\}, \mathbf{C}) = \sum_{p=1}^{P} v(\mathbf{s}_p, \mathbf{C}) \cdot ||\mathbf{x}_p - \rho(\mathbf{C} \cdot \mathbf{s}_p)||^2, \quad \mathbf{s}_p = \sum_{l=1}^{L} \alpha_l \mathbf{b}_{lp}, \tag{2.2}$$

where the vectors $\mathbf{b}_{lp}$ are known from the training phase. The joint shape-pose problem for an input image I can be solved by a minimization of L with respect to the shape coefficients $\{\alpha_l\}$ and projection parameters $\mathbf{C}$.

The loss function for the training phase can be obtained in a similar fashion. There, the squared projection errors of K object instances needs to be measured over all images of the training set. The loss function for training can be written

as

$$L_T = \sum_{k=1}^{K} \sum_{n=1}^{N_k} \sum_{p=1}^{P} v(\mathbf{s}_{kp}, \mathbf{C}_{nk}) \cdot ||\mathbf{x}_{klp} - \rho(\mathbf{C}_{nk} \cdot \mathbf{s}_{kp})||^2, \;\; \mathbf{s}_{kp} = \sum_{l=1}^{L} \alpha_{kl} \mathbf{b}_{lp}, \qquad (2.3)$$

where $\mathbf{s}_{kp}$ is the 3D location of the p-th part of the k-th object instance, and $\mathbf{C}_{nk}$ is the camera matrix corresponding to the training image I_{nk} as introduced in Sect. 2.4.1.

In the followings, we present our approach for learning the basis shapes and part appearance from multi-view 3D mesh reconstructions of our input sequences.

2.4.3 From Dense 3D Reconstructions to Part-Based Shape Models

In order to learn the 3D basis shapes, a 3D surface model of each object instance of the training set is needed. Moreover, we will augment the shape model with an image-based appearance model per object part (Section 2.4.5). This requires the additional knowledge of all camera matrices $\mathbf{C}_{nk}$ for the training images I_{nk}. We now discuss how these prerequisites are obtained and postpone the learning algorithms to Sections 2.4.4 and 2.4.5.

Prior to training, we first apply a state-of-the-art 3D reconstruction pipeline to each sequence, separately. A Structure-from-Motion (SfM) procedure computes the camera matrices $\mathbf{C}_{nk}$, while a dense Multi-View Stereo (MVS) and surface reconstruction algorithm computes a triangle mesh surface of the visible surface areas of the scene, given the camera models. We use 123DCatch that integrates

all these steps, but note that other similar tools are also possible here. As a result, each 3D object instance in the training set is reconstructed as a mesh with an arbitrary number of vertices and triangles (see Figure 2.2). Intra-class variations and the varying vertex counts make meshes difficult to relate, not to mention that most vertices may not correspond to any salient entity on the object surface or its corresponding images.

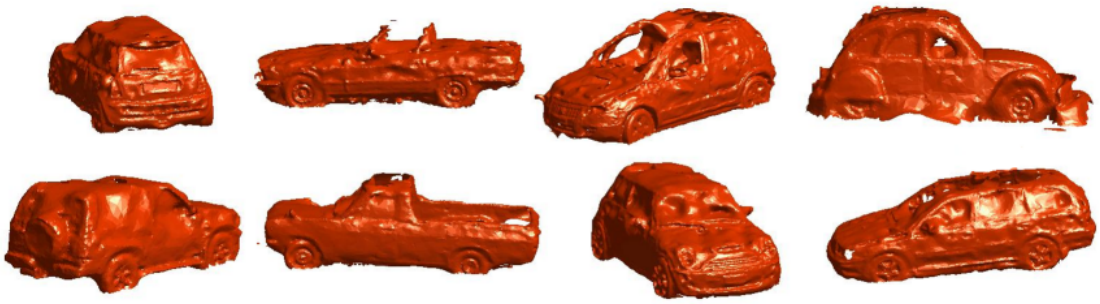

Figure 2.2: Training set examples for 'car': 3D meshes obtained from real-world 2D image sequences from 123DCatch. These models are used for data-driven 3D geometric reasoning throughout the paper. Note the intra-class shape variability.

In a subsequent step, we annotate each 3D mesh (Fig. 2.2) with a fixed set of parts (up to the closest vertex location), where each part is a repeatable and semantically meaningful region of the object, *e.g.* (center of) *front-left-wheel* or *rear-licence-plate*. The 3D part annotations are obtained via an intuitive user interface by performing a multi-view triangulation of part annotations from two or more images observing the same object instance. As a result, each object instance (indexed by $k \in \{1 \dots K\}$) yields an ordered set of 3D object part locations $\{\mathbf{s}_{kp}\}_{p=1}^{P}$.

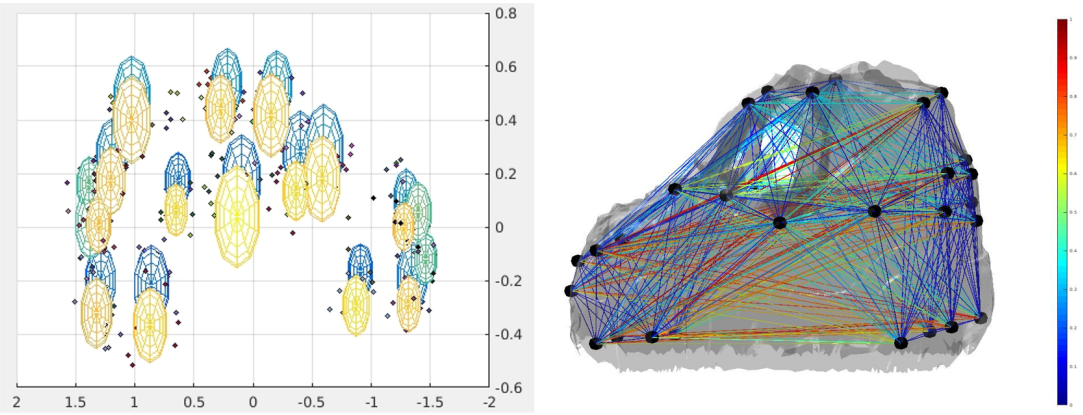

Figure 2.3: 3D Part Geometry. *Left:* The standard deviation of part location is plotted in spheres (yellow on car's right, blue on the left). Interestingly, the front door handles vary considerably in location, while the bumpers and lights, not so much. *Right:* Variances in the mutual distance between each pair of parts are overlaid on a sample graph (*red* corresponds to higher variation, *blue* corresponds to lower variation).

Once the 3D meshes are annotated, the coordinate frames can be aligned using the part annotations. Due to the shape variations, this gives a more accurate alignment than simply applying the Iterated-Closest-Point (ICP) algorithm in our experience. Figure 2.3 shows the scatter of object part locations (across training instances), as well as the covariance ellipsoids (corresponding to 1σ) to visualize intra-class shape variations in our example training set.

2.4.4 Learning a Class-Specific Object Shape

Based on the 3D shape model discussed in Section 2.4.2, we perform a Principal Component Analysis (PCA) on the object part positions and retrieve the top M

modes of deformation ($M = 4$ in all our experiments), which gives us a set of $L = M + 1$ basis shapes (where $\mathbf{B}_1$ is explicitely defined as the mean shape) for an effective and compact linear subspace model to describe the subspace of possible intra-class shapes.

2.4.5 Learning the Appearance of Object Parts

The shape bases define a subspace of possible shapes for a particular object class. However, we also need to understand the appearance of the class in order to efficiently relate the shape model to new images. For each object part in the 3D shape representation, we construct an appearance model.

For the training sequences, by estimating visibility and projection, an appearance model for each part is learnt from the ground truth image sequences under real illumination, projection *etc*. For every part, CNN features (*conv5* layer) are extracted from the input images at their projections (when visible), using publicly available network weights [14]. These weights are obtained via training on the ImageNet Challenge 2014 (ILSVRC) dataset based on the part annotations. A mixture model [3, 10] over these CNN features is then used to represent the variation in appearance, viewpoint *etc*. We learn a binary SVM classifier for each mixture component of each part of the class, to act as a part detector in images.

2.4.6 Detecting an Object Shape and Pose in a Query Image

Given a new query image I, and the learnt shape subspace spanned by basis shapes $\{\mathbf{B}_l\}$, and given the appearance-based object part detectors based on deep features

and on SVM classifiers, our goal is to jointly fit the deformable shape model and compute the camera matrix $\mathbf{C}$ for this image such that 3D part locations of the fitted 3D shape model project to corresponding part observations in the image. The corresponding loss function is formulated in Eq. 2.2. The proposed pipeline is outlined in Algorithm 1 (which also invokes Algorithm 2).

Algorithm 1 Shape recovery, pose estimation and detection

1: **Part Detection**. Possible candidates for part detections are collected by convolving the trained SVM weight filters on conv5 feature pyramids [23]. Filter responses across multiple scales are combined using Non-Maxima Supression followed by Platt's Scaling [20] to obtain the probabilities of positive responses, such that the responses of different SVM classifiers are comparable. Responses stronger than a certain probability (p=0.35) are considered plausible candidates for the next step.

2: **Viewpoint Estimation**. We find the best camera parameters to project the mean shape to the test image by performing a RANSAC-based view estimation routine explained in Algo.2. In this case, the minimal set needs to be size 3 and the unknown parameters correspond to those of scaled orthographic projection.

3: **Viewpoint and Shape Refinement**. We perform a subsequent pass of viewpoint refinement allowing for shape deformation. This is equivalent to optimizing Eq. 2.2, with respect to the deformation parameters $\{\alpha_l\}_{l=1}^{L}$ in addition to the scaled-orthographic camera parameters. The RANSAC-based procedure can be repeated, but in each pass, one more mode of shape deformation is considered for a stable, incremental optimization. Finally, the a minimal set of 5 2D part candidates is needed for estimation of the extra $L-1=4$ basis shape weights. The optimization of the loss function in Eq. 2.2 is modified to reflect the new parameters.

4: **Object Mask**. The estimated deformable shape and camera parameters represent the best reconstruction estimate for this image. When projected to the image, this gives us an object detection silhouette for this image.

Algorithm 2 RANSAC-based Viewpoint Estimation Algorithm

1: Perform part detection using the trained part appearance classifiers to obtain *Filter Response* $\mathcal{F}$, on the test image. Threshold these to obtain a set of possible candidates.
2: **for** N iterations **do**
3: Assemble a minimal set of randomly-sampled unique parts from the candidates (constraint: they must be simultaneously visible in at least one view).
4: Fit the unknown parameters minimizing the projection loss between the mean 3D shape parts corresponding to the 2D minimal set selected above.
5: Check for inliers, based on whether candidate detections are within threshold τ_1 for the remaining visible parts projected according to the above derived projection.
6: If the number of inliers are greater than τ_2 then store this minimal set and the estimated parameters.
7: For the set with maximum inliers, re-estimate the parameters minimizing the projection loss, through least-squares fitting on all the inliers, instead of only the minimal set. This is the best parameter estimate.

2.5 Evaluation

2.5.1 Dataset

Our RealCar dataset consists of 35 image sequences taken around unique and distinct instances of cars, captured in real world conditions with challenging variations in scale, orientation, illumination with instances of occlusion. The total number of images per sequence varies between 30 and 115, across the dataset. When an SfM method like 123DCatch is used to estimate the car shapes and camera matrices, we get full mesh shapes along with full projection matrices (see Fig. 2.2). We use 29 of these sequences (and associated SfM results) for training and reserve 6 for testing.

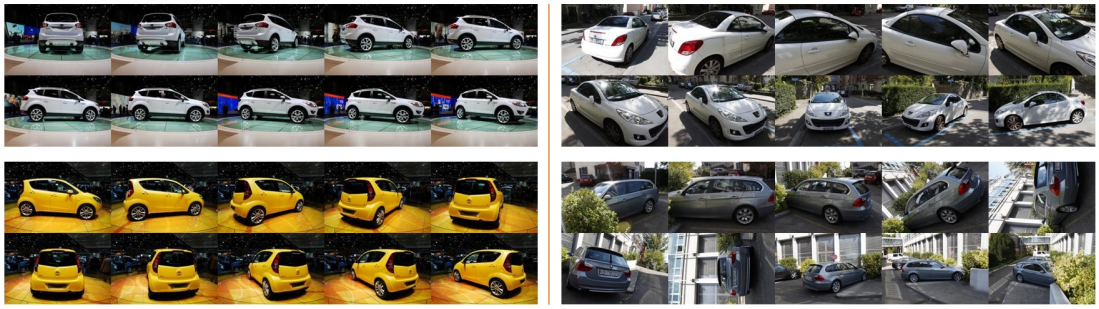

Figure 2.4: Examples of 2D image sequences from the EPFL Multi-view Cars dataset (*Left*) and our RealCar dataset (*Right*).

The EPFL Multi-view cars dataset [26] contains image sequences of car instances on a turntable. Such sequences do not respond well to SfM pre-processing like RealCar dataset as the scene is not rigid, so this provide images to test on, but no ground truth 3D meshes or part annotations. This dataset is used purely as a second test set of images.

2.5.2 Experimental setup

In this section, we evaluate the performance of our approach based on two tasks, (1) Viewpoint Estimation, to measure the accuracy of the estimated camera projection and, (2) Reconstruction, to measure how well the shape of the object in the test image is recovered.

Viewpoint Estimation:

In order to evaluate the viewpoint estimation performance of the proposed approach, we run Algorithm 1 and report the Mean Precision of Pose Estimation [19] and Mean Angular Error [27], on individual images from the 6 test sequences of our RealCar dataset as well as from all 20 sequences of the EPFL Multi-view Cars dataset [26], where each car is imaged over a complete 360 degrees, with approximately one image for every 3-4 degrees. To measure viewpoint estimation accuracy we report our results using two standard metrics, Mean Precision of Pose Estimation (MPPE) [19] and Mean Angular Error (MAE) [27]. To report MPPE, we discretize azimuth angles (ϕ) into k number of bins where $k \in \{8, 12, 16, 18, 36\}$ and compute the precision of the viewpoint estimation for different number of bins. Table 2.1 shows the MPPE obtained using our approach on both images from our RealCar dataset and the EPFL dataset, and compares with Pepik et. al. [13] and Ozuysal et. al. [26], on EPFL dataset. Similarly, the Mean Angular Error [27], to evaluate the continuous viewpoint estimation performance of the proposed system, on both datasets is shown in Table 2.2 in comparison with Pepik et. al. [13] and Glasner et. al. [27] on the EPFL Multi-view cars dataset. In addition to estimating the Mean Angular Error for predicting the azimuth angle, we also estimate MAE for predicting all *3 Euler angles* [15], to provide a more accurate measure of performance of the proposed approach, for continuous viewpoint estimation. Table 2.3 shows MAE (Mean Angular Error) computed by estimating all *3 Euler angles*.

θ	RealCar Dataset		EPFL-Multiview Cars Dataset [26]		
	(Ours) Training set	(Ours) Test set	(Ours)	3D^{2}PM-D [13]	Ozuysal et. al. [26]
$\pi/4$	93.79	86.09	59.86	78.5	-
$\pi/6$	89.44	79.13	50.06	75.5	-
$\pi/8$	83.85	71.30	40.47	69.8	41.6
$\pi/9$	78.26	65.22	36.67	71.8	-
$\pi/18$	46.58	43.48	19.22	45.8	-

Table 2.1: Viewpoint Classification Accuracy using MPPE [19] on our RealCar dataset *(left)*, and on EPFL Multi-view Cars dataset [26] *(right)*. For our dataset, in addition to the test set, pose estimation experiments are also conducted on a subset of the training set to demonstrate the performance of the proposed approach in estimating viewpoint & recovering shape, on images, where the part detection accuracy is quite high.

θ	RealCar Dataset		EPFL-Multiview Cars Dataset [26]		
	(Ours) Training set	(Ours) Test set	(Ours)	3D^{2}PM-D [13]	Glasner et. al. [27]
$\pi/4$	13.02	14.13	17.35	12.9	24.8
$\pi/6$	11.88	12.35	13.58	9.0	-
$\pi/8$	11.05	10.87	10.68	7.2	-
$\pi/9$	10.32	9.92	9.58	6.2	-
$\pi/18$	5.47	6.2	4.81	5.2	-

Table 2.2: Continous/Fine-Grained Viewpoint Estimation error using MAE [27] on our dataset *(left)* and on EPFL Multi-view Cars dataset [26] *(right)*.

θ	Our Dataset		EPFL-Multiview Cars Dataset [26]
	Training set	Test set	(Ours)
$\pi/4$	16.08	18.28	31.48
$\pi/6$	14.92	16.31	22.71
$\pi/8$	13.15	14.45	17.27
$\pi/9$	12.32	13.54	15.06
$\pi/18$	6.92	6.59	8.05

Table 2.3: Continous/Fine-Grained Viewpoint Estimation using our Ransac-based viewpoint estimation technique, MAE [27] on EPFL Cars dataset [26] by computing all *3 Euler angles*.

The result tables show that our method performs very well on our dataset and competes well with the state of the art on the EPFL dataset, despite training on a smaller dataset appearance-wise. We report the viewpoint estimation accuracy on our dataset as well as on EPFL Multi-view cars dataset, we used our dataset (barely 29 3D object instances) to learn part appearances and 3D part geometry, and test it on EPFL Multi-view cars dataset. The performance of our approach relies heavily on the part detection performance generating inliers for at least a few parts. If part detections are even reasonable, the viewpoint/shape estimation is generally accurate, and so the accuracy on the RealCar dataset tends to be high (running our approach on the data that it has been trained on, shows best case results and an upper bound on how well our algorithm can do, due to the familiarity with appearance, though projection must still be figured out). The experiments show that, most of the bad viewpoint estimations are mainly due to

Figure 2.5: Qualitative results of the proposed *RANSAC*-based Viewpoint Estimation and Shape Recovery, on EPFL Multi-view Car dataset. Odd columns illustrate the test image with corresponding Viewpoint/Shape estimations overlayed on it. Even columns illustrate the Viewpoint Estimation of their corresponding test image (on its left), using a sample mesh (from our dataset) for better visualization. (*note:* meshes (in even columns) are not generated/reconstructed by our viewpoint estimation approach, and are used only for the purpose of better visualization in all our qualitative results).

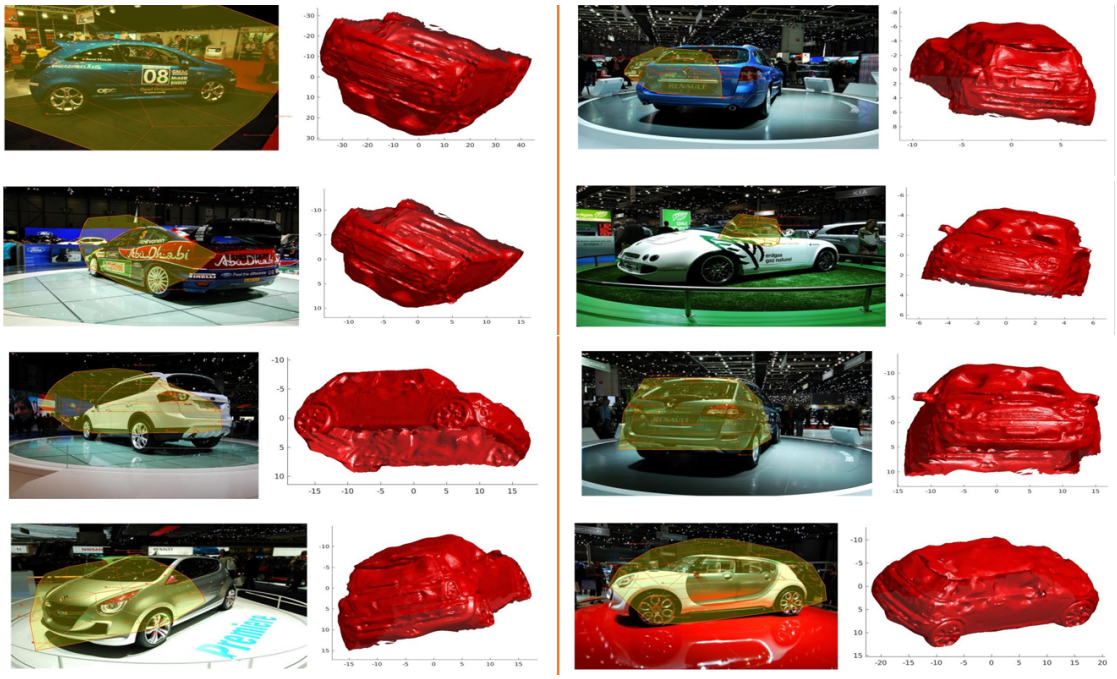

Figure 2.6: A qualitative illustration on how the failure of part detection and the effect of symmetry in part appearances affect the the viewpoint estimation performance.

Figure 2.7: Qualitative results of the proposed *RANSAC*-based viewpoint estimation on *our* dataset, with Viewpoint/Shape estimations overlayed on the object.

Figure 2.8: An illustration on the improvement in Viewpoint and Shape Estimation due to the *Viewpoint and Shape refinement* step. Each pair of the image represents the Viewpoint and Shape estimations before *(left)* and after *(right)* the *Viewpoint and Shape refinement* step. The shapes on the right (of each pair) tend to be more compact and has a better viewpoint estimate, than the ones on the left.

bad part detection performance as shown in shown in Fig. 2.6 or mistakes due to symmetry of the car class.

Another important factor that affects the viewpoint estimation performance of our approach is the lack of a strong global appearance prior or a root filter. Unlike other regression based methods, we solely rely on detected 2D part locations for reasoning the 3D shape of the object, where slight anomalies with one or more part detections can cause a considerable error in the estimated final viewpoint. In the future, we will train robust part appearance classifiers over more appearance data with hard-mined data negatives, along with strong root filters, to try improving part detection accuracy and performance.

Fig. 2.5 shows qualitative results on the EPFL dataset. Fig. 2.6 demonstrates the challenges of part detection and appearance symmetry in viewpoint estimation success. Fig. 2.7 shows the viewpoint/shape recovery results on our dataset. Also Fig. 2.8 compares the shape recovery results before and after the viewpoint and shape refinement step.

Reconstruction

Unlike EPFL dataset, the RealCar dataset has the ground truth 3D parts annotated, so we can qualitatively compare the estimated 3D part based model with its actual ground truth, to evaluate shape accuracy. To report the shape recovery performance of our approach, we computed the average sum of squared distances between the estimated and ground truth 3D part locations of the object in the test image, for the 3D parts normalized to unit scale. The reconstruction/shape

recovery error is 0.07 on the training set and 0.082 on test set of the RealCar dataset.

2.6 Conclusion and Future Work

We have shown qualitatively that our method for class-specific shape detection, recovery and pose estimation can yield good results on unseen state-of-the-art data as well as the original training data. We expect our RANSAC based process to be faster, while still efficient, than brute force exhaustive search and also models the projection process more accurately than regression. The fine-grained part representation and linear subspace representation allows us to model deformation effectively, but work with far fewer vertices than an SfM mesh with thousands of vertices. Importantly, we aim to learn such a part representation automatically, and automatically warp and improve the full mesh reconstructions also. As mentioned, better training and engineering should help perform even better. Going forward, using more image evidence (edges, contours, textures *etc.*) to fit the camera projection and reconstruction parameters, should allow for more accurate estimation. We could also perform GraphCut based segmentations for improved detection outlines.

Bibliography

[1] Choy, C. B., Stark, M., Corbett-Davies, S., and Savarese, S. Enriching object detection with 2D-3D registration and continuous viewpoint estimation. *Proc. IEEE CVPR*, 2015.

[2] Felzenszwalb, P.F., and Huttenlocher, D.P. Pictorial structures for object recognition. *IJCV*, 61(1), 55-79, 2005.

[3] Felzenszwalb, P.F., Girshick, R., McAllester, D., and Ramanan, D. Object detection with discriminatively trained part based models. *IEEE T-PAMI*, 2009.

