Autonomous Pathfinding in Simulated 3D Environment

Dr. O.P. Verma*, Sidharth Raja†, Tanuj Bhatia‡ and Akshat Mishra§
Department of Computer Science and Engineering
Delhi Technological University, New Delhi, Delhi - 110042

***Abstract*—This paper presents an approach for navigating an autonomous agent in a simulated room environment with multiple obstacles. It presents a technique for depth map estimation using stereoscopic images - a vital intermediate technique for autonomous navigation and pathfinding. It also propose a path planner interfaced with the depth map for locally planning a sequence of steps to the desired location. A single RGB camera does not contain sufficient information to estimate depth with satisfactory accuracy. Depth cameras are significantly more expensive than RGB cameras, and hence would not be suited for projects requiring lower budget constraints. Hence a middle ground is required which requires depth calculation from limited available input. Depth estimation is accomplished with a convolutional neural network trained on synthetic randomly generated stereo image data. The path planner comprises a short-term path planner for obstacle avoidance, and a long-term path planner for approaching the destination co-ordinates. The results show that this approach yields a factor of 1.8 times the steps of human performance on the same task.**

I. Introduction

The path planning problem for mobile robots can be defined as the search for a path which a robot (with specified geometry) has to follow in a described environment, in order to reach a particular position and orientation B, given an initial position and orientation A. [1].

Path planning can be divided into two categories - global and local planning. If the knowledge of the environment is known beforehand, the global path can be planned before the robot starts to move. If there exists no prior knowledge about the environment, the path must be computed online and the agent must avoid obstacles in real-time [2].

In the past year, neural network based models have shown remarkable results in a sensory role for obtaining depth information from standard RGB camera images [3], which motivated our inquiry for the use of neural networks to enhance the sensory data used by agents.

This paper describes an approach for an agent to navigate from source co-ordinates to destination co-ordinates while avoiding obstacles that may appear on it's path. Through this project, we also introduce an autonomous control policy to govern agents.

II. Related Work

Synthetic generated imagery is a useful tool in computer vision as it allows for generation of a large number of arbritrary examples in various positions and orientations. This would allow a more precise sample set than would ordinarily be possible with real world data. It also provides us with precise groundtruth data such as depth information which might not be trivial to collect from real data.

Synthetically generated data has been used in several instances in the past to augment performance. Taylor et al. present a system called Object Video Virtual Video (OVVV) [4] based on Half-life for evaluation of tracking in surveillance systems.

Peng et al. [5] use synthetic CAD generated images models to fine-tune on the object detection task and significantly outperform previous methods with this approach. Lim et al. [6] , and Aubry et al. use CAD models for detection and object alignment in the image. Aubry and Russell use synthetic RGB images rendered from CAD models to analyze the response pattern and the behavior of neurons in the commonly used deep convolutional networks.

There has recently been increased interest in using video game data to train computer vision models [7]. Although video games generate images from a finite set of textures, there is variation in viewpoint, illumination, weather, and level of detail which could provide valuable augmentation of the data.

Nikolaus Mayer et al. [8] present a method for visual scene generation using an open source 3D modelling software Blender3D which includes stereo color images and ground truth for bidirectional disparity, bidirectional optical flow and disparity change, motion boundaries, and object segmentation.

Existing methods for disparity map estimation include semi global matching (SGM) proposed by Hirschmuller [9] and MC-CNN, proposed by Zbontar and LeCunn, a convolutional neural network which compares image patches to initialize the stereo matching cost [10].

Mayer et. al. also propose a convolutional neural network, DispNet, in their work which they evaluate against the beforementioned approaches. They find that DispNet outpermforms the other two in various datasets [8]. The neural network architecture mentioned in our approach is based on DispNet.

There has been historical prior work in localised path planning, where the mobile robot aims to use local sensory information in a reactive fashion [11].

Ziegler et al. present a mechanism for depth sensor based detection of target object and for generation of trajectories for a robot-aided scanning process for inspection and verification processes. [12].

Fig. 1. Shapenet Dataset

III. Method

In this section, we describe our learning strategy for depth map estimation and control policy for the agent in the environment.

A. Depth Map Estimation

A blender script was created to generate randomized 3D situations for analysis. This stochastic approach generates a large number of 3D scene environments and their associated depth maps. Through this training, we aim to obtain a neural network that takes input as stereo images and output the depth parity. This depth parity would later used as part of the short-term path planner in the control policy. Shapenet [13], an annotated, large-scale dataset of 3D shapes was used as obstacles in the training phase of the CNN for depth parity estimation, and as the obstacles for the agent to circumnavigate in the test phase.