[4] Fenzi, M., Leal-Taix, L., Ostermann, J., and Tuytelaars, T. Continuous pose estimation with a spatial ensemble of Fisher regressors. *Proc. ICCV*, 2015.

[5] Fergus, R., Perona, P., and Zisserman, A. Object class recognition by unsupervised scale-invariant learning. *Proc. IEEE CVPR*, 2003.

[6] Ghodrati, A., Pedersoli, M., and Tuytelaars, T. Is 2D information enough for viewpoint estimation? *Proc. BMVC*, 2014.

[7] Hariharan, B., Malik, J., and Ramanan, D. Discriminative decorrelation for clustering and classification. *Proc. ECCV*, 2012.

[8] He, K., Sigal, L., and Sclaroff, S. Parameterizing object detectors in the continuous pose space. *Proc. ECCV*, 2014.

[9] Hejrati, M., and Ramanan, D. Analyzing 3D objects in cluttered images. *Proc. NIPS*, 2012.

[10] Hejrati, M., and Ramanan, D. Analysis by synthesis: 3D object recognition by object reconstruction. *Proc. IEEE CVPR*, 2014.

[11] Lim, J.J., Khosla, A., and Torralba, A. FPM: Fine pose parts-based model with 3D CAD models. *Proc. ECCV*, 2014.

[12] Pepik, B., Gehler, P., Stark, M., and Schiele, B. $3D^2$PM - 3D deformable part models. *Proc. ECCV*, 2012.

[13] Pepik, B., Stark, M., Gehler, P., and Schiele, B. Teaching 3D geometry to deformable part models. *Proc. IEEE CVPR*, 2012.

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

[15] Tulsiani, S., Malik, J. Viewpoints and keypoints. *Proc. IEEE CVPR*, 2015.

[16] Xiao, J., Russell, B., and Torralba, A. Localizing 3D cuboids in single-view images. *Proc. NIPS*, 2012.

[17] Yu, T-H. *Classification and pose estimation of 3D shapes and human actions.* Ph.D. Thesis, University of Cambridge, 2013.

[18] Zia, M. Z., Stark, M., and Schindler, K Are cars just 3D boxes? Jointly estimating the 3D shape of multiple objects. *Proc. IEEE CVPR*, 2014.

[19] Lpez-Sastre, R. J., Tuytelaars, T., Savarese, S. Deformable part models revisited: A performance evaluation for object category pose estimation. *Proc. IEEE ICCVW*, 2011.

[20] Platt, John. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 1999.

[21] Zia, M. Z., Stark, M., Schiele, B., Schindler, K. Detailed 3d representations for object recognition and modeling. *IEEE Transactions on PAMI*, 2013.

[22] Kar, A., Tulsiani, S., Carreira, J., Malik, J. Category-specific object reconstruction from a single image. *IEEE CVPR*, 2015.

[23] Girshick, R., Iandola, F., Darrell, T., Malik, J. Deformable part models are convolutional neural networks. *IEEE CVPR*, 2015.

[24] Yingze Bao, S., Chandraker, M., Lin, Y., Savarese, S. Dense object reconstruction with semantic priors. *IEEE CVPR*, 2013.

[25] Khosla, A., Zhou, T., Malisiewicz, T., Efros, A. A., Torralba, A. Undoing the damage of dataset bias. *IEEE ECCV*, 2012.

[26] Ozuysal, M., Lepetit, V., Fua, P. Pose estimation for category specific multi-view object localization. *IEEE CVPR*, 2009.

[27] Glasner, D., Galun, M., Alpert, S., Basri, R., Shakhnarovich, G. Viewpoint-aware object detection and pose estimation. *IEEE ICCV*, 2011.

Chapter 3

Learning Hierarchical Models for Class-Specific Reconstruction from Natural Data

Arun CS Kumar[1], Suchendra M. Bhandarkar, Mukta Prasad

[1]First Author. In the Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR) 2018 workshops, Reprinted here with permission of publisher, April, 2018.

3.1 Abstract

We propose a novel method for class-specific, single-view, object detection, pose estimation and deformable 3D reconstruction, where a two-pronged (sparse semantic and dense shape) representation is learned from natural image data automatically. Then, given a new image, it can estimate camera pose and deformable reconstruction using an effective, incremental optimization. Our method extracts a continuous, scaled-orthographic pose (without resorting to regression and/or discretized 1D azimuth-based representations). The method reconstructs a full free-form shape (rather than retrieving the closest 3D CAD shape proxy, typical in state-of-the-art). We learn our two-pronged model purely from natural image data, as automatically and faithfully as possible, reducing the human effort and bias typical to this problem. The pipeline combines data-driven deep learning based semantic part learning with principled modelling and effective optimization of the problem's physics, shape deformation, pose and occlusion. The underlying sparse (part-based) representation of the object is computationally efficient for purposes like detection and discriminative tasks, whereas the overlaid dense (skin like) representation, models and realistically renders comprehensive 3D structure including natural deformation, occlusion. The results for the car class are visually pleasing, and importantly, outperform the state-of-the-art quantitatively too. Our contribution to visual scene understanding through the two-pronged object representation shows promise for more accurate 3D scene understanding for real world applications on virtual/mixed reality, autonomous navigation, to cite a few.

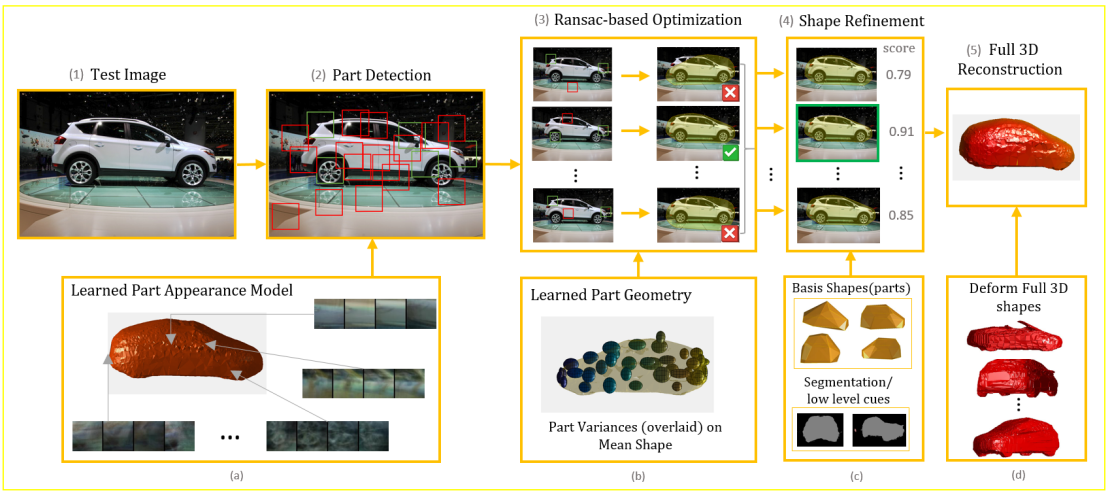

Figure 3.1: Pipeline: Given a set of class image sequences, a two-pronged (dense shape based + sparse part based) model is learned. The part based geometry and appearance model is learned as in (a, b). At test time, the part based model is used to bootstrap the view and deformation parameters in an initial optimization (3), followed by a refinement of these parameters and shape according to the dense shape model (4), to reconstruct a full 3D mesh (5).

3.2 Introduction

The ability to jointly reason about the shapes, viewpoints and locations of objects and their relative interactions from input images, is fundamental to almost all strands of research in computer vision, *e.g.*, object recognition, scene reconstruction, content-based retrieval and problem parametrization. Each individual research strand in computer vision, has been pushed to its limits in terms of peak performance in the past few years; hence the final lap of progress lies in being able to perform joint reasoning and scene understanding. This will enable applications at the intersection of vision and robotics, *e.g.* robotic navigation, autonomous driving *etc.*, to become more "intelligent", robust and efficient. The need for joint scene understanding has been long recognized but progress has been made in only small and steady steps owing largely to the fact that the underlying computer vision problems are ill-posed (many combinations of 3D object shapes and viewpoints can result in a given image), entail a combination of discrete and continuous variables and the input image data is often contaminated by noise with complex statistics that is difficult to model. However, the information extracted while solving one problem, such as object recognition, can help reduce the ambiguities in another, such as scene reconstruction. Class-specific treatment provides a useful, sensible prior, *e.g.*, 3D instances of the *car* class share a similar topology and a learnable family of shapes and appearances. Additionally, recent advances in machine learning and optimization, allow the exploration of more powerful mathematical models and data-driven approaches for joint scene understanding.

As detailed in the abstract, we propose a novel approach to learn a two-pronged class representation for jointly solving class-specific object detection, pose estimation and deformable 3D reconstruction from a single 2D image. At test time, given an unseen image, the sparse model allows us to detect parts efficiently, reason shape deformation and model occlusion (self-occlusion), to estimate pose and recover the underlying 3D shape of the object. The dense model, subsequently, allows us to build on the recovered sparse representation to render a detailed 3D reconstruction of the object.

What's more, this work has implications on many real world applications such as VR/Motion capture games, self-driving cars *etc*. Before delving further, we will now discuss some related work and to better highlight how we adapt and advance it.

3.3 Related Work

Object detection relies on powerful mathematical models that can represent the object shape, camera viewpoint and input noise effectively, and can be learned and deployed effectively. Early approaches to object detection and recognition employed sparse, 2D object representations (with considerable hand-crafted elements and some parameter learning) typically in the form of templates, pictorial structures [9] and constellations [11]). As the field progresses steadily (see PASCAL VOC [7]), abstract, less supervised and more data-driven representations [19] and joint object detection and object reconstruction [2, 15, 26] are being increasingly

explored. In this work, we exploit the progress in deep learning [35] to discover 3D parts (spatial distribution and image appearance), characteristic of the class from real data, registered with respect to their dense shape reconstructions, thus providing a mapping between the semantics and shape. This extends the models of [11, 2, 26] to a more comprehensive 3D part representation derived fully and automatically from 2D data.

Interpreting and handling camera viewpoint well is crucial, even if it is only a means to an end (of recognition/reconstruction). Initial attempts at 3D object recognition/reconstruction were limited to frontal views [21, 30]; each new viewpoint used a new model [10]. Multiple viewpoints were often handled using discriminative, inverse modeling-based approaches based on classification [26] or regression [27]), wherein viewpoint was a 1D (sometimes 2D) variable. While these have performed well in limited evaluations, we model the physics of projection more faithfully (also see [15, 36]) and show that a 6-DOF, continuous, scaled orthographic projection, can be solved effectively in a RANSAC-based perspective-n-point-like ([16]) framework, rather than resorting to A* search or regression/classification.

Reconstruction (when combined with recognition) has spanned many ideas, from warping a shape [2] to approximative, coarse, depth prediction-based models [33] to wireframe reconstructions [22, 36, 15] over the years. In recent work [26, 36], reconstruction is approximated by retrieving a reasonably similar CAD model. Modelling deformation in a linear subspace from annotated data or 3D CAD datasets has also been proposed [15, 36, 26]. The deformation is estimated us-

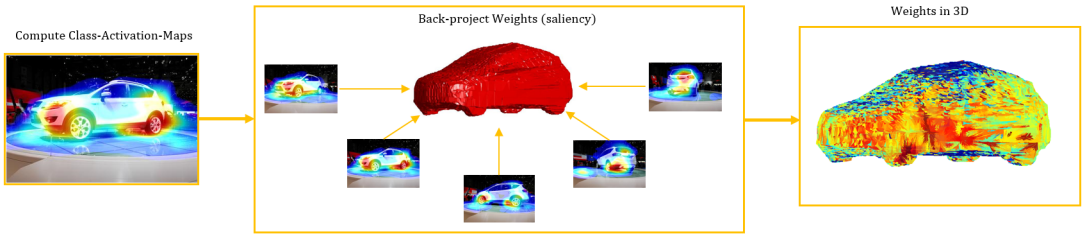

Figure 3.2: A heatmap of the most salient regions in the image are computed (left) using *Class-Activation-Mapping* [35]. Backprojecting and aggregating the saliency maps from different views (center) results in a 3D estimate of saliency (right).

ing a variety of cues, *e.g.* sparse part-based sampling [15, 23, 36], voting-based approaches [14] *etc.* Continuous optimization like [22, 23], which use image edges (or other features) for fitting, are interesting to us, as they allow for more fine-grained and effective optimization of shape and pose.

In recent times, the rapid increase in the availability of training data and the upsurge of sophisticated deep learning methods have led researchers to tackle this problem in a more data-driven manner. While most approaches simply harness the power of regression, there has been a few attempts to train deep neural networks end-to-end, using geometry-aware loss functions [17, 8].

In that regard, Choy *et al.,* [5] proposed a novel Recurrent Neural Network (with LSTM) based architecture for both single and multi-view reconstruction tasks, that learns mapping from 2D images to the underlying 3D shapes, using a large collection of synthetic data. At test time, given one or more images, the framework outputs the reconstruction in the form of 3D occupancy grid. Similarly,

Gwak *et. al.,* [17] proposed a Generative Adversarial Network (GAN) architecture, where a 3D model generator that uses image masks and ray trace pooling to generate 3D shapes, alongside a discriminator which is trained using synthetic 3D shapes, to obtain smoother & realistic 3D reconstructions at test time. Both aforementioned approaches are able reconstruct object from single views, but the quality of reconstruction is comparable only as the number of views increases. On the other hand, Kurenkov *et al.,* [20] proposed DeformNET, an end-to-end single view reconstruction system that extends Spatial Transformer Networks [18] to learn geometric transformations in 3D, coupled with the use of Point Set Generation Network [8], where [8] demonstrated the use of point cloud in contrast to voxel or mesh representation, to be superior in terms of computation with ability to capture natural invariance for a single view object reconstruction problem. DeformNet architecture entails a Free-Form Deformation (FFD) layer that deforms the shape (represented as point clouds) using to achieve smooth geometric deformations.

Finally, the linear subspace that models deformation in most state of the art is still overwhelmingly learned from 3D CAD datasets, which are tedious to build and biased by artists. This approach emphasizes learning from real, arbitrary 2D image sequences, easy to collect, making generalization across classes more plausible. Progress in Structure from Motion and Multi-view Stereo based reconstruction [12] makes this increasingly realizable. Kumar *et al.,* [3] used Structure From Motion reconstructions generated from 2D image sequences (instead of 3D CAD models), for learning deformable shape representation. We extend [3] to model

object shape using a more sophisticated two-pronged representation instead of a coarse deformable part model, which allows us to render a dense 3D reconstruction at test time, as opposed to a coarse wireframe reconstruction of [3]. Moreover, we discover parts in 3D automatically, instead of relying on manual annotations like [3]. In our approach, we use a combination of Structure From Motion [1] and space carving methods [12] to reconstruct arbitrary class-specific image sequences and register the dense reconstructions to learn a linear subspace.

Additionally, we learn the sparser, semantically meaningful part representation (appearances and their spatial configurations). Our methods discovers the parts are most essential in identifying a class (see [35]); this is easier, scales to more classes and is more principled than manually annotated CAD data.

The above two-pronged representation is very useful; the sparser part model allows one to detect and reconstruct the object effectively and bootstrap optimization, while the denser representation provides a comprehensive reconstruction. Since the parts and meshes deform together, one can reason about occlusion accurately, rather than heuristics and statistics.

To summarize, the major contributions of this paper are,

- A two-pronged model for sparse (part-based) and dense (comprehensive) representation of 3D meshes; for jointly solving class-specific object detection, continuous pose estimation and deformable dense 3D reconstruction of the test object instance.

- We use sequences of images taken around objects to learn the 3D shape representation (both dense and sparse), using SfM reconstructions as opposed to using tediously acquired CAD models.

- Instead of relying on data driven regression models for binning poses, we reason the image based evidence in a *PnP*-like scheme that is cognizant of surface occlusion, to estimate pose that is physically faithful.

- We represent the *sparse* part model using a linear shape subspace, where parts (3D) are discovered automatically from images —based on a combination of image saliency and appearances —and the SfM reconstructions (to project them to 3D); replacing the need for human part annotations,

- The final reconstruction is optimized using least squares minimization, in an incremental, stable manner.

3.4 Proposed Approach

We will now detail our approach, our model, how it is learnt and how to apply it to unseen test images.

3.4.1 A two-pronged shape model

The Dense Shape Model: A 3D shape instance of a class is given as $\underset{3\times P}{\mathtt{S}}$, which can be modelled by a linear combination of basis shapes so that each vertex is a weighted sum of the corresponding basis vertices: $\mathbf{s}_p = \sum_l \alpha_l \cdot \mathbf{b}_{lp}$, ($\boldsymbol{\alpha}$ are the shape

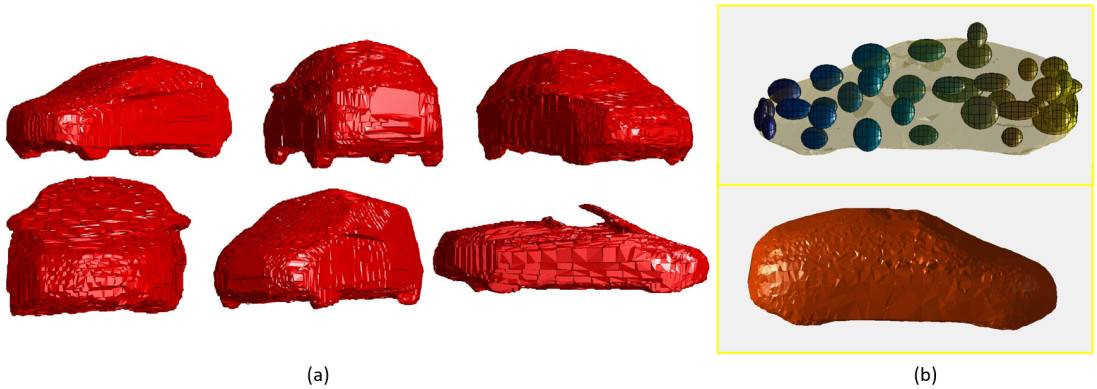

Figure 3.3: *(a)* Examples of 3D meshes obtained using space carving techniques on 2D image sequences. *(b)* Standard deviations of the part positions in 3D plotted using ellipsoids *(top)*. The larger the ellipsoid, the higher the standard deviation is in the corresponding direction. The mean car shape obtained by averaging full 3D meshes *(bottom)*.

coefficients). Note, the same representation applies to the dense shape mesh and the sparse set of parts. When viewed through a camera $\mathtt{C} = \mathtt{K}[\mathtt{R}\,\mathbf{t}]$ at a particular viewpoint, the projected shape vertices are given by:

$$\hat{\mathbf{u}}_p = \pi\left(\mathbf{C} \cdot \mathbf{s}_p\right), \tag{3.1}$$

where $\pi(.)$ represents the perspective projection in a pinhole camera. Here we approximate this with a scaled orthographic camera. $\mathtt{K}$ represents the intrinsic camera matrix, and $\mathtt{R}$ and $\mathbf{t}$ stands for rotation matrix and translation vector.

Figure 3.3 shows examples of dense shape models learned from real images. Using real image sequences for the car class, each from around a unique 3D object class instance, and space carving based SfM [12], we reconstruct rigid shapes for

each sequence, minimizing the image reprojection error. These meshes are normalized to a uniform mesh resolution, scale and registered in orientation using ICP. A linear subspace is then learned using Principal Component Analysis from this data. Also, learning 3D shapes directly from 2D images allows us to go back and forth between 2D and 3D representations easily (projection and back-projection); thus the appearance features (of parts) can be learned from 2D images, whereas their position and deformation can be reasoned about in 3D.

The Sparse Part Model: The sparse part model follows the conventional deformable representation of an object using a set of part positions in 3D. A part can be described as a salient regions of an object, that are repeatably identifiable, and invariant to deformation, viewpoint and illumination *etc*. Thus the proposed automatic 3D part discovery technique is based on finding the most distinctive and informative 2D features needed for identifying the category correctly against background. We use a Class Activation Mapping (CAM) based approach (based on [35], detailed in § 3.5) leveraging the *Global activation pooling* layer. A deep network is trained to perform a binary classification task (in our case *cars vs. not-cars*). Then, for every image fed into the network, the CAM approach outputs a heat map which can be understood as the relative importance of image features for the classification task.

The heat maps for training sequence images are then back-projected to its corresponding 3D instance reconstruction, which leaves us with a large number of 3D points (that correspond to salient image regions in 2D), across multiple 3D shape instances. The goal of our part discovery is to identify regions in 3D that are

salient as well as repeatable (occurs in multiple shapes). For that purpose, these 3D part estimates are then clustered using K-means, where the distance between two parts is a sum of their 3D distance (in the normalized frame of reference) and distance in the appearance space (between feature descriptors extracted from their corresponding 2D image regions). This extracts a sparse set of parts (with consistent appearance and location), in the registered and normalized frame of reference. For each part a Gaussian 3D location distribution and a mixture model for part image appearance is learned (see § 3.5).

3.4.2 Pose Estimation

At test time, given a query image, we first convolve the learned part appearance classifiers on the test image, to obtain a set of candidate part detections. In the correct camera pose, the correctly deformed shape instance should project to the image, so that the visible parts match their projections in appearance. This reasoning is performed using a RANSAC-based perspective-n-point [16] like approach described in Algorithm 3. We search randomly sampled part combinations to jointly estimate the best view projection parameters assuming a mean shape, that agrees best with the visual evidence. To reason pose, the RANSAC scheme only uses part subsets that are jointly visible, for accurate 3D-2D fitting (joint visibility statistics can also be learned during training for efficiency). Each fit is evaluated by maximizing the part appearance score, and minimizing the distance between estimated and actual part positions along with a *root filter*.

The root filter as used by [10], is a categorization the pose of the object (as a whole) into fixed-sized viewpoint bins, that acts much like a regularizer (more details in Section 3.5).

Algorithm 3 RANSAC-based Pose Estimation Algorithm

1: A set of candidate parts are obtained by performing part detection on the test image.
2: **for** N iterations **do**
3: A minimal set of parts candidates are chosen randomly. In this case the minimal set needs to be of size 3 (to compute Scaled Orthographic projection). The chosen part candidates must be collectively visible in at least one of the views.
4: Estimate the *pose and deformation parameters* by estimating reprojection loss between the mean shape and the 2D part positions chosen in step 1 (above).
5: Check for inliers (support), the remaining parts (or corresponding part detections in 2D), that satisfies visibility criteria, projects within a threshold τ_1.
6: Store the set and the estimated parameters if the number of inliers are greater than threshold τ_2.
7: Re-estimate the parameters minimizing the projection loss for all inliers (candidate parts), through least-squares fitting, instead of just the minimal set, to obtain the best parameter estimate.

3.4.3 Pose and Shape Refinement

The estimation is further refined by estimating the shape deformation parameters in addition to refining the camera parameters. We perform a least-squares optimization but instead of using just the mean shape, we estimate the shape deformation parameters $\boldsymbol{\alpha}$ in addition to the camera parameters $\mathtt{C}$ (initialized from the previous estimation) allowing the optimization to jointly reason about the test object's shape deformation and pose. For a query image I, the loss function can

be defined as the difference between the projected and observed object parts:

$$L_1(\boldsymbol{\alpha}, \mathtt{C}) = \frac{1}{P}\sum_{p=1}^{P} v(\mathbf{s}_p, \mathtt{C}) \cdot ||\mathbf{u}_p - \underbrace{\pi(\mathbf{C} \cdot \mathbf{s}_p)}_{\hat{\mathbf{u}}_p}||^2 \tag{3.2}$$

where $v(\mathbf{s}_p, \mathtt{C})$ is boolean with a value of 1 if $\mathbf{s}_p$ is visible to $\mathtt{C}$ and 0 for occluded parts. $\hat{\mathbf{u}}_p \in \hat{\mathbf{U}}$ is the 2D projection of 3D part $\mathbf{s}_p$ where $\mathbf{u}_p \in \mathbf{U}$ is corresponding part detection. The shape and pose parameters are estimated by minimizing L with respect to $\boldsymbol{\alpha}$ and $\mathtt{C}$.

The loss function in equation 3.2 can be augmented with terms that score a part in terms of how well the image appearance matches a learned model $p(\hat{\mathbf{u}}_p|\gamma_p)$ and how likely its relative 3D position is with respect to the learned related 3D part distribution $p(\hat{\mathbf{s}}_p|\delta_p)$ similar to [28]. Thus the loss for a part p projected using camera $\mathbf{C}$ and shape (α) parameters is given as:

$$L_2(\boldsymbol{\alpha}, \mathtt{C}) = -\frac{1}{P}\sum_{p} \left(\ln(p(\hat{\mathbf{u}}_p|\gamma_p)) + \ln(p(\hat{\mathbf{s}}_p|\delta_p))\right) \tag{3.3}$$

We additionally employ a regularizer in our optimization based on off the shelf object detectors like [10]. Here, we convolve the learned root filters with the image to get possible image bounding boxes that the parts should lie within. This is converted to a map with low values inside the bounding box detections (and high outside). The loss term minimizing the cost of projected part positions $\hat{\mathbf{u}}_p$

on this map can be evaluated as :

$$L_3(\boldsymbol{\alpha}, \mathtt{C}) = -\frac{\lambda}{P} \sum_p \text{map}^2(\hat{\mathbf{u}}_p) \tag{3.4}$$

Thus the total loss between estimated projection and actual image evidence can be formulated as:

$$L(\boldsymbol{\alpha}, \mathtt{C}) = L_1(\boldsymbol{\alpha}, \mathtt{C}) + L_2(\boldsymbol{\alpha}, \mathtt{C}) + L_3(\boldsymbol{\alpha}, \mathtt{C}) \tag{3.5}$$

In Algorithm 3, steps 4 to 5 minimizes L_1 loss (equation 3.2), where the shape refinement (step 7) minimizes the total loss $L(\boldsymbol{\alpha}, \mathtt{C})$ (equation 3.5). Please refer to Section 3.5 for a detailed explanation of the optimization.

Full Reconstruction: Estimating the deformation and camera parameters allows us to recover the underlying the skeleton of the shape instance in the test image along with its viewpoint. To perform a complete 3D reconstruction, we deform the full 3D meshes corresponding to the basis shapes used for estimating the shape of the object instance, using the deformation weights estimated above.

3.5 Implementation Details

The performance of our approach depends on the implementation of several units, described at a high level above.

CAM based learning: We use the publicly available pre-trained weights of VGGnet 16-layer architecture [32], trained on ImageNet challenge [31]. We modify the network architecture by replacing the fully connected layers (layers after *conv5-3* in VGGnet) with a *global average pooling* layer, which simply computes the spatial average of the feature map (from *conv5-3*) at each unit κ, whose weighted sum outputs the final saliency map. The global average pooling layer is then followed by a softmax layer that outputs the likelihood of the image belonging to the category or not (for two-way/binary classification). We learn weights corresponding to each class for each unit κ. The weighted sum of the feature maps of the *conv5-3* layer is used to compute class activation maps.

Learning the sparse part model: After discovering parts from the clustering process of 3.4.1, the location distribution is learned by fitting a Gaussian distribution to the 3D clusters. Part appearance models are learned as a mixture model similar to [10, 15]; for each part we train individual SVM classifiers for each discrete viewpoint bin in which the part is visible. We discretize the viewing sphere into 12 bins and on an average each part has 3-6 mixture components (classifiers). Part appearance classifiers operate on CNN-based *conv5* layer features (see [13]) extracted from the images. To train part appearance classifiers, we fine-tune the top layers of the 16-*layer* VGGnet [32] (pretrained for ImageNet Classification task [31]), to adapt the network to perform a 12-*way* viewpoint classification task (suggested by [6] for domain-specificity).

We categorized the Epfl Multi-view Cars dataset [25], into 12 bins (of 30° each). Since our goal is to fine-tune the fully connected layer weights and the

last convolutional layer weights, we freeze the weights of the first three sets of convolutional layers *(conv1 - conv3)* for faster learning. We replace the last (1000-way softmax classification) layer with a 12-way softmax layer. We then train the modified network for viewpoint classification task, so that the weights (*conv* layer) are fine-tuned to adapt to our dataset, and also the network learns to discriminate appearances cues (of parts/regions) with respect to viewpoint.

Supplementary model: In addition to the sparse part and dense shape model, we learn a supplementary view-specific root filter model that learns the holistic object appearance conditioned on the viewpoint [10, 26]. The root filter based view proposals are an additional cue in RANSAC based estimation process and help filter out the incorrect poses estimated due to object symmetry. The root filter is learned by binning each image into a certain viewpoint interval and then we extract *fc7 (fully connected layer)* features from the images (using the fine-tuned CNN model to train sets of SVM classifiers, one per viewpoint bin.

3.6 Evaluation

Most available 3D object datasets such as Pascal3D [34] or ShapeNet [4], use synthetic CAD models for 3D representation of objects along with manually annotated pose, thus not feasible for demonstrating our work. In order for us to learn 3D shape representation from images, we need datasets of image sequences taken around the object. We demonstrate the performance of our proposed framework on EPFL-Multiview [25] Cars dataset, one of the most commonly used datasets

for viewpoint estimation problems. EPFL-Multiview Cars dataset [25] contains 20 sequences of images taken around car instances, We randomly split the dataset into two and use 10 sequences for training and 10 for testing. Below are a few evaluation metrics on which we evaluate the performance of our proposed framework.

3.6.1 Part Detection

Part detections are obtained by convolving learned part appearance filters (SVM classifiers), on *(conv5)* features extracted from the test image pyramid. We combine part detections across all scales, and use non-maxima suppression to remove redundant bounding boxes. Then each part detection score is approximated to a probability using Platt scaling [29]. To evaluate our part detection performance, we use the standard evaluation metric of [7]. A part detection is considered valid if there is a 50% overlap between the groundtruth and the detected bounding box. Also we use Mean Average Precision (mAP) to evaluate the performance of our part detection. The mAP of our part detection system on Epfl-cars dataset [25] is 45.97%.

3.6.2 Estimation of Camera Parameters

In order to evaluate the viewpoint estimation performance of our system, we compute Mean Precision in Pose Estimation (MPPE) [24] and Median Angular Error (MAE) [14]. MPPE is a measure of viewpoint classification accuracy where we discretize azimuths into k number of bins and compute the classification accuracy

θ	EPFL-Multiview Cars [25]				
	(Ours)	3D^{2}PM-D [26]*	3D^{2}PM-C Lin [26]*	[24]	[25]
$\pi/4$	99.4 / **88.74**	99.4 / 78.5	97.8 / 78.3	91.0 / 73.7	-
$\pi/6$	99.4 / **83.07**	97.9 / 75.5	98.3 / 76.2	-	-
$\pi/8$	99.4 / **76.85**	99.0 / 69.8	97.5 / 69.0	97.0 / 66.0	85.0 / 41.6
$\pi/9$	99.4 / **72.94**	99.2 / 71.8	99.3 / 71.2	-	-
$\pi/18$	99.4 / 46.21	99.3 / 45.8	99.2 / **52.1**	-	-

Table 3.1: Viewpoint Classification Accuracy using MPPE [24] on EPFL Multi-view Cars dataset [25]. (* - 3D^{2}PM [26] uses synthetic images generated from 3D CAD models for better appearance training in addition to real images. It is unfair to directly compare the performance between the two, as we rely only on available real image data for training).

θ	EPFL-Multiview Cars [25]				
	(Ours)	3D^{2}PM-D [26]	3D^{2}PM-C [26]	3D^{2}PM-D [26]*	[14]
$\pi/4$	12.84	13.1	13.7	12.9	24.8
$\pi/6$	9.04	-	-	9.0	-
$\pi/8$	7.14	-	-	7.2	-
$\pi/9$	6.70	7.4	7.0	6.2	-
$\pi/18$	4.68	6.4	5.6	5.2	-

Table 3.2: Continous/Fine-Grained Viewpoint Estimation error using MAE [14] on EPFL Multi-view Cars dataset [25].(* - 3D^{2}PM-D [26] uses synthetic images generated from 3D CAD models for better training).

θ	*top-2*	*top-3*	*top-5*
$\pi/4$	91.14	92.07	92.71
$\pi/6$	87.22	88.32	89.46
$\pi/8$	78.35	79.61	81.58
$\pi/9$	74.42	75.84	77.98
$\pi/18$	48.81	51.83	55.61

Table 3.3: *Top-N* accuracy for Viewpoint Estimation (MPPE [24]) using our Ransac-based viewpoint estimation technique, on EPFL Cars dataset [25].

θ	(ours)	[3]
$\pi/4$	0.4713	0.5492
$\pi/6$	0.3680	0.3931
$\pi/8$	0.2791	0.3011
$\pi/9$	0.2280	0.2628
$\pi/18$	0.1272	0.1404

Table 3.4: Continous/Fine-Grained Viewpoint Estimation using MAE (Error/distance between quaternions, where a distance of 3.14 = 180°) on EPFL Cars dataset [25] using all *3 Euler angles*.

using precision for the different number of bins. On the other hand, MAE is for fine/continuous viewpoint estimation, where we compute the median of the angular error between the estimated and groundtruth viewpoints (camera parameters). Table 3.1 and Table 3.2 shows the MPPE and MAE respectively, obtained using our approach on Epfl-cars dataset [25] and compares our approach with Pepik et. al [26], Ozuysal et. al. [25] and Lopez-Sastre et. al. [24]. Table 3.1 shows that our approach consistently outperforms the current state-of-the-art [26] in this dataset, especially the Discrete version of $3D^2PM$, despite [26] uses synthetic images in addition to real images, to train appearance models.

In addition, we also report our *Top-N* accuracy using MPPE for the Epfl-cars dataset [25] in Table 3.3. A classification is considered correct if one of the *top-N* estimates are correctly classified. This metric accounts for how consistent the system is in predicting, and when it fails how badly it fails. The system is consistent if most of *top-N* detections are accurate. In addition, since the solution space is highly noisy, even if the most confident detection is inaccurate, computing top-N

Figure 3.4: Qualitative results. *Left:* Test images overlaid with the projection (2D convex hull) of the estimated shape and viewpoint. *Right:* Full 3D reconstruction of the test object instance obtained by deforming basis shapes (meshes) using estimated shape parameters rotated to the estimated pose.

accuracy provides insight into how close our other estimates are. Analyzing the results further, we identify that one of the most important factors that affects the viewpoint estimation accuracy is the misclassification due to the appearance symmetry or the ill-posed nature of the problem. Though most deformable part based models are engineered to address the ill-posed nature of the viewpoint estimation problem, estimating viewpoint with high precision is quite challenging. We conducted an experiment to see what percentage of mis-classifications fall in the viewpoint range (of 30°) in the opposite side of the actual viewpoint. We found that $\{41.61, 25.91, 16.92, 13.87, 8.71\}$ percent of the total misclassification (MPPE) for bin sizes $\{\pi/4, \pi/6, \pi/8, \pi/9, and\, \pi/18\}$ respectively, are due to the appearance similarity. This experiment also shows that a considerable amount of our error is due to the ill-posed nature of the problem and not due to other noise.

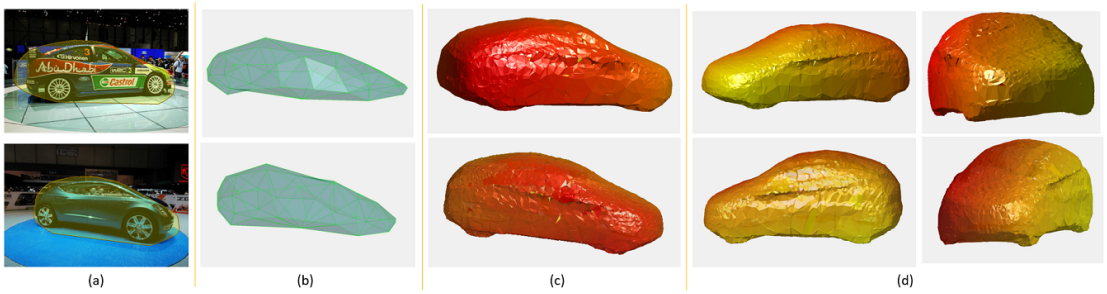

Figure 3.5: Qualitative results. *(a)* Test images overlaid with the projection (2D convex hull) of the estimated shape and viewpoint. *(b)* Wire-frame reconstruction (of estimated part positions) *(c)* Full 3D reconstruction (mesh) of the test object instance rotated *wrt.* estimated camera parameters*(d)* The full 3D reconstruction rotated by different angles to demonstrate the details of the estimated shape better.

3.6.3 Shape Estimation

We evaluate the full 3D reconstruction performance of our framework quantitatively by computing pixel-level accuracy of the projected 2D silhouette of the reconstruction (using the estimated shape and camera parameters) with respect to the groundtruth segmentation. The full 3D reconstruction using the estimated camera and shape parameters projects into a 2D silhouette. Ideally, if we are able to accurately estimate shape and camera parameters, the reconstructed shape/mesh would project compactly into the object's (groundtruth) silhouette, resulting in a high overlap (close to 100%). We use 220 manually segmented images from the test set of Epfl-cars dataset [25] as the groundtruth to evaluate the full 3D reconstruction performance. Our 3D reconstruction framework has a *precision* (pixel-level) of 81.39% and a *recall* of 88.82%.

Figures 3.4 & 3.5 show qualitative results of the full 3D reconstructions on Epfl-cars dataset. Our framework does a pretty good job in reasoning the unknown shape of the test object instance based on the visual evidence. As shown, it is able to deform the shape estimates to render an accurate 3D reconstruction of the test object instance.

3.7 Conclusion

As promised, we have demonstrated an end-to-end pipeline that learns a two-pronged class model automatically from arbitrary class image sequence data. The dense shape representation allows for realistic deformable reconstruction and occlusion modelling, while the sparse model discovers a class part model based on class characteristics, which help in efficient and reliable view estimation and object detection and bootstrapping. We achieve qualitatively and quantitatively pleasing results, thanks to modelling the problem using continuous variables, realistic physics and an incremental and (largely) continuous optimization that learns completely from natural, 2D data (without tedious content creation, annotation and minimizing human bias). Going forward, we would like to test the power of this method for unorganized data from more classes and possibly leverage image sequences for improving reconstructions via temporal reasoning. The proposed two-pronged representation lend themselves readily to real world applications such as self-driving cars, mixed reality or motion capture based gaming *etc.*, that rely

on modelling objects in 3D, for more accurate scene understanding, that in turn, aids decision making.

Bibliography

[1] Autodesk 123d catch. `https://en.wikipedia.org/wiki/Autodesk_123D`.

[2] Y. Bao, M. chandraker, Y. Lin, and S. Savarese. Dense object reconstruction using semantic priors. In *Proc. CVPR*, 2013.

[3] A. C. S. Kumar, A. Bodis-Szomuru, S. Bhandarkar, and M. Prasad. Class-specific object pose estimation and reconstruction using 3d part geometry. In *Proceedings of the ECCV Workshops*, pages xx–yy, Oct 2016.

[4] A. Chang, F. Thomas, G. Leonidas, H. Pat, H. Qixing, L. Zimo, and S. Silvio. Shapenet: An information-rich 3d model repository. In *arXiv preprint arXiv:1512.03012*, 2015.

[5] C. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. ECCV*, 2016.

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proc. ICML*, 2014.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.

[8] H. Fan, H. Su, and L. Guibas. A point set generation network for 3d object reconstruction from a single image. 2016.

[9] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005.

[10] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE PAMI*, 32:1627–1645, 2010.

[11] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, volume 2, pages 264–271, Jun 2003.

[12] A. W. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3D model construction for turn-table sequences. In R. Koch and L. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments, LNCS 1506*, pages 155–170. Springer-Verlag, Jun 1998.

[13] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *Proc. CVPR*, 2015.

[14] D. Glasner, A. Galun, M., R. S., Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *Proceedings of the ICCV Workshops*, 2011.

[15] M. Hejrati and D. Ramanan. Analysis by synthesis: 3d object recognition by object reconstruction. In *Proc. CVPR*, 2014.

[16] J. Hesch and S. Roumeliotis. A direct least-squares (dls) method for pnp. In *Proc. ICCV*, pages 383–390, 2011.

[17] G. J., C. Choy, A. Garg, M. Chandraker, and S. Savarese. Weakly supervised generative adversarial networks for 3d reconstruction. 2017.

[18] M. Jaderberg, K. Simonyan, and A. Zisserman. Spatial transformer networks. In *NIPS*, 2015.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *NIPS*, pages 1106–1114. 2012.

[20] A. Kurenkov, J. Ji, A. Garg, V. Mehta, J. Gwak, C. Choy, and S. Savarese. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. 2017.

[21] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77:259–289, 2008.

[22] M. Leotta and J. Mundy. Vehicle surveillance with a generic, adaptive, 3d vehicle model. *IEEE PAMI*, 33(7):1457–1469, 06 2011.

[23] Q. Lin, Chen, and S. Yan. Automatic 3d model construction for turn-table sequences. In *Proc. ICLR*, 2014.

[24] R. J. Lpez-Sastre, T. Tuytelaars, and S. Savarese. Deformable part models revisited: A performance evaluation for object category pose estimation. In *Proceedings of the ICCV Workshops*, 2011.

[25] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *Proc. CVPR*, 2009.

[26] B. Pepik, P. Gehler, M. Stark, and B. Schiele. $3d^2pm$ - 3d deformable part models. In *Proc. ECCV*, 2012.

[27] B. Pepik, M. Stark, P. Gehler, T. Ritschel, and B. Schiele. 3d object class detection in the wild. In *CVPR Workshops*. IEEE, 2015.

[28] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *Proc. CVPR*, 2012.

[29] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods.

[30] M. Prasad, A. Zisserman, and A. W. Fitzgibbon. Single view reconstruction of curved surfaces. In *Proc. CVPR*, volume 2, pages 1345–1354, June 2006.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. In *IJCV*, 2015.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint*, 2014.

[33] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, and L. Van Gool. Shape-from-recognition: Recognition enables meta-data transfer. *CVIU*, 113(12):1222–1234, 2009.

[34] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Proc. WACV*, 2014.

[35] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proc. CVPR*, 2016.

[36] M. Z. Zia, M. Stark, and K. Schindler. Towards scene understanding with detailed 3d object representations. *IJCV*, 112(2):188–203, 2015.

Chapter 4

DepthNet: A Recurrent Neural Network Architecture for Monocular Depth Prediction

Arun CS Kumar[1], Suchendra M. Bhandarkar, Mukta Prasad

[1]First Author. In the Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR) 2018 workshops, Reprinted here with permission of publisher, April, 2018.

4.1 Abstract

Predicting the depth map of a scene is often a vital component of monocular SLAM pipelines. Depth prediction is fundamentally ill-posed due to the inherent ambiguity in the scene formation process. In recent times, convolutional neural networks (CNNs) that exploit scene geometric constraints have been explored extensively for supervised single-view depth prediction and semi-supervised 2-view depth prediction. In this paper we explore whether recurrent neural networks (RNNs) can learn spatio-temporally accurate monocular depth prediction from video sequences, even without explicit definition of the inter-frame geometric consistency or pose supervision. To this end, we propose a novel convolutional LSTM (ConvLSTM)-based network architecture for depth prediction from a monocular video sequence. In the proposed ConvLSTM network architecture, we harness the ability of long short-term memory (LSTM)-based RNNs to reason sequentially and predict the depth map for an image frame as a function of the appearances of scene objects in the image frame as well as image frames in its temporal neighborhood. In addition, the proposed ConvLSTM network is also shown to be able to make depth predictions for future or unseen image frame(s). We demonstrate the depth prediction performance of the proposed ConvLSTM network on the KITTI dataset and show that it gives results that are superior in terms of accuracy to those obtained via depth-supervised and self-supervised methods and comparable to those generated by state-of-the-art pose-supervised methods.

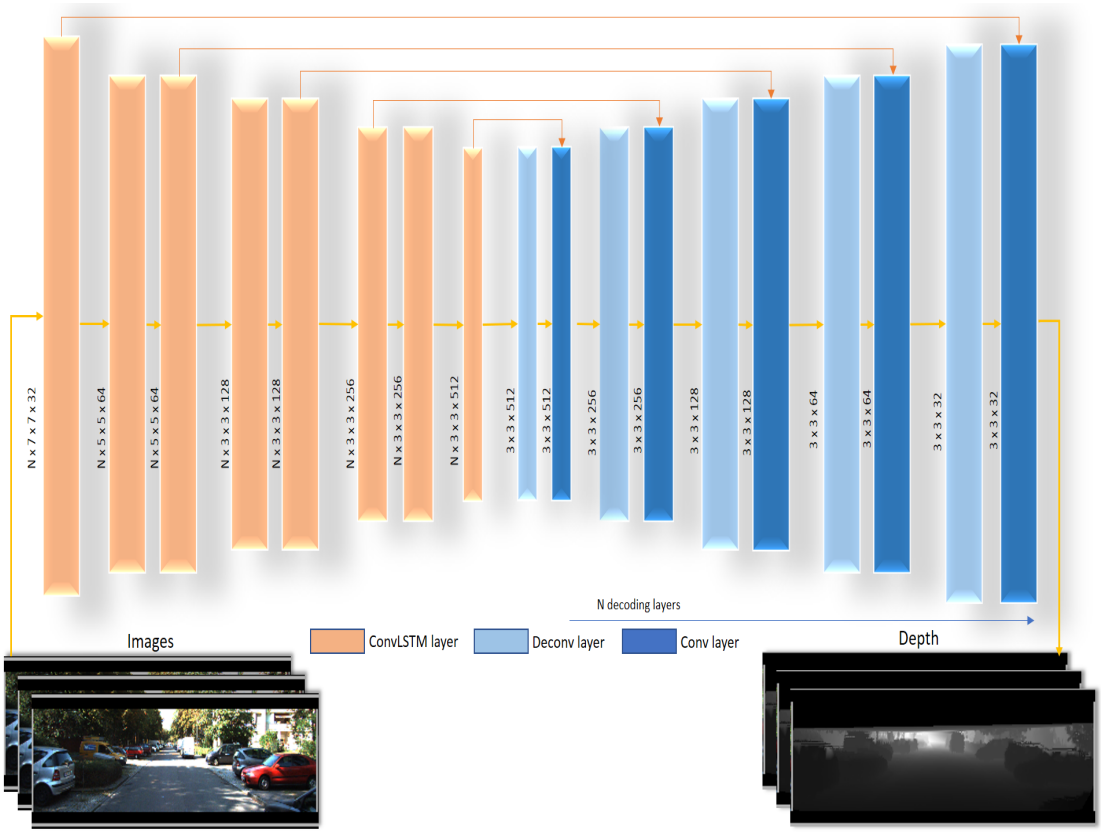

Figure 4.1: Proposed network architecture: The encoding layer consisting of multiple ConvLSTM [31] layers (orange blocks) takes a single image or image sequence as input at test time. The decoding layer consisting of an alternating sequence of deconvolutional and convolutional layers (*blue blocks*) reconstructs the depth maps.

4.2 Introduction

Scene reconstruction is one of the fundamental problems in computer vision research. In recent times, learning-based approaches to depth estimation have been explored and exploited widely for 3D scene reconstruction in a wide range of applications including simultaneous localization and mapping (SLAM) for self-driving cars and virtual reality (VR)-based and motion capture (MOCAP)-based gaming, to cite a few. For a given input image, infinite depth maps can be conjured up and determining the correct one is very difficult. However, by understanding the underlying scene semantics and employing suitable priors one can narrow down the possibilities to obtain realistic depth maps in a reasonable time frame. For example, for a continuous video of a slowly changing scene, the corresponding depth map also exhibits low temporal variation. Consequently, temporal smoothness is an important prior that is exploited in almost all current SLAM techniques.