A dataset of 20,000 training examples was generated using 3D models from the Shapenet dataset. For each example, 8-10 models were chosen at random and placed in the scene. The placement and rotation of each model is random, and 3 scenes are rendered:

1) Left camera stereo image
2) Right camera stereo image
3) Depth map

This dataset of 20,000 images was normalized along each pixel to aid the training process.

A Convolutional Neural Network for estimating depth maps using a stereo image pair was trained. The network accepts as input 2 grayscale images and produces an estimate of the depth map corresponding to the images [8] [14]. A loss function of mean squared error was used to as the objective. The network was trained using an Adam Optimizer [15], with learning rate 0.01. Hold out validation with split 10% was performed i.e 18,000 training examples were used as the training set and 2000 as validation set. The validation loss and training loss at each epoch were recorded. The training was halted once validation loss did not improve significantly. Loss function used:

$$L(x_n, y_n) = \sum_{i=1}^{200}\sum_{j=1}^{200}(y_{ij} - x_{ij})^2$$

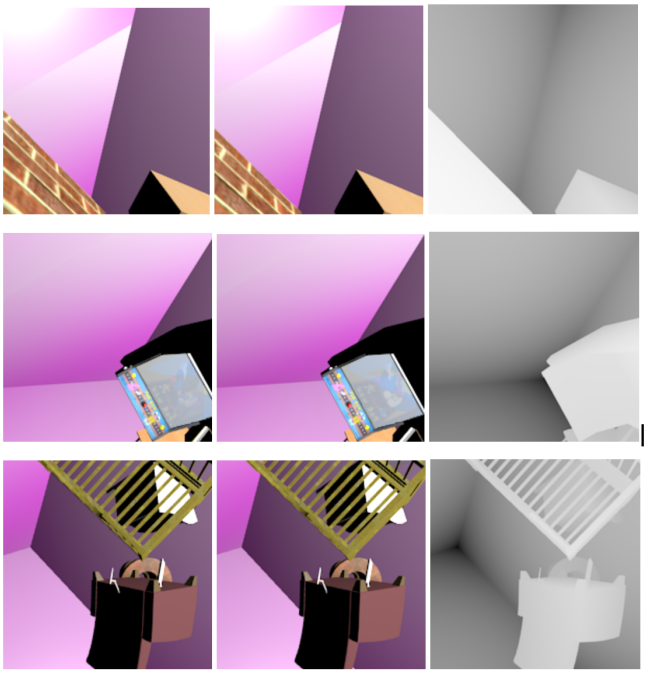

Fig. 2. Depth map estimation

B. Simulated Environment

The agent was placed inside a simulated 3D room environment containing multiple point light sources for illumination. The environment contains various real life room obstacles from the Shapenet Dataset. We measure the performance of our algorithm using a fitness function that determines the path count to destination. We compare the results of this to human results on the same problem, and the observations are mentioned in table in the subsequent chapter. Room structures were created using Blender 3D, an open source 3D modelling software. The overall room environment was then imported as .egg files into Panda3D.

We set up 3 cameras in Panda3D while conducting the experiment.

Camera 0: Overview Primarily for the observer to maintain a view of the overall system.

Camera 1: Left Camera Left component of the stereo vision system

Camera 2: Right Camera Rght component of the stereo vision system

Camera 1 and Camera 2 send buffered streams to the robotic agent.

Camera 0 is not viewed by the agent, and is simply for observational purposes.

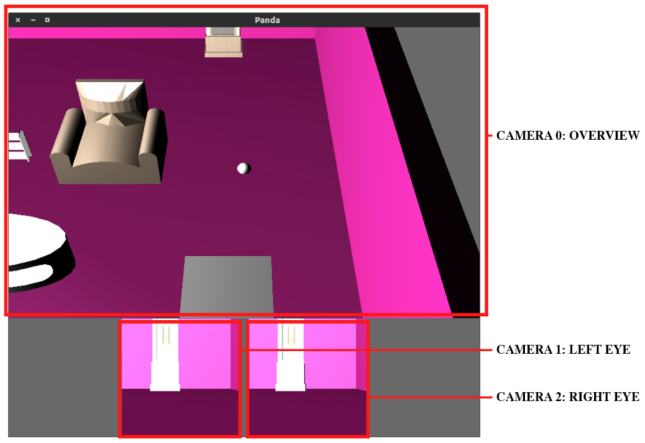

Fig. 3. Simulated Environment

C. Path Planning

The entire 3 dimensional environment is divided into N unit volume cubes, centred at integer coordinates.