Significant recent progress has been made in single- and multi-view 3D scene reconstruction, deriving 3D scene structure from motion (SfM) and simultaneous localization and mapping (SLAM) [9, 22]. However, accurate monocular depth prediction through deep learning is considered the ultimate test of the efficacy of modern learning- and prediction-based 3D scene reconstruction techniques. The ready availability of RGBD sensors (such as Kinect and LiDAR) in recent times has made acquiring pairs of images with accompanying depth maps considerably easier, at least for the purpose of learning, even if too expensive for perpetual deployment. Pre-calibrated stereo rigs also provide an effective substitute for RGBD sensors, but require reasoning about scene disparity. Monocular prediction using pairs of

frames is the toughest, as one needs to reason about the relative camera pose as well as disparity/flow and there is an inherent ambiguity in scale, unless we resort to a consistent SLAM like reconstruction pipeline.

Learning to predict depth, even if only approximately, provides an opportunity to inject valuable information into the 3D reconstruction, 3D pose estimation and inference procedures in SLAM. Learning complex semantic scene relationships by capturing the spatio-temporal relationships between image entities (such as regions and textures) across different imaging modalities (such as RGB images and depth maps) calls for the formulation of complex learning models accompanied by large datasets. In recent times, aided by the rapid progress in deep learning methods and availability of large datasets [11], learning-based techniques for some of the sub-problems in SLAM has become viable, *e.g.*, [6, 19, 12, 32, 28]. Methods in predictive reconstruction also need to account for uncertainty and noise in the video sequence data, especially when distinct objects in a scene have motion parameters that are independent of the global camera motion parameters (*e.g.* multiple cars moving in different directions on a road). Additionally, unlike depth maps which can be readily obtained using depth sensors, getting accurate ground truth pose data for objects moving independently in a scene is much more difficult.

In this paper, we propose a scheme for learning object pose implicitly from sequences of image and depth map pairs for training. We demonstrate that we are able to effectively learn and predict depth as a function of image appearance over time using an LSTM-based deep learning model [27] The proposed model is shown to capture inter-frame dependencies and variations, without explicit model-

ing of the object pose. Given a sequence of images, the proposed spatio-temporal approach predicts the depth map using both the current image frame and its predecessors. We also demonstrate the performance of the proposed approach on predicting the depth maps for future or unseen image frames given the current image frame and/or previous image frames by harnessing the sequential reasoning capability of the LSTM.

While use of depth as supervision or priors based on reasoning about forward-backward image consistency have been substantially explored, temporal smoothness as a prior is relatively unexploited when using a video sequence as input. In this paper, we propose a novel convolutional LSTM-based recurrent neural network architecture that learns depth as a function of appearance while implicitly learning the object pose and its smooth temporal variation. The goal of the paper is to demonstrate that the use of temporal information is particularly effective in estimating depth. The proposed convolutional LSTM-based recurrent neural network architecture is depicted in Figure 4.1.

4.3 Related Work

Traditional *structure-from-motion* (SFM) and SLAM techniques jointly estimate structure and motion parameters, either using point correspondences [9, 22, 29, 30] or direct methods [7, 23]. In recent times, CNN-based approaches [6, 18, 12, 28, 17] have been seen to achieve good performance on well constrained subsets of the general monocular reconstruction problem, *e.g.* predicting depth and pose

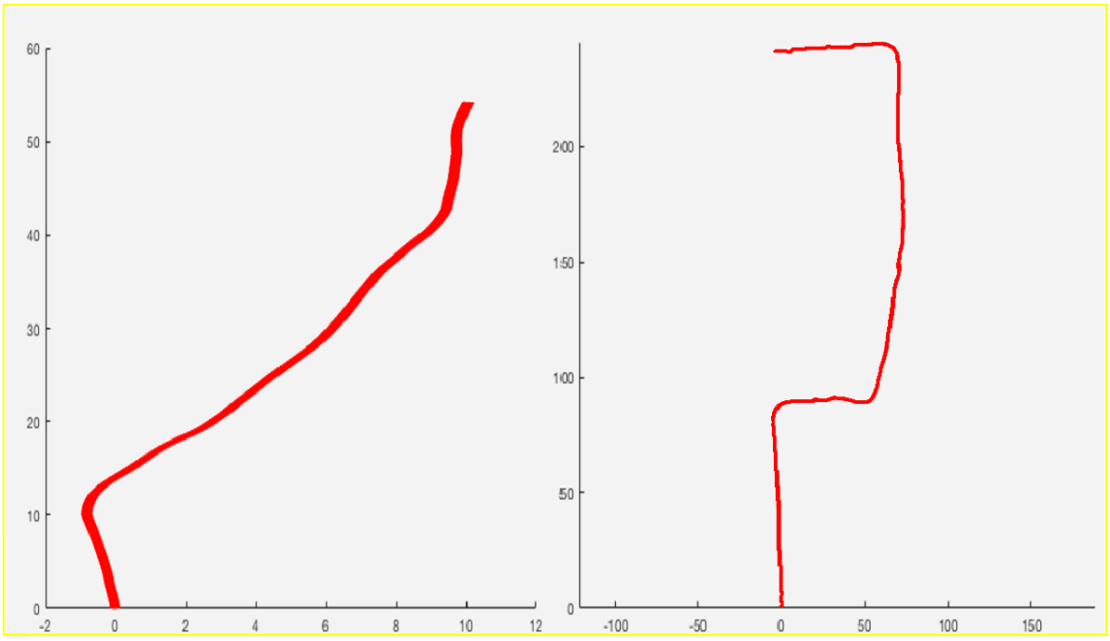

Figure 4.2: Quality of pose predictions using Learning-based models: The 3D coordinates obtained using camera matrices are plotted for an example image sequence from the KITTI dataset [11]. The plot is constructed by simply aggregating the camera motions, obtained using deep learning technique [32] (left) versus the ground-truth (right). Evidently, the deep learning-based camera motion estimate is far from matching the form of the ground truth; since the scale for each relative pose prediction is unknown the uncertainty adds up quite rapidly. Therefore, though the predicted pose parameters model relative motion reasonably well, they clearly lack the ability to model global pose efficiently.

given a single or 2-3 consecutive images in space and/or time. The ability of CNN-based approaches to capture complex relationships between the depth maps and the corresponding image textures along with other scene semantics has been demonstrated in [6, 12, 28]. These CNN-based approaches are trained either in a depth supervised manner [6, 16] from a single view to predict corresponding depth maps, or in a pose/self-supervised mode employing photo-consistency with input stereo images [12, 17] or consecutive images in time and their capabilities are often demonstrated on single-view depth prediction [28, 32]. The *FlowNet* architecture proposed by Dosovitskiy *et al.* [5] for optical-flow estimation is based on an encoder-decoder CNN where channel-concatenated image pairs are provided as input to the network that learns optical flow in a supervised fashion. A CNN variant termed as *FlowNetCorr*, merges two different convolutional networks from pairs of adjacent images, by correlating the tensors to learn the disparity map, mimicking the traditional point correspondence-based optical flow methods.

Since the formulation of *FlowNet* and *FlowNetCorr* architectures, several approaches have proposed encoder-decoder CNN architectures for computing disparity maps, which are subsequently used for depth prediction [12, 21, 28]. Additionally, the architecture proposed by Godard *et al.* [12] learns to minimize the left-right consistency between adjacent image pairs to improve the pose estimation accuracy in a pose-supervised setting. The stereo problem formulated in [12] assumes a known pose, making it equivalent to one of estimating depth through disparity. On the other hand, Zhou *et al.* [32] propose a joint pose and depth prediction technique that learns reconstruction up to scale in an unsupervised setting

using video frames as input. In addition, explainabilty masks are used to isolate individual motions of objects that do not agree with the predicted motion parameters of the scene [32]. Vijayanarasimhan *et al.* [28] extend the work of Zhou *et al.* [32] to model the individual motions of the objects isolated using the explainability maps. In their more recent work, DeTone *et al.* [4] propose a CNN-based approach that identifies isolated and evenly distributed feature points from image pairs, which are then fed to another neural network that learns to compute the homography between them.

While the use of object pose information has been shown to improve depth prediction accuracy considerably [12], object pose predicted using optical flow-based approaches on real-world images [28], is far from accurate, and in fact often falls short in comparison to traditional point correspondence-based pose estimation approaches. Figure 4.2 displays visual plots of deep learning-based pose estimations whose pose estimation accuracy, as shown, is highly inaccurate. Eigen *et al.* [6] propose a simple, but effective scheme for monocular depth prediction from image appearance, albeit with considerable supervision. Their network consists of two components, the first component is a traditional convolutional network that captures coarse global scene structure, followed by the refinement of the estimated coarse depth map using the image color/texture information. In addition, they also propose a scale-invariant error measure to address the global scale ambiguity. In this paper, we extend the scheme proposed by Eigen *et al.* [6], to predict depth from an image explicitly, while modelling pose information implicitly via temporal reasoning using LSTMs.

4.3.1 Recurrent Neural Networks for Temporal Learning

Recurrent neural networks (RNNs) [14] are a class of neural networks that models the temporal behavior of sequential data using hidden states with cyclic connections. In feed-forward convolutional networks the gradient is backpropagated through the network; an RNN additionally backpropagates through time (across multiple instances of the network) that enables them to learn dependencies across time. The long short-term memory (LSTM) [14] is an extension of the traditional RNN, that is capable of learning long-term dependencies within an input sequence.

Recently several approaches have used the LSTM for learning temporal dependencies across image frames in a video sequence [8, 20, 26]. Srivastava *et al.* [26] learn a video representation using an encoder-decoder framework which is then used for future frame prediction. Similarly, Lotter *et al.* [20] propose a predictive neural network that is inspired by predictive coding and trained in an unsupervised fashion for the purpose of video prediction. Choy *et al.* [3] present a scheme for learning the mapping between images and the corresponding 3D object shapes using a gated recurrent unit GRU [2], a variant of LSTM with fewer parameters. At test time Choy *et al.* [3] reconstruct a 3D occupancy grid for the underlying scene using one or more images.

In our paper, we employ a convolutional LSTM (ConvLSTM) [31], to model the spatio-temporal dependencies between video frames for the purpose of predicting depth maps. The use of the ConvLSTM instead of the traditional fully-connected LSTM allows us to jointly exploit the ability of the multiple convolutional layers to capture appearance cues at multiple spatial scales along with the ability of the

LSTM to reason about the temporal variations in the input data, without losing any spatial information.

The major contributions of our paper can be summarized as follows:

- We adapt the convolutional LSTM (ConvLSTM)-based encoder-decoder architecture for scene depth prediction from monocular video sequences. Given temporally adjacent image frames and their corresponding coarse ground truth depth maps, the proposed ConvLSTM network has the opportunity to learn a spatio-temporal mapping between the image and depth data. At test time, the network can predict depth maps from both, image sequences and single images.

- We demonstrate the ability of the network to reason sequentially, by extrapolating the current depth maps for the future (or unseen) image frames, without explicitly training it to do so.

- We present new results for monocular depth prediction, on the KITTI dataset [11], that outperforms other depth(only) [6, 19], pose/stereo-supervised [12, 10] and other self-supervised [32] methods, and are comparable to some state-of-the-art [17] that use depth+pose/stereo.

4.4 Proposed Approach

We propose using a convolutional LSTM (ConvLSTM)-based network architecture for depth prediction from a monocular video sequence. In contrast to traditional

depth prediction models that process a single input image, the proposed ConvLSTM network learns depth maps from a set of N consecutive video frames in a depth-supervised setting, allowing the ConvLSTM network to perform spatio-temporal reasoning about the image-depth map relationship. In addition, unlike the traditional LSTM-based approaches [3], where a *fully-connected* LSTM layer is introduced between *encoder* and *decoder* networks, we stack a set of ConvLSTM layers [31] on top of each other to construct the encoding phase. The ConvLSTM is a variant of the traditional *fully connected* LSTM that has convolutional structures underneath. The traditional LSTM layer unfolds the input tensor into a vector, thus does not take into consideration the spatial correlations between the grid cells [31]. The advantage of stacking multiple ConvLSTM layers is that, the multiple LSTM layers allows the network to better learn the temporal information whereas the underlying convolutional structure helps retain the spatial relationships between the grid cells. Moreover, the use of the ConvLSTM also reduces the total number of trainable network parameters significantly [31]. This is in contrast to the fully connected LSTM layers which unfold to generate a densely connected vector with much spatial data redundancy. In the proposed network, the ConvLSTM layers can be shown to effectively capture the spatio-temporal information with much higher accuracy than the traditional LSTM layers.

In the proposed network, the encoder layer is comprised of ConvLSTM layers, each layer holding N states where N is the total number of timestamps. The decoder layer reconstructs the depth maps learned for each of the states separately. The decoder layer follows an architecture similar to that of the U-Net [24], with

N separate deconvolutional layers and skip connections between the encoder and decoder layers. This decoder architecture, which has been shown to work well for several reconstruction tasks [32], also allows for more accurate reconstruction of the depth map. In the proposed network, the encoder layer learns the spatio-temporal relationships between N image frames for the purpose of predicting the depth maps whereas the decoder layer learns to reconstruct the N individual depth maps.

4.5 Network Architecture

The proposed network architecture illustrated in Figure 4.1, consists of an encoding (contraction) phase followed by a decoding (expansion) phase. The encoding (contraction) phase takes as input a set of N consecutive video frames and computes an intermediate depth representation towards the end. The decoding (expansion) phase reconstructs the depth maps from the intermediate depth representation. The encoding (contraction) phase comprises of a stack of K ConvLSTM layers that takes as input an image sequence across N time points to learn the depth representation as a function of time. As mentioned above, in comparison with a *Fully-connected* LSTMs (FC-LSTM) [27], ConvLSTM [31] has fewer connections, with shared weights, that make them easier to learn than the dense connections of the traditional FC-LSTMs. In addition FC-LSTM as opposed to ConvLSTM, does not take spatial correlation into consideration [31]. Thus the ConvLSTM well suited for learning the underlying representation from spatio-temporal data.

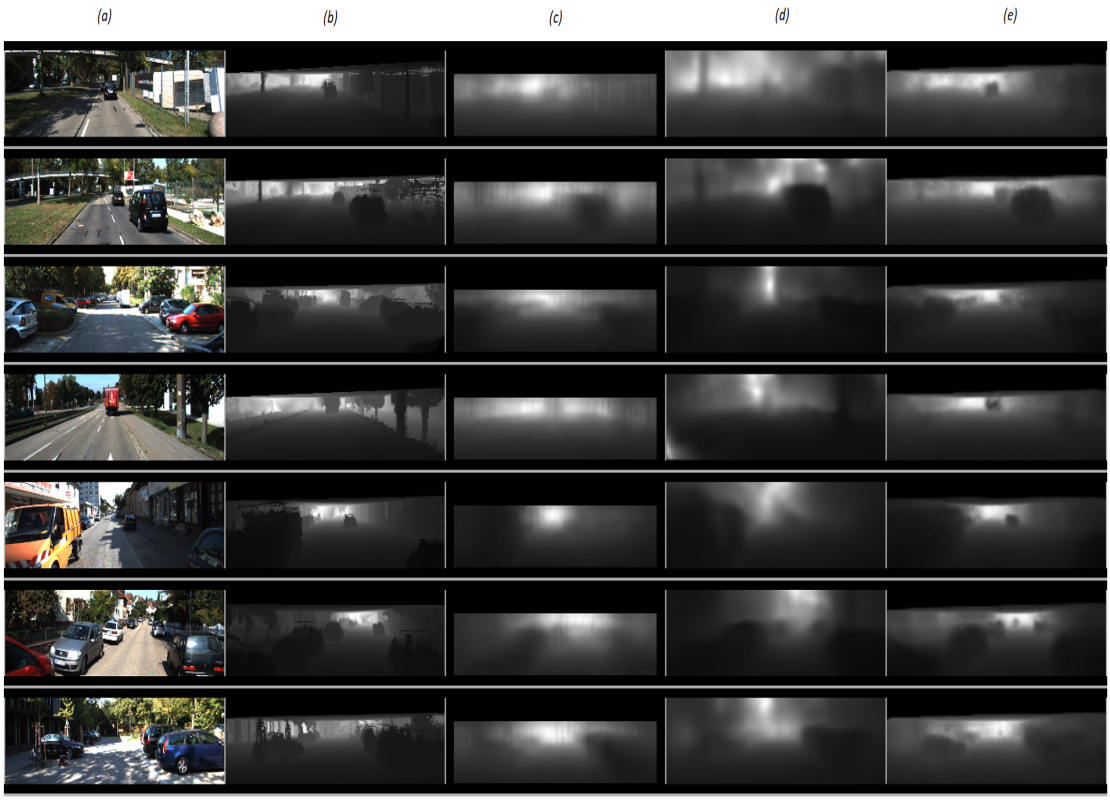

Figure 4.3: Qualitative results (*good*). (a) Image (t) (b) Corresponding ground truth depth map (c) Depth predictions from Eigen *et al.* [6] (depth-supervised) (d) Depth predictions from Zhou *et al.* [32] (e) Depth predictions from the proposed scheme (*note*: the proposed scheme uses $N-1$ preceding images in addition to the test image (a), for predicting the depth map)

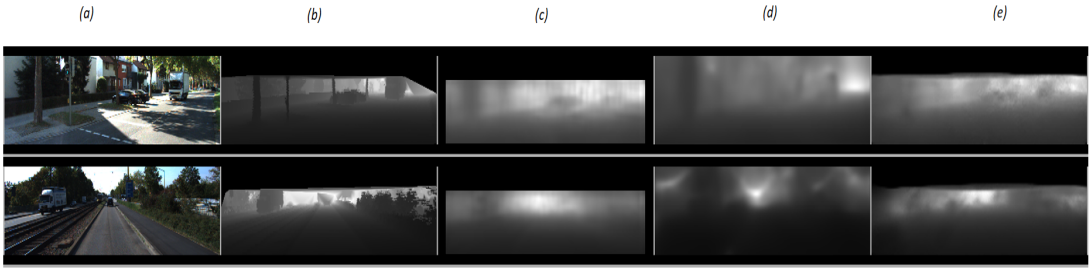

Figure 4.4: Qualitative results (*bad*). (a) Image (t) (b) Corresponding ground truth depth map (c) Depth predictions from Eigen *et al.* [6] (depth-supervised) (d) Depth predictions from Zhou *et al.* [32] (e) Depth predictions from the proposed scheme (*note*: the proposed scheme uses $N-1$ preceding images in addition to the test image (a), for predicting the depth map)

A more comprehensive description of the network architecture and details pertaining to its training are provided in Section 4.6.

Unlike traditional ConvNets where the network learns and predicts from a single view and the error is propagated from the bottom layer of the decoder to the top layer of encoder, in our case the error propagates temporally thereby capturing the time-dependent progression of the scene depth. While the contraction (encoding) phase learns to encode depth as a spatio-temporal function of the image sequences, the expanding (decoding) phase learns to reconstruct the depth map from the intermediate representation. We use deconvolutional layers with skip connections for the purpose of depth reconstruction. The architecture for the decoding phase is detailed in Section 4.6. Towards the end of the expansion phase, we use a 1×1 convolutional layer with sigmoid activation to obtain the depth map. For training, we use a scale-invariant error metric proposed by Eigen

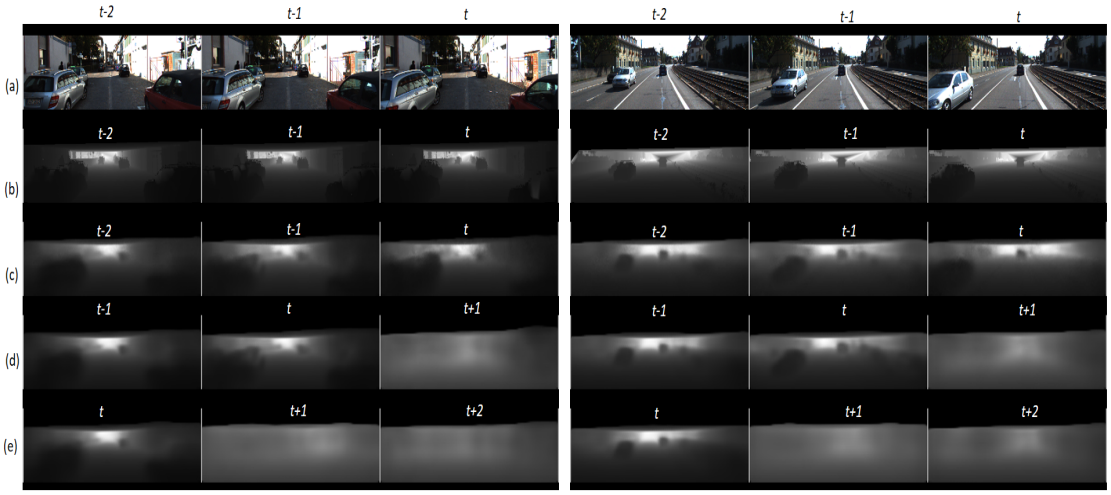

Figure 4.5: Qualitative demonstration of depth prediction results for the current image frame and future image frames obtained by unfolding the LSTM layers (a) Images at times t_{n-2}, t_{n-1}, t_n, (b) The corresponding ground truth depth maps (c) Depth predictions of the proposed ConvLSTM network for image frames at time steps t_{n-2}, t_{n-1}, t_n (which is the way the network is actually trained to predict) (d) Predictions of the proposed ConvLSTM network, for future frames at time t_{n-1}, t_n, t_{n+1} (e) predictions of the proposed ConvLSTM network for future frames at time t_n, t_{n+1}, t_{n+2}. For (d) & (e), it must be noted that, the proposed ConvLSTM network is not trained to predict future frames; instead we mask the inputs for specific time steps and force the network to predict the frames, thereby exploiting its recurrent nature. Qualitative analysis of the results over several images showed that the proposed network was able to reliably estimate the layout of the scene, but failed to interpolate accurately the motion of the scene objects into the future.

Table 4.1: Comparison of monocular depth prediction results on KITTI dataset [11].

θ	Supervision			Error Metric				Accuracy Metric		
	Depth	Pose	Unsupervised	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al. [6]* (Coarse)	✓			0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen *et al. [6]* (Fine)	✓			0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al. [19]*	✓			0.202	1.614	6.523	0.275	0.678	0.895	0.965
(**Ours**–*image sequence*)	✓			**0.137**	**1.019**	**5.187**	**0.218**	**0.809**	**0.928**	**0.971**
(**Ours**–*single image*)	✓			0.176	1.3711	5.971	0.265	0.740	0.896	0.959
(**Ours** t_{n+1} *frame*)	✓			0.296	3.251	9.849	0.469	0.535	0.749	0.855
(**Ours**–*CNN*)	✓			0.145	1.062	5.424	0.273	0.754	0.904	0.969
Godard *et al.* [12]		✓		0.148	1.344	5.927	0.247	0.803	0.922	0.964
Garg *et al.* [10] (50m cap)		✓		0.169	1.080	5.104	0.273	0.740	0.904	0.962
Zhou *et al.* [32] (w/ exp. mask)			✓	0.221	2.226	7.527	0.294	0.676	0.885	0.954
Zhou *et al.* [32]			✓	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou *et al.* [32] (50m cap)			✓	0.208	1.551	5.452	0.273	0.695	0.900	0.964
Kuznietsov et al. [17]	✓	✓(stereo)		**0.113**	**0.741**	**4.621**	**0.189**	**0.875**	**0.964**	**0.988**
Kuznietsov et al. [17]		✓(stereo)		0.308	9.367	8.700	0.367	0.752	0.904	0.952

et al. [6] as the loss function. Given a predicted depth map y_i and its ground truth depth map y_i^*, the loss function [6] is given by:

$$L(y, y^*) = \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}\left(\sum_i d_i\right)^2 \tag{4.1}$$

where $d_i = log(y_i) - log(y_i^*)$ for the i^{th} pixel and n corresponds to the total number of pixels, with the value of λ set to 0.5 [6].