A heuristic aided search algorithm is used as the basis for path planning. It is augmented by the depth map discussed in the previous section. The interfacing between the algorithm and the generated depth map will be discussed in the subsequent section.

The following implicit data structures and objects are maintained by the agent controller object:

set_visited A set of previously visited nodes

set_prospects The currently available prospects for immediate motion

stack_active A stack of nodes on the currently active path

to_cube The highest priority cube in the immediate surroundings on the path to destination

Two types of step movement policies were examined for this control policy:

1) 4-connected policy
2) 8-connected policy

Two distance heuristics were compared in the experiment:

1) Manhattan distance
2) Euclidean distance

D. Interfacing depth map with path planner

The entire 3 dimensional environment is divided into virtual unit volume cubes, centred at integer coordinates. There are 8 immediately neighbouring cubes to which the agent can move. The path planner locally generates the next cube towards which it intends to move based on the heuristic.

The agent receives two 200×200 input image streams from the left and right cameras based on which it must plan it's short term path and avoid obstacles. The agent step rotates and faces the heuristic selection. The agent uses the trained CNN to generate a single 200×200 pixel depth map from the stereo image camera stream.

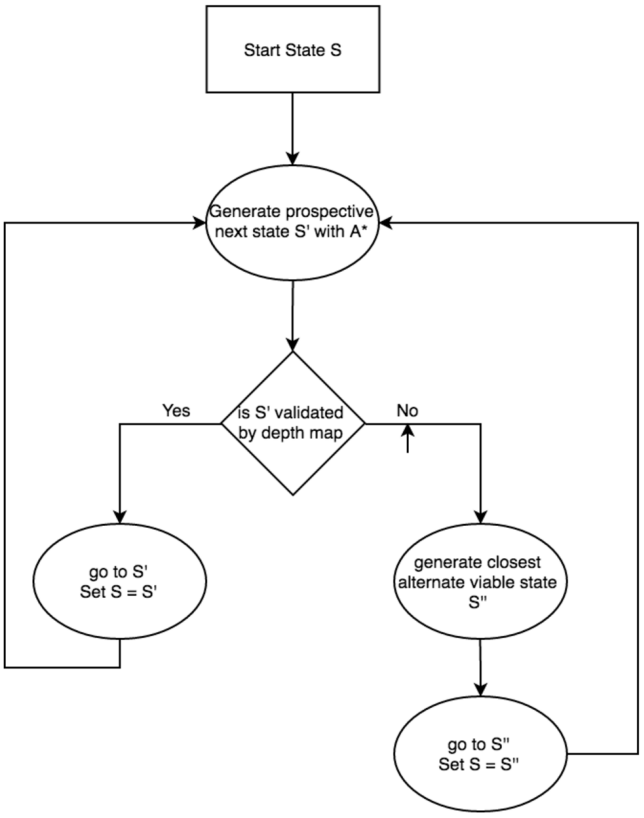

Fig. 4. Interfacing depth map with path planner

This depth map is filtered and cropped to focus on the target window W for the agent, and mean intensity statistics are computed. If the agent evaluates the mean intensity to be above a threshold T, it decides that the heuristic generated next-cube isn't safe for traversal and directs it to present an alternate option. This process continues until the heuristic selects a subsequent neighbouring cube with no detected obstacles.

Algorithm 1 Experiment Workflow

1: Import Blender model, and obstacles from the Shapenet Database
2: Initialize neural network and load CNN model weights
3: Two cubes are selected (src_cube or dest_cube)
4: Ensure there is no collision with src_cube or dest_cube
5: Initialize agent controller with these initial values
6: Interface agent controller with left and right camera stream
7: Start game and log the step counts
8: End log when destination reached

IV. Results

The neural network was able to accurately estimate the depth parity given 2 input stereo images. The validation loss was calculated by comparing the ground truth depth map generated using Blender against the predicted depth map obtained from the Neural Network.

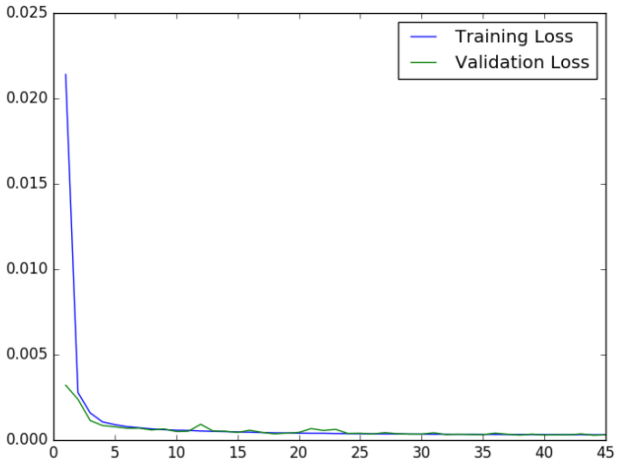

Fig. 5. Training and Validation loss of depth map

Validation loss was determined to be in the order of 10-3 .