4.6 Implementation Details

Encoding Phase: The encoding phase consists of a series of 3 × 3 ConvLSTM layers consisting of {32, 64, 64, 128, 128, 256, 256, 512} filters respectively, with

alternating strides of 2's and 1's, except for the first two layers that use filters of size 7×7 and 5×5 respectively. While the *relu* activation function is used in each convolutional step of the ConvLSTM layer, the recurrent step uses the *hard sigmoid* activation function. The padding is set to be the same for all layers. The first ConvLSTM layer takes an input I of size $\{B \times N \times H \times W \times C\}$, where H and W are the height and width of the image respectively, C represents the number of image channels (C is 3 as we use standard RGB images), N is the total number of time steps and B is the batch size. In our experiments we set the value of N to 3. Each ConvLSTM layer is designed to return the entire sequence (comprising of all states) instead of just the final state, so that it can be used in the decoding phase.

Decoding Phase: The decoding layer consists of alternating sequence of deconvolutional and convolutional layers. The deconvolutional or transposed convolutional layer takes as input the output sequence the last ConvLSTM layer and performs a deconvolution operation on it. We also use skip connections across the encoding and decoding phases, by concatenating the deconvolved tensor with the output of the corresponding original convolutional layer. Since concatenating the tensors doubles the number of channels, we affix the concatenation layer with an additional convolutional layer to reduce the tensor size. In recent times, the use of skip connections has shown to work well, especially when dealing with the vanishing gradient problem [13] thereby effectively allowing the exploration of deeper network architectures. In a manner similar to the encoding phase, we use a series of deconvolutional layers of sizes $\{512, 256, 256, 128, 128, 64, 64, 32\}$ respectively

(which are then followed by a concatenation and convolutional layer for the skip connections) with alternating strides of 2's and 1's. The filter sizes are set to 3×3 for all deconvolutional layers. Towards the end, we use a 1×1 convolutional layer with a *sigmoid* activation function that converts the tensor to a depth map.

Training: The network is trained with inputs as temporally concatenated image sequences of size N, with the corresponding ground truth depth map as output. We use *batch normalization* [15] layers after each pair of convolution or deconvolutional layers, and train the network using the Adam optimizer [16], with a learning rate of 1×10^{-4} and the loss function described in Section 4.5. In most cases, the validation loss is observed to converge within 20 epochs. Details regarding the dataset split are provided in Section 4.7.

4.7 Evaluation

We train and evaluate our model on the KITTI dataset [11]. The KITTI dataset consists of video sequences of outdoor scenes along with their corresponding depth maps, procured using car-mounted cameras and Velodyne LiDAR sensors. We use the train/test split described in [6], where we train on 28 sequences and test on the 697 images provided in [6]. Throughout our experiments, the number of time steps for training the ConvLSTM is set to 3. We evaluate our approach by using the standard metrics proposed by [6].

4.7.1 Depth Prediction from Monocular Sequences and Single Images

Although the proposed network is trained using fixed-length image sequences, at test time we evaluate its performance on both monocular image sequences and single images. In addition, we also evaluate, qualitatively and quantitatively, the accuracy of the extrapolated depth maps corresponding to future (or) unseen image frame(s).

Monocular Image Sequences: For predicting depth on image sequences, we gather image sequences of of size N —of the 697 images provided by [6], 23 images are the first of their respective sequences, which we had to omit for this experiment as they have no preceding images. We tabulated the results of our approach against other state-of-the-art in Table 4.1. The proposed approach is observed to outperform depth-supervised approaches while yielding results comparable to those of pose-supervised techniques [10, 12]. The qualitative depth prediction results are shown in Figures 4.3 and 4.4, where Figure 4.4 shows instances where the proposed approach fails to predict the scene depth reasonably. In order to demonstrate the improvement due to the use of the ConvLSTM component over the CNN baseline, we trained a CNN with a similar architecture and reported the results in Table 4.1. Our results outperforms most (reported) state-of-the-art methods that are depth-supervised [6, 19], pose/stereo-supervised [12, 10] supervised and other self-supervised approaches [32]. While our numbers are inferior to [17] (Table 4.1, *row* 13), it must also be noted that [17], uses both depth and stereo as supervi-

sion, along with more sophisticated and deeper network architecture (ResNet [13]) with pre-trained weights, as opposed to our less sophisticated network (in terms of depth) that does not rely on pre-trained weights [25].

Single Images: Although the network is trained using image sequences, the decoding layer is designed to individually reconstruct each state of the encoding phase. Doing so allows us to use a single image at test time, and still get reliable depth reconstruction, although the network is trained using image sequences only. Using a single image at test time would mean that only the first recurrent layer will receive input (others will get empty placeholders), in which case the network will act like an end-to-end ConvNet or CNN instead of a recurrent network. The quantitative results for depth prediction using single images are presented in Table 4.1. The results are comparable to the predictions obtained using monocular sequences and are even better than those of most other approaches.

Future Depth Prediction: As an attempt to exploit the ability of the LSTM to reason temporally, we analyze quantitatively its ability to predict depth maps of future frames. The goal of the experiment is to see how well the network is able to learn inter-frame dependencies. For that purpose, in a manner similar to our previous experiment, we replace images in the image sequence with empty placeholders, and force the network to predict depth maps. The quantitative results are presented in Table 4.1. Though the prediction results are not comparable, the future prediction results show, both qualitatively and quantitatively, how the information propagates over time, and how well the network is able to learn inter-frame dependencies. The qualitative results for future depth prediction are depicted in

Figure 4.5. The results suggest that the future frame predictions, though not quite accurate especially when modeling individual objects, are still able to estimate the layout of the scene reasonably well. Also, it has to be noted that we do not train the network explicitly for predicting future image frames, instead we simply force the network to predict by masking the input(s).

4.8 Conclusion

In this paper we explored whether recurrent neural networks (RNNs) can learn spatio-temporally accurate monocular depth prediction from video sequences, even without explicit definition of the inter-frame geometric consistency or pose supervision. To this end, we proposed a novel convolutional LSTM (ConvLSTM)-based network architecture for depth prediction from a monocular video sequence. In the proposed ConvLSTM network architecture, we harnessed the ability of long short-term memory (LSTM)-based RNNs to reason sequentially and predict the depth map for an image frame as a function of the appearances of scene objects in the image frame as well as image frames in its temporal neighborhood. We demonstrated quantitatively and qualitatively that the proposed ConvLSTM is able to perform better at depth prediction than traditional CNN models, by obtaining convincing state-of-the-art results on the KITTI dataset compared to current depth-supervised approaches. Although our network is trained to make depth predictions for image sequences, it can predict depth maps, at test time, on single images as well with high accuracy. Also, we have demonstrated the network's abil-

ity to reason temporally, by extrapolating depth maps for future/unseen frames, without the network being explicitly trained to do so. In the future, we plan to automatically learn explainability masks, that would model individually each independently moving object in the scene. The explainability masks could then be used for predicting the depth map for each individual object, in the current image frame and in future image frames, more accurately.

Bibliography

[1] Bailey, T., & Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3), 108-117.

[2] Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[3] Choy, C. B., Xu, D., Gwak, J., Chen, K., & Savarese, S. (2016, October). 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. European Conference on Computer Vision* (pp. 628-644). Springer International Publishing.

[4] DeTone, D., Malisiewicz, T., & Rabinovich, A. (2017). Toward Geometric Deep SLAM. arXiv preprint arXiv:1707.07410.

[5] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., & Brox, T. (2015) Flownet: Learning optical

flow with convolutional networks. In *Proc. IEEE International Conference on Computer Vision*, (pp. 2758-2766).

[6] Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems* (pp. 2366-2374).

[7] Engel, J., Schps, T., & Cremers, D. (2014, September). LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. European Conference on Computer Vision* (pp. 834-849). Springer.

[8] Finn, C., Goodfellow, I., & Levine, S. (2016). Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems* (pp. 64-72).

[9] Furukawa, Y., Curless, B., Seitz, S. M., & Szeliski, R. (2010, June). Towards internet-scale multi-view stereo. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1434-1441).

[10] Garg, R., Carneiro, G., & Reid, I. (2016, October). Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proc. European Conference on Computer Vision* (pp. 740-756). Springer International Publishing.

[11] Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3354-3361).

[12] Godard, C., Mac Aodha, O., & Brostow, G. J. (2016). Unsupervised monocular depth estimation with left-right consistency. arXiv preprint arXiv:1609.03677.

[13] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

[14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, Vol. 9(8) (pp. 1735-1780).

[15] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning* (pp. 448-456).

[16] Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[17] Kuznietsov, Y., Stckler, J., & Leibe, B. (2017, February). Semi-supervised deep learning for monocular depth map prediction. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6647-6655).

[18] Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., & Navab, N. (2016, October). Deeper depth prediction with fully convolutional residual networks. In IEEE 3D Vision (3DV), 2016 Fourth International Conference on (pp. 239-248).

[19] Liu, F., Shen, C., Lin, G., & Reid, I. (2016). Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38(10) (pp. 2024-2039).

[20] Lotter, W., Kreiman, G., & Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint arXiv:1605.08104.

[21] Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., & Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4040-4048).

[22] Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, Vol. 31(5) (pp. 1147-1163).

[23] Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011, November). DTAM: Dense tracking and mapping in real-time. In *Proc. IEEE International Conference on Computer Vision* (pp. 2320-2327).

[24] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 234-241). Springer, Cham.

[25] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252.

[26] Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015, June). Unsupervised learning of video representations using LSTMs. In *Proc. International Conference on Machine Learning* (pp. 843-852).

[27] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

[28] Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., & Fragkiadaki, K. (2017). SfM-Net: Learning of Structure and Motion from Video. arXiv preprint arXiv:1704.07804.

[29] Williams, B., Klein, G., & Reid, I. (2007, October). Real-time SLAM relocalisation. In *Proc. IEEE International Conference on Computer Vision* (pp. 1-8).

[30] Wu, C. (2011). VisualSFM: A visual structure from motion system. http://ccwu.me/vsfm/

[31] Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems* (pp. 802-810).

[32] Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. arXiv preprint arXiv:1704.07813.

Chapter 5

Monocular Depth Prediction using Generative Adversarial Networks

Arun CS Kumar[1], Suchendra M. Bhandarkar, Mukta Prasad

[1]First Author. In the Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR) 2018 workshops, Reprinted here with permission of publisher, April, 2018.

5.1 Abstract

We present a technique for monocular reconstruction, *i.e.* depth map and pose prediction from input monocular video sequences, using adversarial learning. We extend current geometry-aware neural network architectures that learn from photo consistency-based reconstruction loss functions defined over spatially and temporally adjacent images by leveraging recent advances in adversarial learning. We propose a generative adversarial network (GAN) that can learn improved reconstruction models, with flexible loss functions that are less susceptible to adversarial examples, using generic semi-supervised or unsupervised datasets. The generator function in the proposed GAN learns to synthesize neighbouring images to predict a depth map and relative object pose, while the discriminator function learns the distribution of monocular images to correctly classify the authenticity of the synthesized images. A typical photoconsistency-based reconstruction loss function is used to assist the generator function to train well and compete against the discriminator function. We demonstrate the performance of our method on the KITTI dataset in both, depth-supervised and unsupervised settings. The depth prediction results of the proposed GAN are shown to compare favorably with state-of-the-art techniques for monocular reconstruction.

5.2 Introduction

As computer vision matures, it is increasingly clear that in addition to recognition and classification, reconstruction and pose estimation are imperative to do well,

ideally performed jointly. This goal when achieved, would have wide ranging implications especially in areas *e.g.* robotic navigation and simultaneous localization and mapping (SLAM). However, the problem continues to be a difficult one to solve, the ability to relate information across multiple views, while handling noise, uncertainty and estimating pose and shape (encapsulated by Structure from Motion (SfM) and SLAM) continues to be actively researched and improved, despite much progress. While the geometry of image formation, image features, and hand crafted energies and priors have been well studied, there is significant curiosity and hope for what deep learning progress can bring to the table, especially in terms of discovering features, formulations, exploiting priors and large amounts of data. Naturally, in recent times, various sub-tasks of reconstruction *e.g.* single view reconstruction, optical flow (and scene flow!), pose estimation and joint pose and depth estimation for temporal and stereo setups have seen a flurry of activity using supervised and semi-supervised learning.

In this paper, we advance this state of the art, by harnessing the power of adversarial learning over the existing state of the art in geometry aware neural network based monocular reconstruction. Generative Adversarial Networks (GANs) have shown themselves to be a promising tool; rather than purely loss based learning regulated by some prior, a GAN pits a generative neural network (generating samples of a variable of interest) against a discriminative one (that tests its authenticity). This allows the discriminator to learn more flexible distributions from available data than typical manually defined loss functions, and are shown to tackle underfitting-based issues, work well even without supervised train-

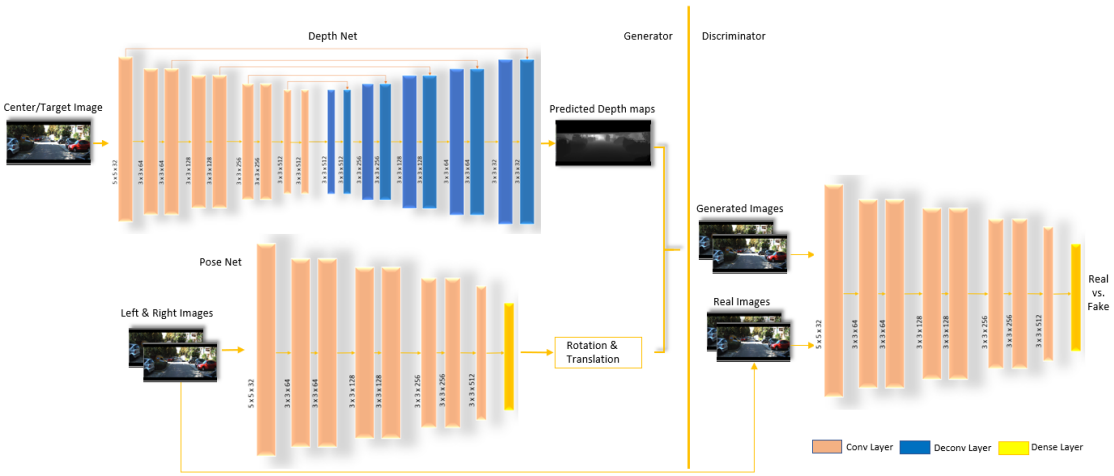

Figure 5.1: Proposed Framework: The *generator* consists of two subnetworks - the *depth* subnetwork that predicts depth map from the target *(center)* image, and the *pose* subnetwork, that learns to predict pose parameters from image triplets. The image triplets are fed to the *pose* subnetwork, that transforms the source *(left & right)* images to the target *(center)* image, while the *center* image is fed to the *depth* subnetwork that outputs a depth map. Using the estimated depth and pose parameters, the generator transforms the source images, which is then interpolated using Spatial Transformer Networks [17], to output a generated pair of images. The *discriminator* subnetwork then learns to differentiate the real and the generated images. Please refer to section 5.5 for more information on network architectures.

ing pairs and tackle confusing, adversarial cases better. We address the following question in monocular reconstruction: given at least 2 contiguous images in time from a monocular camera, can we predict depth and pose (using the generator) such that the predicted images in neighbouring time steps are realistic enough to pass the discriminator's test? The answer, it turns out, is yes. Further, traditional photo-consistency losses help train our system better. Our method, shown in Figure 5.1, compares favourably against state of the art on the latest benchmarks for reconstruction based evaluations and our results are not only geometrically and photometrically consistent but also better trained against adversarial examples. We now introduce this in the context of related work.

5.3 Related Work

Reconstruction traditionally adopted approaches motivated by the physics of image formation from 3D either based on finding consistent 2D matches that yield good reconstruction (triangulation) [25, 12, 20] or proposing 3D structure and texture that is consistent with image evidence (photo-consistency based methods, or direct methods) [24, 21, 10]. Whether machine learning can learn to reconstruct without depending on hand-crafted features and energies, has been discussion for a long time. Though regression based approaches towards subsets of the problem [27, 22] showed promise, substantial headway was made more recently by deep learning based approaches. The deep learning progress, seen initially in image classification [18], was followed closely by reconstruction based learning. In

addition to being an intricate problem, deep learned reconstruction depends on exploiting large amounts of data (often with supervision), which was lacking in reconstruction based benchmarks for a while. A popular workaround was the use of artificial datasets *e.g.* with objects superimposed on artificial backgrounds [7, 19]. But then projects like Kitti [14], CityScapes [3] *etc.* paved the way for larger, more comprehensive benchmarks.

Initial attempts were made on, arguably, sub-parts of the problem like correspondence estimation [26, 30], optical flow or disparity [7]. These approaches slowly consolidated such learning into estimation of scene flow (3D point and velocity estimation implicitly giving flow or disparity) [19] for a small set of frames. Another class of approaches aimed at learning to predict depth maps [9, 8, 16] but learnt this mapping from typical supervised pairs of inputs and outputs while Byravan and Fox [2] learned similar prediction of pose from depth maps. However, supervised data for reconstruction is usually limited, LiDaR scanners are expensive, capture limited depth and have a limited view of the viewing sphere. But the community has been quick to extend such learning where a reconstruction based image-consistency error is used to self-supervise the learning from multiple frames, independent of an explicit depth source, either using stereo or a monocular source over time. The basic principle is that if the correct depth and pose for an image pair are predicted, they should be photo-consistent with the source images. So methods [15, 28, 31] moved towards more effectively predicting both depth and pose with less supervision. While some methods [28] utilize available supervision, others [31] aim to be completely self-supervised. Naturally the problem is still

underconstrained and the use of priors helps estimate a sensible solution. Godard *et al*. [15] enforce a prior to encourage the estimated disparity to be smooth. In [28] additional priors on depth maps, motion maps and even the depth gradients are explored.

5.4 Proposed Approach

The proposed method aims to learn depth and pose parameters via adversarial learning. Given a triplet of images that are adjacent frames of a monocular video sequence, we feed the *center or target* image to the *depth* subnetwork that learns to generate a depth map, and the image triplet is fed to the *pose* subnetwork, that regresses pose parameters that transforms the source *(left & right)* images to the target *(center)* image. The estimated depth and pose parameters are then used to render a pair of predicted images for the left and right (using the well known interpolation of [17]). The discriminator, a network that learns to discriminate between original and generated images, scores a likelihood of how similar the original and the generated images are.

Model

Like a traditional GAN, our network consists of two adversarial components; the *generator* that predicts images neighbouring a given image and the discriminator which classifies the authenticity of such generated neighbourhood images. Our generator is comprised of two subnetworks, the *depth* subnetwork and the *pose*

subnetwork. Similar to [31], the input of the depth subnetwork is the image I_t, for which network predicts a depth map $\mathcal{D}_t$. Images I_{t-1} and I_{t+1} are the *left* and *right* (adjacent) images, in time, of I_t. Given the image triplet $\{I_{t-1}, I_t, I_{t+1}\}$ containing, the *pose* subnetwork predicts two pairs *rotation* and *translation* parameters $\{R_{t,t+1}, \mathfrak{t}_{t,t+1}\}, \{R_{t-1,t}, \mathfrak{t}_{t-1,t}\} \in SE3$, representing the relative transform between the camera at successive instants of time, which is then used with the predicted depth map to generate I_t^1 and I_t^2.

In [31], the generator would be trained on the typical photo-consistency loss (augmented with *left-right* consistency constraints as in [15]), given by:

$$\mathcal{L}_{photo} = \frac{1}{2}\big\{|I_t - I_t^1| + |I_t - I_t^2|\big\} \tag{5.1}$$

where I_t^1 and I_t^2 are the predictions of I_{t+1} and I_{t-1}, are generated by transforming images I_{t+1} and I_{t-1} using predicted depth maps and pose parameters.

Then the objective of the discriminator subnetwork is to learn to classify real images from generated ones $\{I_t^1, I_t^2\}$. It has to be noted that the objective of the discriminator is not to learn explicitly the difference between the instances of each generated and real image pair, instead, to learn to provide a likelihood of how real or fake (or generated) a given image is. With the discriminator as D and the the generator as G, following [5] where a generator is modelled by a context encoder, the adversarial loss can be formulated as,

$$\mathcal{L}_{adv} = \max_{D} \quad \mathbb{E}_{x \in \mathcal{X}}[log(D(x)) + log(1 - D(F(x)))] \tag{5.2}$$

where $\mathcal{X}$ represents the data distribution. We adapt the formulation of [5], and use a generative modeling of images, where we model our autoencoder as our generator, $G \triangleq F$. Training context encoders as generators by propagating adversarial loss via discriminator have been shown to be successful [5] on problems such as image inpainting.

We train and evaluate our network in both *depth*-supervised and unsupervised mode. Our training objective for the unsupervised setting that minimizes the photoconsistency error between true and transformed image pairs (Equation 5.1), along with the adversarial loss (Equation 5.2), is given by:

$$\mathcal{L}^{u}_{total} = \lambda_1 \mathcal{L}_{photo} + \lambda_2 \mathcal{L}_{adv} \tag{5.3}$$

For depth-supervised learning, for a pixel i in target image I_t, the depth loss $\mathcal{L}_{depth}$ can be formulated as

$$\mathcal{L}_{depth}(\mathcal{D}_{ti}, \mathcal{G}_{ti}) = \frac{1}{n} \sum_{i} |\mathcal{D}_{ti} - \mathcal{G}_{ti}|^2 \tag{5.4}$$

where $\mathcal{D}_t$ and $\mathcal{G}_t$ are the predicted and ground truth depth map respectively, and n is the total number of pixels. In case of learning with depth supervision, the loss function then becomes:

$$\mathcal{L}^{s}_{total} = \lambda_1 \mathcal{L}_{photo} + \lambda_2 \mathcal{L}_{adv} + \lambda_3 \mathcal{L}_{depth} \tag{5.5}$$

We tried both L_1 and L_2 norms for computing depth loss while training, but we found the results to be pretty similar.

At initial stages of training, the discriminator learns consistently as images generated are quite distorted, overly smoothed and considerably different from real images, as the pose and depth estimates are highly inaccurate. As training progresses, despite the differences between generated and source images (photo-consistency error) are high, the pose and depth estimates are reasonable enough to render images that *appear* realistic. This is because, the optimization requires the pose and depth parameters to be accurate enough to minimize photoconsistency loss, while the discriminator can be fooled by any reasonable estimates of pose and depth as long as they project to the image plane and the interpolation of the image appear realistic. As the generated images start to appear a bit more realistic, the discriminator, then eventually stops learning half way through training, and reaches equilibrium prematurely. As a means to tackle this problem, we first compute difference images $\delta_t^1 = |I_t - I_t^1|$ and $\delta_t^2 = |I_t - I_t^2|$. We then add the difference images δ_{t-1} and δ_{t+1} to the I_t, to obtain $I_t^{\delta_{t-1}}$ and $I_t^{\delta_{t+1}}$ respectively; this step allows us to externally induce discrepancy to make the image appear fake, generating photoconsistency-aware adversarial examples to confuse the discriminator, and let it learn effectively and continually. The discriminator is fed with these error induced instances of generated images $I_t^{\delta_{t-1}}$ and $I_t^{\delta_{t+1}}$ instead of raw generated images I_t^1 and I_t^2. In reference to the discriminator training, this step reduces generated images from appearing realistic despite inaccurate pose/depth maps, by inducing additional error, which will allow the discriminator to continue learning than stop half way through training.

The algorithmic implication of this step is that, the discriminator is in fact trying to minimize the photoconsistency loss as well, as the adversarial loss would remain high as long as the photoconsistency is not minimized. Thus the discriminator is forced to work against the generator at the same time minimize the same objective of that of the generator. Eventually, when the photoconsistency loss is substantially minimized, generated adversarial examples would start to appear more realistic as the noise added would approach to zero. Given the generated and difference images, an adversarial example is computed using the formulation: ${}^{adv}I_t^1 = \omega I_t^1 + (1-\omega) I_t^1 \delta_t^1$ and ${}^{adv}I_t^2 = \omega I_t^2 + (1-\omega) I_t^2 \delta_t^2$.