The performance of the algorithm with different heuristics is given in Table 1.

TABLE I
PERFORMANCE METRICS

SNo.	Human Baseline	M_{Man}	M_{Euc}	R_{Man}	R_{Euc}
1	179	330	241	1.84	1.35
2	209	472	472	2.26	2.26
3	276	556	520	2.01	1.88
4	269	252	551	0.93	2.05
5	167	401	401	2.40	2.40
6	197	462	361	2.35	1.83
7	123	114	114	0.93	0.93
Median	-	-	-	**2.01**	**1.88**
Average	-	-	-	**1.81**	**1.80**

SNo. : Scenario number, each having different source and destination.
Human Baseline: Step count of a human performing the experiment
$\mathbf{M_{Man}}$: Step count using Manhattan distance as base distance heuristic.
$\mathbf{M_{Euc}}$: Step count using Euclidean distance as base distance heuristic.
$\mathbf{R_{Man}}$: Ratio of Manhattan to Baseline performance (Lower is Better)
$\mathbf{R_{Euc}}$: Ratio of Euclidean to Baseline performance (Lower is Better)

The average relative performance of the Manhattan and Euclidean based heuristic was found to be 1.81 and 1.80 respectively.

The median relative performance of the Manhattan and Euclidean based heuristic was found to be 2.01 and 1.88 respectively.

The average metric is susceptible to outliers in observations and is hence a less reliable metric for evaluation. The median metric is able to circumvent this problem, and is thus a better measure of the relative performances of the heuristics in this situation.

The Euclidean heuristic equaled or outperformed the Manhattan heuristic on 85.71% of the experiments, and is hence a better heuristic for evaluation.

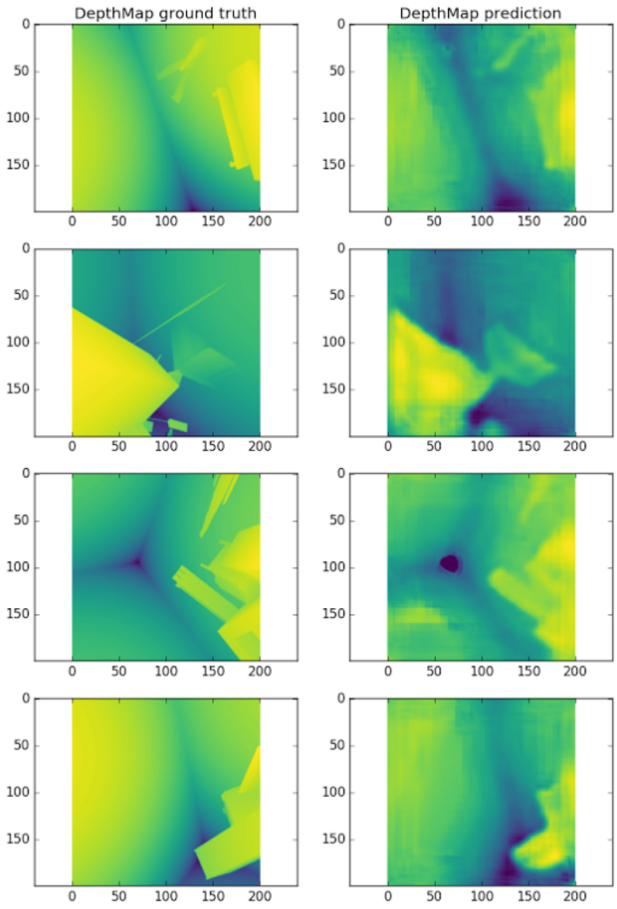

Fig. 6. Performance of depth map on simulated images

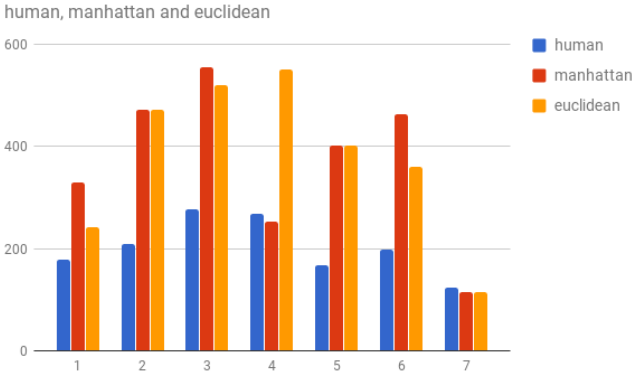

Fig. 7. Performance of heuristics vs humans

V. Future Work

The heuristic could be further improved by adding penalties for changes in direction as currently, a lot of energy of the agent is involved in continuously checking other directions for a marginally better path.