When estimating pose parameters, independent objects, such as a car passing by, or a pedestrian crossing the road, tends to have its own motion, that, in general, does not agree with the actual camera motion. In order to tackle the problem, we use the explainability masks proposed by [31], to ignore or mask the regions with independent motion out, while estimating the pose of the scene.

5.5 Implementation Details

The **depth subnetwork** follows a traditional encoder-decoder architecture, where the encoding or the contracting phase comprises a series of convolutional layers that transforms an image into a latent representation, which is followed by the decoding or expanding phase that consists of deconvolutional or transpose convolutional layers along with convolutional layers, learns to regress depth maps from the latent representation of encoder. We removed pooling layers and used convolutional

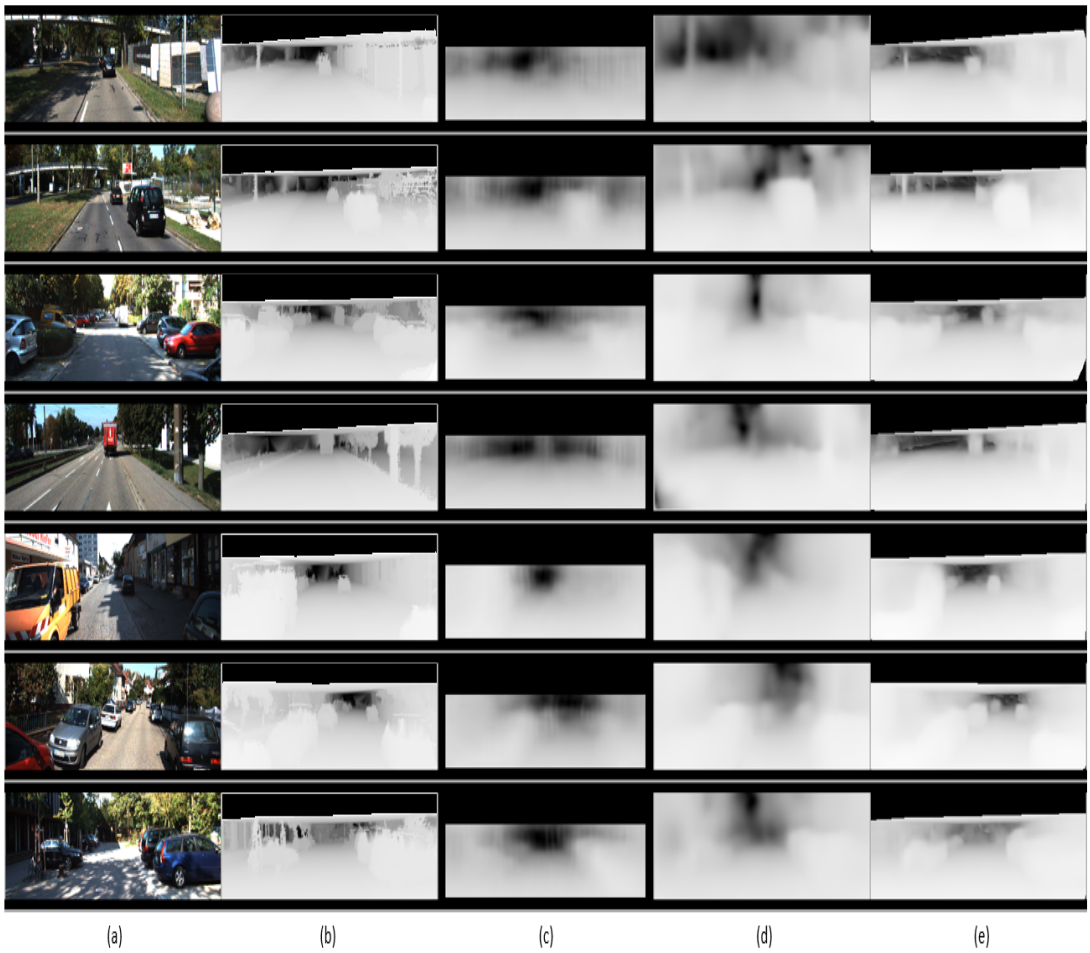

Figure 5.2: Qualitative results *(good)*. *(a)* Ground truth Image *(b)* Corresponding ground truth depth map *(c)* Depth predictions of Eigen *et al* [9] (Depth supervised) *(d)* Depth predictions of [31] *(e)* Our depth predictions (*depth supervision + photoconsistency + adv. loss*).

Figure 5.3: (a) Source images *(left (I_{t-1}) or right (I_{t+1}))* (b) Target image (c) Generated (transformed source) image, using the estimated *pose* and *depth* parameters (d) photoconsistency error between generated *(c)* and *(b)*, (e) Adversarial example with induced photoconsistency loss. The discriminator is trained with images in *(a) and (b)* as real *vs.* images (e) as fake. Image (e) has been enhanced with higher values of ω (=0.6) to amplify the difference for visual demonstration, but during training, the induced photoconsistency loss is kept lower.

Figure 5.4: Qualitative results of depth prediction on KITTI dataset, where the network is trained in an unsupervised manner. *Top* row consists of target images and the *bottom* row consists of the corresponding predicted depth maps.

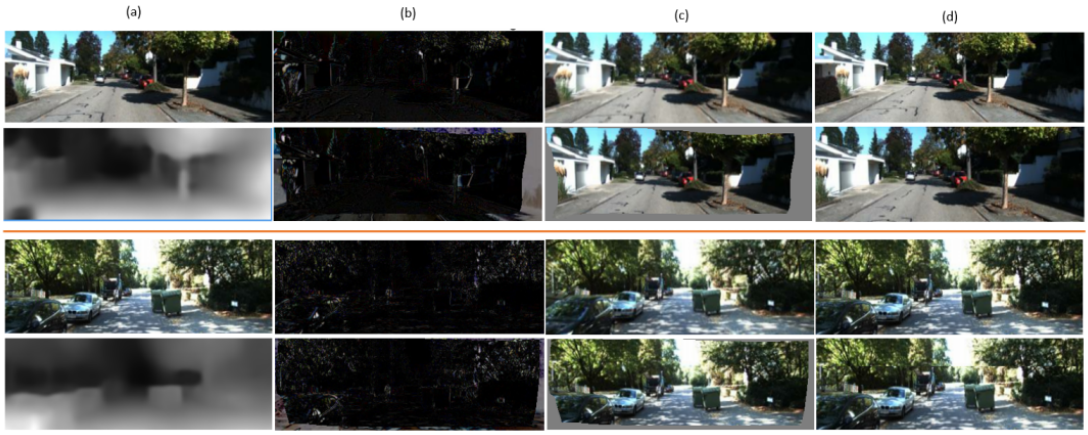

Figure 5.5: (a) Target image and its estimated depth map, (b) left and right photoconsistency error, (c) Projected left and right images (d) groundtruth left and right images.

layers with alternating strides instead, for downsampling the tensor. The pooling layer provides spatial invariance which aids classification, but for autoencoders, removing pooling layers have been shown to improve performance [5]. The encoder consists of 6 pairs of convolutional layers of alternating strides of 2's and 1's, with {32, 64, 128, 256, 512} filters respectively. The *decoder* consists of a series of alternating upconvolutional or transpose convolutional layers and convolutional layers, that maps the latent features of encoding layer into the depth maps. We use skip connections across convolutional and deconvolutional layers as they have been shown improve performance [6], especially for coming across vanishing gradient problems, thus effectively allowing us to explore deeper network architectures. All the convolutional and deconvolutional layers use *relu* as the activation function.

The decoder of the depth subnetwork, is affixed with a convolutional layer with 1 filter and a sigmoid activation, that outputs the final depth map.

The **pose subnetwork**, identical to the encoder of *depth subnetwork*, consists of a series of 8 convolutional layers with alternating strides of 2's and 1's, which is then followed by an average pooling layer, with 12 filters $(6 \times (\eta - 1))$ —3 rotation and 3 translation for two transformations, as we use image triplet ($\eta = 3$). The input of the pose subnetwork is the image triplet that is concatenated across the number of channels. We also use explainability mask to reason motion that do not correspond to the estimated comprehensive camera motion of the scene. We employ the explainability mask algorithm proposed by [31], to mask/ignore the regions with independent motion or occlusion out and use the rest of the regions for inferring pose parameters. We also use resize layers towards the end of the generator, to resize the predicted depth maps to correspond to the actual or required sizes as there are often negligible offsets due to padding. While computing the pose matrix, our method uses intrinsic camera parameters when available, else would default to use $\{0.5, 0.5, 1\}$ for $\{c_x, c_y, f\}$ respectively.

The architecture of the **discriminator** subnetwork is similar to that of a traditional - convolutional layers followed by a set of dense layers (with {512, 256, 128} filters respectively), and a sigmoid layer with a filter size of 1, that simply outputs a probability; all layers use *leaky relu* [1] activation. Like traditional discriminators, the sigmoid layer outputs a single value, the likelihood of the image being real or fake. For all the above subnetworks, except for the first (convolutional)

layer where we use a filter of size 5×5, the filter sizes of all other layers are set to 3×3.

For **Training** the network, we use *Adam* optimizer [4] with an initial learning rate of 0.002 which is periodically adjusted as the training progresses using an exponential decay function, with a rate of decay as 0.95 for every 1500 steps. We gather batches of adjacent images (sequences) in the dataset, as triplets. The *pose* network is fed with the triplets $\{I_{t-1}, I_t, I_{t+1}\}$ of size $B \times H \times W \times (\eta * ch)$, where B is the batch size and ch is the number of channels ($ch = 3$, as we use *rgb* images through out the paper). For all our experiments H and W, the *height* and *width* of images are set to 128 and 384 respectively, and the batch size is set to 32. Also while training, we treat the *left* and *right* stereo pairs of images in KITTI dataset as independent image/video sequences.

The **adversarial examples** are generated as a weighted sum of the generated images and pixel-wise photoconsistency error. We tried various values for weights ω, but we found that ω between 0.90 and 0.95 works better overall. A lesser value for ω implies that the generated adversarial example will be highly noisy, that subsequently leads the adversarial loss to fail to improve as the discriminator accuracy reaches 100 percent hastily, at most times within 3000 iterations. On the other hand, a very high value for ω will introduce too little noise to make a difference in training. Moreover, following [5], we set the hyperparameter (weights) for the adversarial loss λ_2 to be quite low (in comparison to λ_1 or λ_3). For depth supervised learning we set $\lambda_1 + \lambda_3 = 0.995$ and $\lambda_2 = 0.005$, where $\lambda_1 = \lambda_3$, and for unsupervised learning we set $\lambda_1 = 0.995$ and $\lambda_2 = 0.005$.

When training discriminator with adversarial examples, at times, due to the camera motion, regions in the preceding scene, especially corner regions of images, tend to be missing in the current scene causing occlusion. As discussed above, we learn and use the explainability mask to tackle this problem by masking out regions that are occluded. But while training the discriminator, the generated images does not appear realistic because of the mask on the boundaries of the image as shown in Figure 5.5 column (c). Penalizing such images by letting the discriminator classify these as fake, affects the overall pose learning as it restricts the generator by allowing only a negligible motion for camera, which slows down the overall learning rate indeed, or, at times, halts the learning at all. In order to tackle this problem, we apply occlusion masks computed for projected images, to the real images as well, within the minibatch, so that the discriminator learns to classify images as real or fake, irrespective of the presence of the occlusion/explainability masks. Examples of occlusion masks estimated using the explainability mask prediction is shown in Figure 5.5 (c)). In addition, to retain the stability of the network throughout the learning, we train the generator and the discriminator with different learning rates, while the learning rate of discriminator is set to be 20 times lower than the generator, to induce more stability in learning. Also, while training discriminator, we randomly shuffle the source/ground truth and generated images (of the mini-batch) before feeding them to the discriminator, so that the discriminator does not learn to associate the generated and ground truth images as pairs and learns to just differentiate between them, instead learns a more generic objective of classifying real *vs.* fake images.

θ	Supervision			Error Metric				Accuracy Metric		
	Depth	Pose	Unsupervised	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al. [9]* (Coarse)	✓			0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen *et al. [9]* (Fine)	✓			0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al. [11]*	✓			0.202	1.614	6.523	0.275	0.678	0.895	0.965
(**Ours** - *depth+photo.*)	✓			0.1431	0.9741	5.3693	0.2131	0.8001	0.9373	0.9790
(**Ours** - *depth+photo.+adv.*)	✓			**0.1355**	**0.8653**	**5.1736**	**0.2084**	**0.8183**	**0.9450**	**0.9802**
(**Ours** - *depth+photo.+adv.**)	✓			**0.1204**	**0.7466**	**4.7560**	**0.1869**	**0.8486**	**0.9553**	**0.9848**
(**Ours** - *photo.**)			✓	0.2190	1.9758	6.3398	0.2730	0.7081	0.8668	0.9339
(**Ours** - *photo. + adv.**)			✓	0.2114	1.9797	6.1540	0.2636	0.7319	0.8977	0.9593
Godard *et al.* [15]		✓		0.148	1.344	5.927	0.247	0.803	0.922	0.964
Garg *et al.* [13] (50m cap)		✓		0.169	1.080	5.104	0.273	0.740	0.904	0.962
Zhou *et al.* [31](w/o exp. mask)			✓	0.221	2.226	7.527	0.294	0.676	0.885	0.954
Zhou *et al.* [31]			✓	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou *et al.* [31](50m cap)			✓	0.208	1.551	5.452	0.273	0.695	0.900	0.964
Kuznietsov et al. [29]	✓	✓(stereo)		**0.113**	**0.741**	**4.621**	**0.189**	**0.875**	**0.964**	**0.988**
Kuznietsov et al. [29]		✓(stereo)		0.308	9.367	8.700	0.367	0.752	0.904	0.952

Table 5.1: Comparison of Monocular depth prediction results on KITTI dataset [14]. (*-since our depth prediction is not up to scale, we normalized ground truth and estimated depth maps [31]).

θ	Supervision	Error Metric				Accuracy Metric		
	Depth Unsupervised	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou et al.*et al. [31]*	✓	0.267	2.686	7.580	0.334	0.577	0.840	0.937
(Ours-*depth+photo.+adv.* *)	✓	0.3934	4.6831	10.5380	0.4123	0.3518	0.6889	0.9047

Table 5.2: Comparison of Monocular depth prediction results on Cityscapes dataset [3] (* - model trained on KITTI dataset and evaluated on cityscapes dataset (depth capped to 50m), whereas results of [31] are from a model trained explicitly on [3].)

5.6 Experimental Evaluation

For the purpose of evaluation, we train and test our method on the KITTI dataset [14] along with testing on Cityscapes dataset [3], the model that is trained on KITTI dataset. Both KITTI [14] and Cityscapes [3] are similar datasets, comprised of sequences of stereo images, primarily of streets and highways, along with the groundtruth depth maps captured using velodyne laser sensors. With KITTI dataset, we use the train/test split provided by *Eigen et al.* [9], to train the network on 34 image sequences test it on 697 images, whereas for cityscapes we use the default test set provided by the [3]. Also, while training on KITTI, we use the left and right images of the stereo sequences as independent image sequences.

The quantitative evaluation of our algorithm is shown in Table 5.1. We report the performance of our system using the standard metrics used in [9], for both supervised and unsupervised setting. With supervised learning, we obtain *state-of-the-art* results in comparison with other depth-supervised learning techniques. For the task of depth prediction, we obtain a Root Mean Squared Error *(RMSE)* of 4.75, outperforming all other depth-supervised techniques, while comparing equally with *Kuznietsov et al.* [29], who uses the *stereo* information in addition to using ground truth depth maps for supervision. Moreover, *Kuznietsov et al.* [29] uses a more sophisticated and much deeper ResNet [6] with pre-trained weights learned on ImageNet object classification challenge [23], whereas we are able to achieve comparable results by using a much less sophisticated autoencoders with architecture similar to that of [31], without transferring weights learned from other larger datasets. Figure 5.2 shows qualitatively, results of depth maps com-

puted using our approach. It has to be noted that our unsupervised learning models (Table 5.1, *rows 7-8*) were trained only for 75K iterations, which is why the numbers are slightly inferior to Zhou *et al.,* [31]. Our network trained using just photoconsistency loss (Table 5.1, *row 7*) is coarsely our adaptation of [31]; the goal of this paper is to demonstrate that the use of the proposed adversarial scheme improves the overall depth prediction performance of the system in comparison with the baseline (Table 5.1, *row 8*).

Also, for both unsupervised and supervised methods, the estimated depth map is defined up to a scale, so, for the purpose of evaluation, like [31], we normalize them by scaling the predicted depth maps such that their medians match (Table 5.1). In order to demonstrate the generalizable nature of our architecture, we also test our method on the cityscapes [3] dataset, while the training done solely on the KITTI dataset. Despite being trained in a different dataset, our method performs reasonably well in contrast to other methods that are trained on cityscapes (or cityscapes + KITTI datasets). The results are shown in table 5.2. Figure 5.4 shows qualitative results of our method trained unsupervised; we use only the photoconsistency and adversarial losses. We show considerable improvement by training using adversarial loss in contrast to its baseline which is trained using photoconsistency loss alone. We reported results for unsupervised learning at the end of 40K iterations. It has to be noted that the accuracy improves for unsupervised learning with more iterations [31]. Also we observed that when using adversarial loss, it takes slightly longer for convergence than usual, as the

adversarial loss penalizes the pose and depth networks, ceaselessly, throughout the training process.

5.7 Conclusion

We extended a state-of-the-art deep learned depth and pose prediction model and couple it with the adversarial learning to harness the generative power of the GANs, subsequently improving the depth and pose estimation accuracy. We further introduce a technique to generate context-aware adversarial examples, that allows our method to trick the discriminator to work against the generator and at the same time indirectly minimizing the same objective as that of the generator. This proposed method is shown to learn and perform well in both depth-supervised and unsupervised setting, obtaining new state-of-the-art results among depth supervised approaches, and comparing favorably against other pose-supervised and unsupervised techniques. Furthermore, the use of adversarial loss for learning have been successfully demonstrated, both qualitatively and quantitatively, to improve depth and pose prediction accuracy, on two of the important benchmarks.

Bibliography

[1] Xu B, Wang N, Chen T, and Li M. Empirical evaluation of rectified activations in convolutional network. In *arXiv preprint arXiv*, page 1505.00853, 2015.

[2] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. *Proc. Intl. Conf. on Robotics and Automation*, abs/1606.02378, 2017.

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, 2016.

[4] Kingma D and Ba J. Adam: A method for stochastic optimization. In *arXiv preprint arXiv*, page 1412.6980, 2014.

[5] Pathak D, Krahenbuhl P, Donahue J, and Darrell T. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, pages 2536–2544, 2016.

[6] D.He, K Zhang, X Ren S, and Sun J. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016.

[7] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, pages 2758–2766, 2015.

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *Proc. ICCV*, abs/1411.4734, 2015.

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NIPS*, abs/1406.2283, 2014.

[10] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. ECCV*, September 2014.

[11] Liu F, Shen C, Lin G, and Reid I. Learning depth from single monocular images using deep convolutional neural fields. In *Proc. CVPR*, pages 2024–2039, 2016.

[12] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE PAMI*, 32(8):1362–1376, 2010.

[13] R. Garg, B. G. Vijay Kumar, and I. D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *Proc. ECCV*, abs/1603.04992, 2016.

[14] Andreas Geiger. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. CVPR*, pages 3354–3361, Washington, DC, USA, 2012.

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. CVPR*, 2017.

[16] Laina I, Rupprecht C, Belagiannis V, Tombari F, and Navab N. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV)*, pages 239–248, 2016.

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025. 2015.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114. 2012.

[19] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. CVPR*, 2016.

[20] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. 31(5):1147–1163, Oct 2015.

[21] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. ICCV*, ICCV '11, pages 2320–2327, Washington, DC, USA, 2011. IEEE Computer Society.

[22] Bojan Pepik, Peter Gehler, Michael Stark, and Bernt Schiele. 3d2pm – 3d deformable part models. In *Proc. ECCV*, Lecture Notes in Computer Science, pages 356–370, Firenze, Oct 2012.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. In *IJCV*, 2015.

[24] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. CVPR*, pages 1067–1073, 1997.

[25] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *Proc. ACM SIGGRAPH*, pages 835–846, 2006.

[26] James Thewlis, Shuai Zheng, Philip H. S. Torr, and Andrea Vedaldi. Fully-trainable deep matching. *Proc. BMVC.*, abs/1609.03532, 2016.

[27] Alexander Thomas, Vittorio Ferrari, Bastian Leibe, Tinne Tuytelaars, and Luc Van Gool. Shape-from-recognition: Recognition enables meta-data transfer. *CVIU*, (12):1222–1234, 2009.

[28] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *CoRR*, 2017.

[29] Kuznietsov Y, J Stuckler, and B Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proc. CVPR*, pages 6647–6655, 2017.

[30] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *J. Machine Learning Research*, 2016.

[31] Tinghui Zhou, Matthew Brown, Noah Snavely, and David Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. CVPR*, 2017.

Chapter 6

A Deep Learning Paradigm for Detection of Harmful Algal Blooms

Arun CS Kumar[1], Suchendra M. Bhandarkar

[1]First Author. In the Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV) 2017, Reprinted here with permission of publisher, October, 2017.

6.1 Abstract

Effective and cost-efficient monitoring is indispensable for ensuring environmental sustainability. Cyanobacterial Harmful Algal Blooms (CyanoHABs) are a major water quality and public health issue in inland water bodies. The recent popularity of online social media (OSM) platforms coupled with advances in cloud computing and data analytics has given rise to citizen science-based approaches to environmental monitoring. These approaches involve the lay community in the acquisition, collection and transmission of relevant data in the form of tweets, images, voice recordings and videos typically acquired using low-cost mobile devices such as smartphones or tablet computers. While cost effective, citizen science-based approaches are highly susceptible to noise, inaccuracies and missing data. In this paper we address the problem of automated detection of harmful algal blooms (HABs) via analysis of image data of inland water bodies. These image data are acquired using a variety of smartphones and communicated via popular OSM platforms such as Facebook, Twitter and Instagram. To account for the wide variations in imaging parameters and ambient environmental parameters we propose a deep learning approach to image feature extraction and classification for the purpose of HAB detection.

The current system is then extended to perform cyano-vs algal-bloom classification, where in addition to identifying the presence of an algal bloom, we also predict if the bloom is due to cyanobacteria or not. In addition, we have also developed an extension of the above method to leverages the satellite data and the availability data bands that can act as coarse and approximate ground truth data,

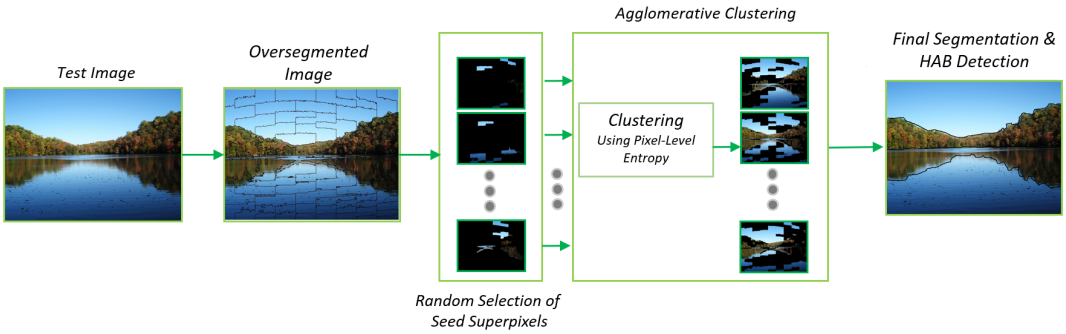

Figure 6.1: Pipeline of the proposed framework for HAB detection

to accurately geo-locate occurrences of cyanobacterial blooms and in addition to act as an early warning system. The proposed method also focuses on using deep learning to learn to interpolate or transfer ground truth bands from one modality (where it is available) of satellite data to another (where the data is not available). We are building a multi-modal fusion framework that combines both the modalities to jointly learn a better representation from the noisy ground truths of both modalities.

6.2 Introduction

Accurate, cost-effective, and targeted monitoring is a critical aspect of sustainable management of the environment. *Cyanobacterial Harmful Algal Blooms* (CyanoHABs) are a major water quality and public health issue in inland water bodies, such as lakes and ponds, since they hamper recreational activities, degrade aquatic habitats and potentially affect human, livestock and wildlife health via toxic contami-

nation [2, 6, 9, 37]. Existing approaches for monitoring of inland water bodies, i.e., *in situ* sampling and satellite-based remote sensing, suffer from serious limitations in terms of cost, data resolution and accuracy. Traditional *in situ* sampling techniques are laborious and time-consuming and prohibitively expensive for real time monitoring of high-frequency environmental phenomena such as water pollution and natural hazard impact assessment. Satellite-based remote sensing techniques can be cost effective but are much less accurate because of mixed pixel issues, geometric noise and radiometric noise. The inherent limitations on the resolution of satellite image data often introduce errors in the resulting prediction models and algorithms. *In situ* remote sensing, using hyperspectral sensors deployed in the field, addresses some of the limitations of satellite image data. Hyperspectral sensors can be useful in rapid, targeted, and cost-effective monitoring of environmental phenomena, especially when these sensors operate, acquire, and transmit data based on an observer-induced triggering mechanism. However, hyperspectral sensors typically have a limited spatial field of view and hence preclude a dense spatial sampling of the underlying environmental phenomena. This may cause the sensors to miss certain environmental phenomena, such as CyanoHABs, in their early stages of development.

With the increasing popularity of online social media (OSM) platforms, such as Facebook, Twitter and Instagram, coupled with advances in cloud computing and data analytics, there has been a growing trend towards adopting citizen science-based approaches to environmental monitoring. The goal is to leverage the OSM platforms to expand the observer base to include not only environmental scientists

and restoration officials but also the lay community at large (i.e., citizen scientists), which includes local area residents and businesses, fishermen and tourists, to encourage more frequent and comprehensive environmental monitoring. The citizen science-based approaches are typically based on acquisition, collection and transmission of data, in the form of tweets, images, voice recordings and videos, using low-cost mobile devices such as smartphones or tablet computers. The citizen science-based approaches are intended to complement the traditional *in situ* and remote sensing-based approaches to environmental monitoring.

In this paper, we describe the design and implementation of a citizen science-based platform for early detection of CyanoHABs in inland water bodies. In particular, we address the problem of automated detection of harmful algal blooms (HABs) via analysis of images of inland water bodies. The image data are acquired using a variety of smartphones and tablet computers and communicated via popular OSM platforms such as Facebook, Twitter and Instagram to a central data server. One of the primary challenges in analyzing these images is ensuring robustness of the resulting algorithms to the wide variations in imaging parameters (i.e., camera resolution, viewpoint, scale and clutter) and ambient environmental conditions (i.e., weather and illumination). To this end we propose a deep learning-based approach to image feature extraction and classification for the purpose of HAB detection. We demonstrate the effectiveness of the proposed approach via experimental results on a diverse image dataset comprising of images of various inland water bodies acquired using a variety of smartphone cameras with wide variations in imaging parameters and ambient environmental conditions. With the

robust integration of multimodal data (i.e., tweets, images, videos, voice, and text) from localized detrimental environmental phenomena, we expect the proposed system to constitute a first step towards the development of an early warning, rapid response and optimal sensor deployment infrastructure for comprehensive environmental monitoring, sustainability and restoration.

6.3 Related Work

Most recent work on the detection of HABs in inland water bodies has focused on analysis of remotely sensed satellite images where the goal is to detect and map CyanoHABs based on the phycocyanin (PC) absorption characteristics of the reflected spectrum in the 620 nm wavelength band [18]. The 620 nm reflectance band is typically contaminated by chlorophyll-a (found in all phytoplankton and green algae), colored dissolved organic matter (CDOM) and totally suspended sediment (TSS) in the water body [3]. Consequently, extracting the characteristic CyanoHAB spectral signature entails a complex deconvolution operation [19]. To this end, several algorithms have been proposed for quantification of cyanobacterial concentration from satellite data: the single band ratio algorithm [32], semi-empirical algorithm [7], nested semi-empirical band ratio algorithm [35] and the maximum peak height (MPH) algorithm [20]-[23]. These algorithms are also used to compute various water quality indices. Matthews and Odermatt [23] use the MPH algorithm to compute the sun-induced chlorophyll fluorescence (SICF) peak, the sun-induced phycocyanin absorption and fluorescence (SIPAF) peak, the nor-

malized difference vegetation index (NDVI) and the backscatter and absorption-induced reflectance (BAIR) peak. Oyama et al. [27] compute the maximum chlorophyll index (MCI), cyanobacterial index (CI), floating algal index (FAI), normalized difference vegetation index (NDVI) and the normalized difference water index (NDWI) using a variant of the MPH algorithm.

Although satellite remote sensing can be used to monitor environmental parameters across a broad study area at a regional scale, it has serious practical limitations when monitoring a localized phenomenon such as the initiation of CyanoHABs in a water body. Satellite data cannot be used to develop an accurate early warning system for localized environmental phenomena because of their limited spatial, spectral, and temporal resolution. Issues such as mixed pixels, atmospheric interference, lack of a useful band center, and occasional poor site revisit time could potentially introduce serious errors in the predictive model or cause it to miss the environmental phenomenon altogether. *In situ* remote sensing using hyperspectral sensors mounted in the vicinity of the water body can address some of the shortcomings of satellite remote sensing [24]-[26]. However, the limited field of view of these sensors precludes dense spatial sampling of large water bodies.

Citizen science-based monitoring has the potential to effectively complement the aforementioned traditional approaches to environmental monitoring. Image data acquired by lay citizens using a variety of smartphones and tablets and communicated via popular OSM platforms such as Facebook, Twitter and Instagram could be potentially used as input to an early CyanoHAB warning system. A

major shortcoming of using such data is that they can be extremely noisy and also exhibit wide variations with respect to the imaging parameters (i.e., camera resolution, viewpoint, scale and clutter) and ambient environmental conditions (i.e., weather and illumination). In this paper, our goal is to perform early detection of HABs in inland water bodies using such image data acquired by lay citizens.

To ensure a low false alarm rate and low miss rate for the detection of HABs, the extracted image features and the image analysis algorithms need to be robust to noise and the aforementioned variations in imaging parameters and ambient environmental conditions. In recent times, deep (i.e., multi-layer) convolutional neural networks (CNNs or *ConvNets*) have been observed to be very effective in extracting robust high-level features (i.e., abstractions) from very diverse input image data [15]. In particular, generic image features generated by a deep *ConvNet* pretrained on the very diverse ImageNet dataset [31] have been shown to be capable of tackling a wide range of image recognition tasks such as object detection, fine-grained object classification, scene recognition, attribute detection and content-based image retrieval on very diverse datasets [34].

In the context of HAB detection, image data acquired by lay citizens are observed to contain a lot of clutter. Images may contain regions or surfaces other than those that can be classified as *lakes* or *ponds*, such as, *sand, sky, trees,* and *grass*, to cite a few. Since region classification is highly sensitive to noise and clutter, these noisy regions can seriously impact the HAB detection accuracy. To address this shortcoming, we propose a two-stage classification pipeline wherein the first stage entails a general classification procedure to extract *lake* regions from

the input image using a bottom-up agglomerative clustering procedure. This is followed by the second stage which comprises of a fine classification procedure to detect HABs in the extracted *lake* regions. We rely primarily on *texture* and *color* cues for region classification by leveraging the deep *ConvNet* features. Textural properties provide useful information for understanding scenes and discriminating between foreground objects and the background, especially when the shapes of the objects do not convey useful information.

The computer vision research community has invested significant research effort in texture recognition and classification [5, 17, 33]. Most of the works on texture recognition and classification have traditionally focused on material recognition [33, 17] and scene understanding [29]. In contrast, recognizing regions or surfaces in cluttered natural scenes containing multiple object categories (such as *cloud, sky, water* etc.) is relatively less explored. Color features, on the other hand, have been observed to perform well for object detection [10], image classification [11] and texture classification [14]. Khan et. al. [12, 14] have shown that color attributes combined with textural cues result in a significant improvement in the performance of texture-based region classification and recognition algorithms.

6.4 Contribution

Our primary goal is to detect and segment HABs in an input image. To accomplish this goal, we first perform a general classification of surfaces or regions in the input image to extract the *lake* regions, followed by a domain-specific classification of the

lake regions. This problem can be formulated as a general surface/region categorization problem, where the main challenge is that, surfaces in natural scenes are typically heterogeneous in terms of their color and texture attributes and highly cluttered. Recognizing surfaces or regions in natural scenes is particularly challenging on account of three reasons:

(1) It is extremely difficult to represent surfaces from natural images in appearance space on account of high intra-class variation. Textures of surfaces in natural images tend to appear as clutter in that they cannot be represented using a single texture category. For example, the surface of a *mountain* could be characterized as either *rocky*, *bushy*, *rugged* or more than one or all of the above (i.e., cluttered) and hence surface extraction using a single classifier would be highly inefficient.

(2) In some cases, lack of identifiable patterns renders texture recognition approaches impractical. For example, surfaces such as *sky* or *water* may not have any reliable textural attributes, which makes recognition challenging.

(3) Lack of high level cues such as shape, that could potentially aid recognition. Exploiting the global shape of an object has been shown to improve texture recognition performance in general [5]. However, in natural scenes many objects do not have a well-defined rigid shape.

The main contribution of this paper is the formulation of an *agglomerative clustering*-based optimization framework, inspired by [36], to integrate surface fragments or superpixels [30] such that the likelihood of surface categories that are too heterogeneous to be characterized by a single textural attribute is maximized. We define a heterogeneous surface category as a combination of multiple textu-

ral attributes, such that any subregion or superpixel of the entire heterogeneous surface, provides only a local view of the underlying surface category. It is only by combining or integrating multiple such subregions or superpixels, in a manner that maximizes the likelihood of the surface category, that we can recover the entire underlying surface reliably. Commonly used segmentation techniques such as Conditional Random Field (CRF) models use edge potentials to exploit similarities between regions or pixels. However in our case, the goal is to group superpixels such that the likelihood of the object increases when grouped than not. We chose agglomerative grouping-based techniques as they are simple, parameter free and, being insensitive to the choice of number of initial segments, are shown to perform well in such situations [36]. The proposed formulation can be viewed as similar to putting jig-saw puzzle pieces together, since the puzzle pieces once properly joined make more sense than before. For the purpose of texture recognition we extend the *deep filter banks* technique proposed by [4]. We improvise the approach proposed in [4] by omitting the fully-connected (FC)-CNN descriptor which provides object shape information (since we do not rely on the high-level shape description of the underlying surface) while retaining the Fisher vector (FV)-CNN descriptor.

In the proposed framework, we take a top-down approach by over-segmenting the image to generate superpixels and cluster the superpixels iteratively based on texture, color and semantic contextual cues. The pipeline of the proposed approach is shown in Figure 6.1. We integrate textural and color attributes along with other learned contextual cues, to address the challenges mentioned above. The combination of textural attributes with color-based attributes has been shown

to yield significant improvement in texture recognition performance [12, 14]. Semantic contextual cues are represented by pairwise spatial locations of surfaces learned during training, and have been shown to improve object recognition performance [8]. Since natural scenes are highly cluttered and contain textureless surfaces such as *sky* or *stagnant water*, incorporating semantic context into the proposed framework allows us to prune a lot of noisy classifications while recovering some of the textureless surfaces.

6.5 Overall System Description

The primary goal of the system is to perform early detection of HABs in inland water bodies using images acquired and communicated by lay citizens via popular OSM platforms such as Facebook, Twitter and Instagram. We have recently developed a mobile software application (i.e., app) that enables users to capture using smartphones and upload over the Internet, images of lakes across the world using the aforementioned OSM platforms. The mobile app also lets users provide a textual description of the lake, along with its *Global Positioning System* (GPS) coordinates and other relevant metadata, which is then published as a *tweet* upon submission. The purpose of the mobile app is to collect data as well as to create general public awareness about the seriousness of CyanoHAB contamination via publication of tweets on OSM platforms (in our case, on Twitter using a *hashtag*).

Data collection on a global scale from lay citizens via OSM is an easy and cost effective solution; the main problem lies in the effective processing of the

data, as explained in Section 6.4. The images acquired via OSM are typically noisy and may contain surfaces or regions other than lakes. We have developed a framework for detecting HABs from images collected by lay citizens to complement traditional approaches, such as *in-situ* sampling and satellite remote sensing, which also suffer from critical shortcomings as explained in Section 6.2. We propose a *two-stage* classification framework to first extract *lake* regions in the image using agglomerative clustering-based optimization, and then classify the extracted *lake* regions as containing HABs or not. The current framework is the first step in the design of a comprehensive automated system for early detection of CyanoHAB contamination. Our ultimate goal is to design a unified framework for robust detection of CyanoHABs in their early stages of development by integrating multi-modal data such as citizen-science image data and satellite image data, *in situ* sensor data, and high-level textual information obtained via analysis of *tweets* and OSM posts (by leveraging recent advances in Natural Language Processing).

6.6 Dataset

To evaluate the performance of our approach, we propose a new benchmark dataset consisting of images of natural scenes of inland water bodies (with and without HABs), obtained using our mobile citizen science app, as well as images gathered from other sources such as *Google Images* and *Flickr*. Figure 6.2 shows a few sample images from our dataset. Our dataset currently consists of 316 images, where we randomly assign 200 images to the training set and 116 images to the

test set. We have identified 5 surface categories that occur predominantly and consistently in all images across our dataset, i.e., *lake (clear), tree, grass, sky* and *lake (HAB)*. We have also identified a few more categories such as *mountain, sand,* etc.; but since we currently do not have sufficient training or test instances for these categories, we label them as a single background category. The general paucity of images in our dataset can be attributed to the fact that we are dealing with a very specific application for which there are very few publicly available benchmark images; hence our focus on acquisition and analysis of crowd-sourced images. Owing to the nature of the application, the crowd-sourced images typically exhibit a significant class imbalance. For example, the images containing lakes with HABs are significantly fewer than those containing clear lakes. But, since our primary goal is to detect HAB-infested lake regions, we aim to obtain as many images of HAB-infested lakes as possible; images of clear lake regions can be efficiently used as a negative set for hard mining of negative instances. We have also developed an annotation toolbox that allows users to manually segment different regions of an image by drawing boundaries around them and to identify and annotate (i.e., tag) these regions. The toolbox also allows users to report noisy images, i.e., images that are not of natural scenes or ones that do not contain lake regions.

Figure 6.2: Sample images from the proposed benchmark dataset.

6.7 Agglomerative Clustering-based Optimization

The goal of the optimization step in the proposed framework is to group superpixels so as to maximize the likelihood of a heterogeneous or cluttered surface category. The proposed optimization procedure effectively integrates the extracted textural and color cues with the learned contextual cues, by iteratively clustering the super-segments or superpixels using a convergence criterion based on a pixel-level *entropy* function [36]. The purpose of the proposed optimization procedure is to arrive at an optimal labeling of each superpixel with a label of one of the predefined surface categories. The proposed agglomerative clustering combines multiple

cues (i.e., color, textural, and contextual) while minimizing a pixel-level entropy function via grouping of superpixels so as to maximize the appearance of a heterogeneous surface category. The agglomerative clustering procedure guided by the aforementioned convergence criterion eliminates the need for deciding the final number of clusters before hand.

An individual superpixel $S_i \in S$, provides only a *local* view to the classifier, which is typically insufficient to ensure *global* agreement with its surface category, especially when dealing with surface categories that contain heterogeneous appearance patterns. The goal of the agglomerative clustering procedure is to iteratively group the superpixels to reduce the *overall* entropy and thereby obtain *globally* meaningful regions. The optimal segmentation G^* is given by the minimization of the pixel-level entropy function as follows:

$$G^* = \arg\min_{G} \quad U(G) \tag{6.1}$$

where

$$U(G) = \sum_{l=1}^{N} \sum_{j=1}^{|S|} -P(l|S_j) \log(P(l|S_j)|S_j| \tag{6.2}$$

Here $U(G)$ is the *pixel-level entropy* of the segmentation state G, $|S_i|$ denotes the total number of pixels in the superpixel S_i , $|S|$ denotes the total number of superpixels in the test image I, and N denotes the number of surface categories.

$P(l|S_i)$ represents the probability of the segment, superpixel or region S_i belonging to a class $l \in N$ and is given by:

$$P(l|S_i) = w^{(1)}P(l|A_{S_i}) + w^{(2)}P(l|D_{S_i}) \tag{6.3}$$

where $0 \leq \{w^{(1)}, w^{(2)}\} \leq 1$ and $w^{(1)} + w^{(2)} = 1$. $P(l|A_{S_i})$ is the probability of a region or superpixel S_i belonging to the class l, given its appearance score A_{S_i}. We use Platt's scaling [28] to convert Support Vector Machine (SVM) classification scores to probabilities. Similarly $P(l|D_{S_i})$ represents the probability of the superpixel S_i belonging to class l given its 2D location, as explained in Section 6.7.1. Weights $\{w^{(1)}, w^{(2)}\}$ are learned by maximizing the classification likelihood on the training data.

For an image I, we initially compute a set of superpixels $\{S_i|i = 1 \dots K\}$ using the method discussed in Section 6.7.2. Subsequently, for each superpixel S_i, the probability that the superpixel belongs to class $l \in L$, is computed using equation (6.3). A pair of superpixels (S_i, S_j) can be merged if (a) the superpixels are adjacent to each other, and (b) the merging reduces the overall pixel-level entropy. The impact of merging a pair of superpixels (S_i, S_j) on the overall pixel-level entropy is denoted by $A(S_i, S_j)$ and is given by:

$$A(S_i, S_j) = \begin{cases} R(S_i, S_j) & \text{if}, S_j \in \mathcal{N}_{S_i} \\ 0 & \text{otherwise}, \end{cases} \tag{6.4}$$

where $\mathcal{N}_{S_i}$ is the adjacency neighborhood of S_i. $R(S_i, S_j)$ is computed as follows:

$$R(S_i, S_j) = U(G) - U(G \cup \{S_k\} \backslash \{S_i, S_j\})(|S_i| + |S_j|) \tag{6.5}$$

where $S_k = S_i \cup S_j$. From equations (6.2), (6.3) and (6.5),

$$\begin{aligned} R(S_i, S_j) = \sum_{l=1}^{N} P(l|S_k) \log(P(l|S_k)|S_i + S_j| \\ - \Bigg\{ \Bigg(\sum_{l=1}^{N} P(l|S_i) \log(P(l|S_i)|S_i| \Bigg) \\ + \Bigg(\sum_{l=1}^{N} P(l|S_j) \log(P(l|S_j)|S_j| \Bigg) \Bigg\} \end{aligned} \tag{6.6}$$

In order to perform agglomerative clustering of the superpixels, we randomly initialize a set of seed superpixels $S_{init} \subset S$. For each superpixel in S_{init}, we merge it with each of its neighbors and choose the pair that results in the maximum reduction in the overall pixel-level entropy. Also we perform multiple random initializations of S_{init} to avoid convergence to a local minimum.

6.7.1 Training

Textural Features

We use FV-CNN features [4] to characterize the texture of each image segment. The FV-CNN features are an orderless representation of the CNN features and hence ideally suited for texture recognition. FV-CNN features are shown to outperform most state-of-the-art texture detectors [4]. We improvise the approach

in [4] by removing the FC-CNN features from their architecture. Since the FC-CNN features provide high-level shape information, they were deemed redundant for our purpose. The FV-CNN is obtained by computing the FV representation over the CNN features. Also, for practical purposes, we extract FV-CNN features for the entire image once, and then compute the scores and histograms individually for each super-segment. This allows efficient search for combinations of superpixels that maximize the appearance of heterogeneous surfaces. To encode the extracted features, we compute η (where $\eta = 64$) Gaussian priors from the data for robust representation of the underlying feature distribution, and using the priors we embed the data into an FV.

Color Features

Instead of using the traditional linguistic label-based representation of colors, we extend the *discriminative color descriptors* (DCDs) [13] approach that clusters color values based on their discriminative power in a classification task. We use the DCDs extracted using [13] and compute a FV representation similar to the one for textural features described in Section 6.7.1. We compute η (where $\eta = 32$) Gaussian priors from the data for embedding the data into an FV representation.

Location Context

Since surfaces in natural scenes are cluttered and sometimes textureless, the surface classification results could be potentially ambiguous. To address this problem, we learn contextual information about the surfaces and use it to resolve the ambiguity.

Location context denotes the information about the location of a surface with respect to other surfaces that can be used as a prior to improve classification performance. Location context is easy to learn and has been shown to perform well in object recognition tasks. In our case, we learn the location context in order to predict the likelihood of a superpixel belonging to a specific surface category. To learn the location context, we simply average the occurrences of each surface category at each location. For each surface category, we load all the annotated instances and compute binary instance masks. By projecting the instance masks into a 1×1 2D unit space, (which can be denoted as the *standard image*) we first compute the likelihood of occurrence of each surface category at each point in the standard image space. We then compute the joint probability of occurrence for each surface category, using the computed individual likelihoods. In addition, we also compute the pairwise (conditional) probabilities, i.e., given that a surface category occurs at a certain location in the standard image space, we compute the likelihood of occurrence for other surface categories at each point in the standard image space.

We used 200 images from the training set to learn the location context. Figure 6.3 is a visualization of the individual and joint probability of occurrences for each surface category, learned from the training set using the method described above. As shown in Figure 6.3, surface categories such as *lake* or *tree*, appear more uniformly distributed since they occur in almost every image in our dataset. In contrast, categories such as *mountain* or *grass* have a patchy appearance in Figure 3 since they occur much less frequently.

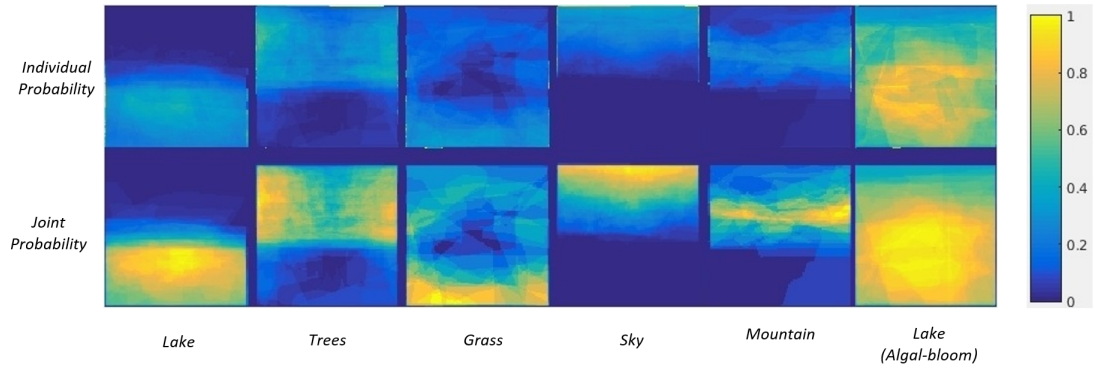

Figure 6.3: Learned location context: The *first* row represents the joint probabilities and the *second* row represents the individual probability of occurrence of each surface category. *Note:* For better visualization, the joint probability values in the second row are normalized to 1.

6.7.2 Testing

Image segmentation using superpixels

To generate superpixels, we use the simple linear iterative clustering (SLIC) procedure [1] to over-segment the image. In our experiments, we use the SLIC procedure with smaller region sizes and a higher value for the regularizer, so that the resulting segmentation aids the iterative clustering procedure during optimization.

Agglomerative Clustering

We perform surface categorization for each superpixel $S_i \in S$. A set of trained SVM-based appearance classifiers followed by Platt's scaling is used to obtain the probability values for each positive response. The agglomerative clustering procedure is outlined in Algorithm 4. In order to prevent the optimization from

Algorithm 4 Agglomerative Clustering

1: Randomly initialize a set of seed superpixels $S_{init} \subset S$;
2: Compute pixel-level entropy $U(G)$;
3: **do**
4: Assign $\hat{U}(G) = U(G)$
5: **for each** superpixel $S_i \in S$, **do**
- Compute $P(l|S_i)$ using equation (6.3).
- Compute $A(S_i, S_j)$ for each neighbor S_j using equation (6.4);
6: Merge the superpixel pair with the least entropy.
7: Compute combined pixel-level entropy $U(G)$,
8: **while** $U(G) < \hat{U}(G)$

converging to a local minimum, we perform multiple initializations of the agglomerative clustering procedure by randomly generating different initial sets of seed superpixels $S_{init} \subset S$ (Algorithm 4).

Detection of HABs

Once we extract the pixels that correspond to *lake* regions, we deploy an instance-level *binary* SVM classifier trained to further classify the *lake* regions as *HAB* vs. *clear* lake surfaces. We train the SVM classifier on features extracted from *HAB* and *clear* lake images.

6.8 Experimental Results

We demonstrate the performance of the proposed texture recognition technique which is modeled as a multi-class classification problem followed by a fine-grained

classification problem. We first extract *lake* regions from the test image followed by fine-grained classification of the extracted *lake* region to determine whether it contains an HAB or not. To the best of our knowledge, most existing approaches rely on either satellite images or *in situ* water sampling to detect HABs in water bodies; there are very few full-fledged image-based approaches to detect HABs [16].

6.8.1 Agglomerative Clustering-based Optimization

Results of the proposed agglomerative clustering-based optimization procedure (i.e., *Stage 1* classification) for joint classification and segmentation of multiple image surfaces are shown in Table 6.1. The results demonstrate the usefulness of *location context* in reducing ambiguity and enhancing classification accuracy, as well as the improvement in classification accuracy due to the agglomerative clustering-based optimization procedure. Figure 6.4 shows qualitative results for joint classification and segmentation of multiple image surfaces. Similarly, Figure 6.5 demonstrates the qualitative improvement in classification accuracy due to the inclusion of location context.

6.8.2 Classification of lakes: HAB vs. clear

Results of the proposed scheme for binary classification of *lake* regions into *clear* and *HAB* categories based on textural (FV-CNN) and color (DCD) features (i.e., *Stage 2* classification) are given in Table 6.2. The proposed scheme is compared with the scheme of Lazorchak et al. [16] which performs binary classification of lake regions into *clear* and *HAB* categories using color histogram features. We have

Table 6.1: Performance analysis of the proposed system for joint classification and segmentation of image surfaces using precision and recall measures. The performance of the proposed system is compared to the performance of the scheme described in [36] which does not use *agglomerative clustering* and the performance of proposed system without the use of *location context*.

Categories	Proposed System		w/o. Agg. Clustering [36]		w/o Location Context	
	Precision	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*
Lake (clear)	**0.78**	**0.61**	0.54	0.52	0.74	0.58
Tree	**0.71**	**0.58**	0.60	0.44	0.67	0.51
Grass	**0.68**	0.71	0.57	0.58	0.66	**0.73**
Sky	**0.84**	**0.88**	0.72	0.66	0.77	0.80
Lake (HAB)	**0.71**	**0.68**	0.61	0.46	0.64	0.48

Figure 6.4: Qualitative results of the proposed agglomerative clustering-based optimization for joint detection and segmentation of clear and HAB *Lake* regions. The *left* column shows test images, where the *right* column shows extracted lake regions.

Figure 6.5: Improvement in classification accuracy due to location context. *Left:* Test image; *Middle:* Extracted *lake* region without location context; *Right:* Improvement in classification using location context.

implemented the scheme of Lazorchak et al. [16] using hue-saturation-intensity (HSV) histograms with 300 bins and a binary SVM classifier. Note that the scheme of Lazorchak et al. [16] is restricted to binary classification and, unlike the proposed framework, does not perform segmentation of multi-surface (i.e., heterogeneous) images. Table 6.2 shows that, given the segmented *lake* regions, the proposed scheme performs significantly better than the scheme of Lazorchak et al. [16] in classifying lake regions as either *clear* or *HAB* surfaces.

Table 6.2: Comparison of HAB detection performance within *lake* regions (using precision and recall) of the proposed approach and that of Lazorchak et al. [16].

	Proposed Scheme	Lazorchak et al. [16]
Precision	**0.87**	0.764
Recall	**0.844**	0.81

6.9 Multimodal Learning for Detection of Cyanobacterial Algal Blooms

In this section, we propose an end-to-end trainable convolutional neural network architecture that can jointly learn from multiple modalities simultaneously, to identify Cyanobacterial algal blooms (CyanoHABs) from images of other algal-blooms. While stock photos acquired by lay citizens are rich in textural cues, they are hard to obtain and require extensive amounts of annotations, whereas the satellite images are relatively easy to obtain with little need for manual annotations. On the other hand, the textural cues of satellite images are less distinctive and are significantly inferior in terms of representational power to that of stock photos in comparison. Thus, we propose a model that learns jointly to classify cyano *vs.* non-cyano images, using a multi-modal neural network with intermediate fusion; a technique that combines two separate neural network designed for the task of classification of respective modalities, such that both networks learn a better representation given the noisy ground truths from both image modalities.

6.9.1 Dataset

We have extended our dataset comprising stock images introduced in section 6.6, from 316 images to a total of 1673 images of algal blooms, in which 855 images are manually classified as CyanoHABs and 818 images as containing non-cyano or other types of algal blooms. Furthermore, for improving training accuracy, we have also gathered and included a collection of 2136 clear lake images to our dataset. Sample images from the newly introduced dataset is shown in Figure 6.6.

In addition to stock photos, we also have collected satellite images of known cyano-infested and clear lakes. We obtained Tier-1 Surface Reflectance data (T1_SR) of Landsat 8 satellite from January of 2013 to August of 2018 for 9 different lakes. Alongside RGB images of the satellite data, we also used NIR (Near Infra Red) to compute NDVI (Normalized Difference Vegetation Index) that provides an approximation of the per-pixel intensity of chlorophyll. We categorize these satellite images as cyano-positive or not, based on the following criteria: if the image belong to a known cyano infested lake, and the average NDVI (signifies chlorophyll presence) is high (Figure 6.7); such classification is required even for a known cyano-infested lake, as the cyano infestation is seasonal. We also use the FMask algorithm [38], to identify pixels of clouds, cloud shadows, pixels of water, which we use to isolate the images that are informative from noisy images. Figure 6.7 shows an example of cloud covered images before and after applying FMask.

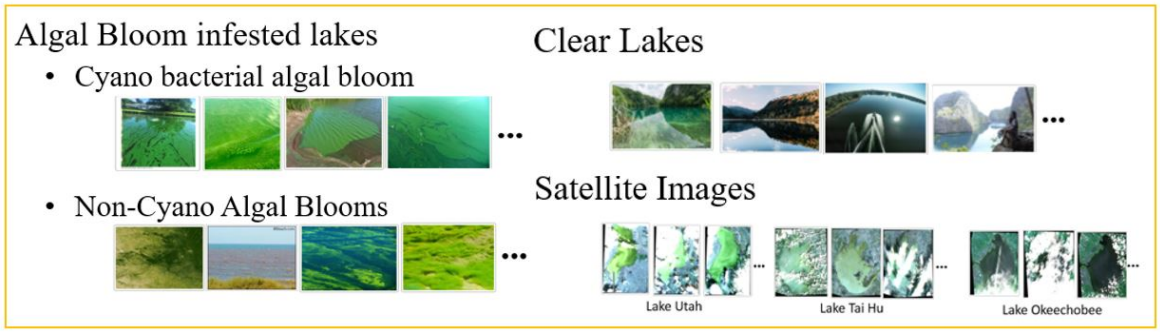

Figure 6.6: Sample images from the extended dataset.

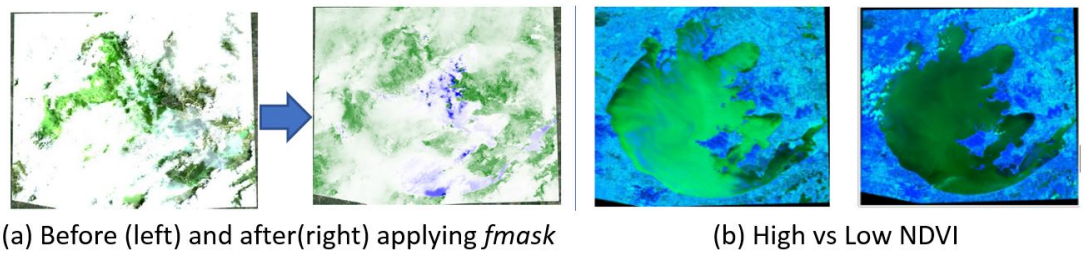

Figure 6.7: Examples of use of Fmask algorithm [38] for removal of noisy pixels in satellite images (*left*); In this example of a noisy image, the pixels in blue (color coded) are the only informative pixels of the waterbody, and the rest are lost due to cloud cover. Demonstration of images with low vs high NDVI index (*right*).

6.9.2 Training

We propose a end-to-end trainable neural network architecture with multi-modal fusion that learns cross-modality features to identify images of cyano-bacterial algal blooms from across both satellite and stock imagery. Jointly learning from multiple modalities have been shown to mutually improve the performance for both the modalities [39]. We extend the multi-modal end-to-end trainable convolutional neural network architecture of [39] for classification of cyano vs. non-cyano bacterial blooms. The network architecture as shown in Figure 6.8 follows a traditional CNN architecture of VGGNet [40] till the convolutional layers, followed by a dense layer (of size 512), which is then concatenated and further separated to individually output the binary classification.

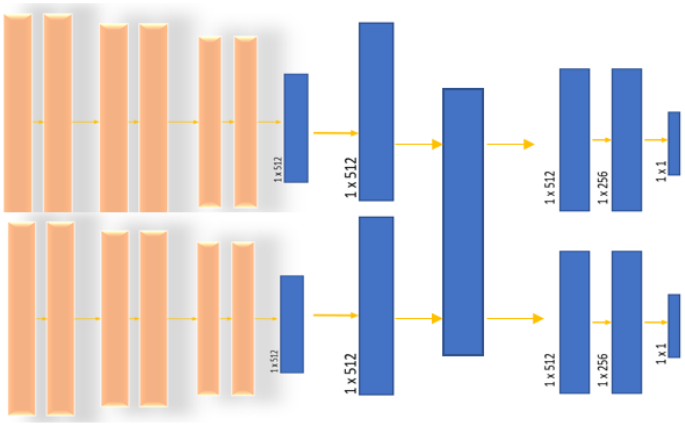

Figure 6.8: Pipeline of the proposed Multimodal CNN architecture.

6.9.3 Evaluation

We compare the proposed method against the Lazorchak et al. [16] and our method introduced in 6.4. The proposed method significantly outperforms [16] and performs comparably against our method from 6.4, for the task of Cyano *vs.* non-cyano classification on stock images. Table 6.3 shows the classification accuracy of the proposed method for classification of cyano *vs.* non-cyano on stock images. On the other hand, the precision and recall for classification on satellite images are 0.772 and 0.745 respectively, marginally higher its CNN baseline with a precision and recall of 0.75 and 0.731 respectively. Thus, the use of multi-modal learning shows reasonable improvement in classification accuracy for satellite data, over its traditional counterpart, whereas for stock images the use of multi-modal learning shows no considerable gains.

Table 6.3: Comparison of CyanoHAB detection performance of the proposed multi-modal learning framework on stock photos with that of Lazorchak et al. [16] & Kumar et. al. (2018) (deep filter banks), on the extended dataset 6.9.1.

	Proposed Scheme	DFB* (Sec. 6.4)	Lazorchak et al. [16]
Precision	**0.782**	0.78	0.64
Recall	**0.811**	0.75	0.681

6.10 Conclusions and Future Work

We have demonstrated quantitatively that the proposed *agglomerative clustering*-based approach for automatic detection of HABs using images acquired by lay

citizens, can yield good results, providing an effective and cost-efficient means for monitoring of HABs in inland water bodies. The current system is an initial step towards a design of an automated early detection, warning and rapid response system, to mitigate the detrimental effects of CyanoHAB contamination. In future, we intend to integrate multi-modal information, such as citizen-science image data, satellite image data, *in situ* sensor data and textual information obtained via tweets and OSM posts within the current framework. Our ultimate goal is to design a unified framework for robust detection of CyanoHABs in their early stages of development.

Acknowledgment: This research is supported by the National Science Foundation (NSF) under Grant No. 1442672. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

Bibliography

[1] Achanta, R. et al. (2010) SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 34(11), pp. 2274-2282.

[2] Backer, L.C. (2002). Cyanobacterial harmful algal blooms: developing a public health response, *Lake Reservoir Management*, Volume 18(1), pp. 20-31.

[3] Brezonik, P.L. et al. (2015). Factors affecting the measurement of CDOM by remote sensing of optically complex inland waters, *Remote Sensing Environment.*, Volume 157, pp. 199-215.

[4] Cimpoi, M. et al. (2015) Deep filter banks for texture recognition and segmentation, *In Proceedings of IEEE Conference on Computer Vision And Pattern Recognition*, pp. 3828-3836.

[5] Cimpoi, M. et al. (2015). Describing textures in the wild, *In Proceedings of IEEE Conference on Computer Vision And Pattern Recognition*, pp. 3606-3613.

[6] Codd, G.A. et al. (1999). Cyanobacterial toxins, exposure routes and human health, *Eur. Journal of Phycology*, Volume 34(4), pp. 405-415.

[7] Dekker A.G. & H.J. Hoogenboom (1997). *Operational tools for remote sensing of water quality: A prototype toolkit.* The Netherlands Remote Sensing Board (BCRS) Report 96-18, Vrije Universiteit Amsterdam, Institute for Environmental Studies.

[8] Divvala, S.K. et al. (2009). An empirical study of context in object detection. *In Proceedings of IEEE Conference on Computer Vision And Pattern Recognition*, pp. 1271-1278.

[9] Falconer, I.R. & A.R. Humpage (2005). Health risk assessment of cyanobacterial (bluegreen algal) toxins in drinking water, *International Journal of Environmental Research and Public Health*, 2(1), pp. 43-50.

[10] Khan, F. S. et al. (2012) Color attributes for object detection, *In Proceedings of IEEE Conference on Computer Vision And Pattern Recognition*, pp. 3306-3313.

[11] Khan, F.S. et al. (2012). Modulating shape features by color attention for object recognition, *International Journal of Computer Vision*, Volume 98(1), pp. 49-64.

[12] Khan, F.S. et al. (2013). Evaluating the impact of color on texture recognition, *In Proceedings of International Conference on Computer Analysis of Images and Patterns*, Volume 8047, pp. 154-162.

[13] Khan, R. et al. (2013). Discriminative color descriptors, *In Proceedings of IEEE Conference on Computer Vision And Pattern Recognition*, pp. 2866-2873.

[14] Khan, F.S. et. al. (2015). Compact color/texture description for texture classification, *Pattern Recognition Letters*, Volume 51(1), pp. 16-22.

[15] Krizhevsky, A. et al. (2012). Imagenet classification with deep convolutional neural networks, *advanced Neural Information Processing Systems*, Volume 1, pp. 1097-1105.

[16] Lazorchak, J. et al. (2016). Harmful algal bloom smart device application: using image analysis and machine learning techniques for early classification of harmful algal blooms, *In Proceedings of SETAC Europe 2016*.

[17] Leung, T. & J. Malik (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons, *International Journal of Computer Vision*, Volume 43(1), pp. 29-44.

[18] Li, L-H. et al. (2015). Remote sensing of freshwater cyanobacteria: An extended IOP Inversion Model of Inland Waters (IIMIW) for partitioning absorption coefficient and estimating phycocyanin, *Remote Sensing Environment.*, Volume 157, pp. 9-23.

[19] Lunetta, R.S. et al. (2015). Evaluation of cyanobacterial cell count derived from MERIS imagery across the eastern USA, *Remote Sensing Environment.*, Volume 157, pp. 24-34.

[20] Matthews, M.W. et al. (2012). An algorithm for detecting trophic status (chlorophyll-a), cyanobacterial-dominance, surface scums and floating vegetation in inland and coastal waters, *Remote Sensing Environment.*, Volume 124, pp. 637-652.

[21] Matthews, M.W. (2014). Eutrophication and cyanobacterial blooms in South African inland waters: 10 years of MERIS observations, *Remote Sensing Environment.*, Volume 155, pp. 161-177.

[22] Matthews, M.W. & S. Bernard (2015). Eutrophication and cyanobacteria in South Africa's standing water bodies: A view from space, *South African Journal of Science*, Volume 111, Nos. 5/6, May/June, pp. 1-8.

[23] Matthews, M.W. & D. Odermatt (2015). Improved algorithm for routine monitoring of cyanobacteria and eutrophication in inland and near-coastal waters, *Remote Sensing Environment.*, Volume 156, pp. 374-382.

[24] Mishra, S. et al. (2009). A novel model for predicting phycocyanin concentrations in lab cultured cyanobacteria: A proximal hyperspectral remote sensing approach, *International Journal of Remote Sensing*, Volume 1, pp. 758-775.

[25] Mishra S. et al. (2014). Bio-optical inversion in highly turbid and cyanobacteria dominated waters, *IEEE Transactions on GeoScience Remote Sensing*, Volume 52(1), pp. 375-388.

[26] Ogashawara, I. et al. (2013). A performance review of reflectance based algorithms for predicting phycocyanin concentrations in inland waters, *Remote Sensing.*, Volume 5(10), pp. 4774-4798.

[27] Oyama, Y. et al. (2015) Distinguishing surface cyanobacterial blooms and aquatic macrophytes using Landsat/TM and ETM+shortwave infrared bands, *Remote Sensing Environment.*, Volume 157, pp. 35-47.

[28] Platt, John. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, Volume 10(3), pp. 61-74.

[29] Quattoni, A. & A. Torralba (2009). Recognizing indoor scenes, *In Proceedings of IEEE Conference on Computer Vision And Pattern Recognition*, pp. 413-420.

[30] Ren, X-F. & J. Malik (2003). Learning a classification model for segmentation, *In Proceedings of IEEE International Conference on Computer Vision*, Volume 1, pp. 10-17.

[31] Russakovsky, O. et al. (2015). ImageNet large scale visual recognition challenge, *International Journal of Computer Vision*, Volume 115(3), pp. 211-252.

[32] Schalles, J. & Y. Yacobi (2000). Remote detection and seasonal patterns of phycocyanin, carotenoid and chlorophyll-a pigments in eutrophic waters, *Hydrobiology special Issues advanced Limnology*, Volume 55, pp. 153-168.

[33] Sharan, L. et al. (2013). Recognizing materials using perceptually inspired features, *International Journal of Computer Vision*, Volume 103(3), pp. 348-371.

[34] Sharif Razavian, A. et al. (2014). CNN features off-the-shelf: An astounding baseline for recognition, *In Proceedings of IEEE Conference on Computer Vision And Pattern Recognition Workshops*, pp. 806-813.

[35] Simis, S.G.H. et al. (2005). Remote sensing of the cyanobacterial pigment phycocyanin in turbid inland water, *Limnol. Oceanography*, Volume 50(1), pp. 237-245.

[36] Vijayanarasimhan, S. & K. Grauman (2010) Top-down pairwise potentials for piecing together multi-class segmentation puzzles. *In Proceedings of IEEE Computer Vision And Pattern Recognition Workshops*, pp. 25-32.

[37] Wilde, S.B. et al. (2005). Avian vacuolar myelinopathy (AVM) linked to exotic aquatic plants and a novel cyanobacterial species, *Environmental Toxicology*, Volume 20, pp. 348-353.

[38] Zhu, Z., & Woodcock, C. E. (2012). Object-based cloud and cloud shadow detection in Landsat imagery. Remote sensing of environment, 118, 83-94.

[39] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. In Proceedings of the 28th international conference on machine learning (ICML-11) (pp. 689-696).

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

Chapter 7

Conclusions & Future Work

While the conclusions and future work for each proposed method have been discussed in their respective chapters, this chapter is devoted to summarizing what the overall research has accomplished thus far and the overall directions for future research. The chapter is organized as follows: We summarize the overall contributions of this dissertation, and where they lie in the global scheme of things relating to computer vision, followed by a brief summary of the future directions of our research.

7.1 Summary of Contributions

The overall contributions of the thesis are summarized as follows:

1. We replace the need for human-designed 3D CAD models and manual pose/part annotations for 3D object detection.

2. We estimate the continuous pose of an object geometrically, rather than training appearance regressors that learn to categorize pose into viewpoint bins.

3. We perform object detection from a single image, by estimating pose and recovering the underlying 3D shape jointly, driven by geometric reasoning.

4. We design an automatic 3D part discovery module (for objects) that replaces the need for human annotation of parts on objects.

5. We design a two-pronged (sparse and dense) 3D object detection model, where the sparse model reasons about object part motion, and the dense model renders a comprehensive 3D reconstruction (instead of a wireframe-based or coarse 3D reconstruction).

6. We design a 3D scene reconstruction scheme through monocular depth prediction using LSTMs. Reasoning about the scene depth is modeled not only as a function of the appearance of the scene (i.e., the 2D image), but also as a function of time using the image frames in the temporal neighborhood of the current frame.

7. We demonstrate the ability of the LSTM to reason temporally, by extrapolating depth maps for future/unseen frames, without the LSTM being explicitly trained to do so.

8. We harness the power of adversarial training to improve the depth prediction accuracy.

9. We use context-aware adversarial examples to train the adversarial networks to improve the depth prediction accuracy, and to ensure network stability during training.

10. We present state-of-the-art results for depth prediction in comparison to a similar class of techniques (at the time of publication).

11. We demonstrate the application of deep learning to a real-world problem of detecting algal and cyanobacterial blooms. We develop a deep learning- and agglomerative clustering-based detection and early warning system that learns from general crowd-sourced images.

7.2 Future Work

One of the prominent directions that we intend to focus on for building better 3D object and scene reconstruction pipelines is to combine both the semantic (object-driven) reasoning about the image and generic reconstruction of the whole scene. Inclusion of object-driven reasoning (Chapters 2 and 3) into the deep learning pipelines (Chapters 4 and 5) supplements the learning methods with global pose and shape priors. For example, estimating the pose of a car in a scene provides the scale, approximate global pose, and a depth prior for the scene.

In addition, we also intend to address some of the shortcomings with the pose prediction network (used in Chapter 5) for the task of monocular depth prediction detailed in Chapter 4. The learning-based pose estimation techniques tend to regress the pose given image sequences resulting in a system that (1) ignores

image saliency causing the inclusion of noisy correspondences that affect overall pose prediction, and (2) is unable to adapt to varying frame rates. A possibility to improve would be to replace optical flow based pose prediction network in the pipeline with end-to-end differentiable interest point based correspondence estimation pipeline to compute pose predictions.

Furthermore, most of current state-of-the art monocular depth predictions techniques do not propagate the depth map generated for an image to the consecutive or successive frames, leaving the system to predict depth maps for every frame from scratch. Inspired by traditional SLAM models, it would be worth building a comprehensive system, in the direction of [1], for depth and pose prediction that builds a global understanding of pose and depth as opposed to exploiting local correspondences alone.

Bibliography

[1] Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., & Davison, A. J. (2018). CodeSLAM-Learning a Compact, Optimisable Representation for Dense Visual SLAM. arXiv preprint arXiv:1804.00874.