The long term path planning algorithm is not effective with complex maze patterns. The inability of the agent to have awareness about large obstacles like walls leads to inefficient path planning. Simultaneous Localization and Mapping (SLAM) using the depth map can be performed, in order to obtain better relative performance and more awareness about the surroundings.

References

[1] Sariff N. Buniyamin N. Wan Ngah W.A.J. and Mohamad Z. "A Simple Local Path Planning Algorithm for Autonomous Mobile Robots". In: *INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, ENGINEERING and DEVELOPMENT*. June 2011.

[2] NORLIDA BUNIYAMIN NOHAIDDA BINTI SARIFF. "Ant Colony System for Robot Path Planning in Global Static Environment". In: *SELECTED TOPICS in SYSTEM SCIENCE and SIMULATION in ENGINEERING*. 2010.

[3] Fayao Liu, Chunhua Shen, and Guosheng Lin. "Deep Convolutional Neural Fields for Depth Estimation From a Single Image". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[4] G. R. Taylor, A. J. Chosak, and P. C. Brewer. "OVVV: Using Virtual Worlds to Design and Evaluate Surveillance Systems". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. June 2007, pp. 1–8. DOI: 10.1109/CVPR.2007.383518.

[5] Xingchao Peng et al. "Exploring Invariances in Deep Convolutional Neural Networks Using Synthetic Images". In: *CoRR* abs/1412.7122 (2014). URL: http://arxiv.org/abs/1412.7122.

[6] J. J. Lim, H. Pirsiavash, and A. Torralba. "Parsing IKEA Objects: Fine Pose Estimation". In: *2013 IEEE International Conference on Computer Vision*. Dec. 2013, pp. 2992–2999. DOI: 10.1109/ICCV.2013.372.

[7] Alireza Shafaei, James J. Little, and Mark Schmidt. "Play and Learn: Using Video Games to Train Computer Vision Models". In: *CoRR* abs/1608.01745 (2016). URL: http://arxiv.org/abs/1608.01745.

[8] Nikolaus Mayer et al. "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation". In: *CoRR* abs/1512.02134 (2015). URL: http://arxiv.org/abs/1512.02134.

[9] Heiko Hirschmuller. "Stereo Processing by Semiglobal Matching and Mutual Information". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 30.2 (Feb. 2008), pp. 328–341. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1166. URL: http://dx.doi.org/10.1109/TPAMI.2007.1166.

[10] Jure Zbontar and Yann LeCun. "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches". In: *CoRR* abs/1510.05970 (2015). URL: http://arxiv.org/abs/1510.05970.

[11] Elon Rimon, Ishay Kamon, and John F. Canny. "Local and Global Planning in Sensor Based Navigation of Mobile Robots". In: *Robotics Research: The Eighth International Symposium*. Ed. by Yoshiaki Shirai and Shigeo Hirose. London: Springer London, 1998, pp. 112–123. ISBN: 978-1-4471-1580-9. DOI: 10.1007/978-1-4471-1580-9_11. URL: https://doi.org/10.1007/978-1-4471-1580-9_11.

[12] Jakob Ziegler et al. "Automated, Depth Sensor Based Object Detection and Path Planning for Robot-Aided 3D Scanning". In: *Advances in Service and Industrial Robotics: Proceedings of the 26th International Conference on Robotics in Alpe-Adria-Danube Region, RAAD 2017*. Ed. by Carlo Ferraresi and Giuseppe Quaglia. Cham: Springer International Publishing, 2018, pp. 336–343. ISBN: 978-3-319-61276-8. DOI: 10.1007/978-3-319-61276-8_37. URL: https://doi.org/10.1007/978-3-319-61276-8_37.

[13] Angel X. Chang et al. *ShapeNet: An Information-Rich 3D Model Repository*. Tech. rep. arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[14] Louis Foucard. *StereoConvNet*. Mar. 2016. URL: https://github.com/LouisFoucard/StereoConvNet.

[15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). URL: http://arxiv.org/abs/1412.6980.

APPENDIX A
NETWORK ARCHITECTURE

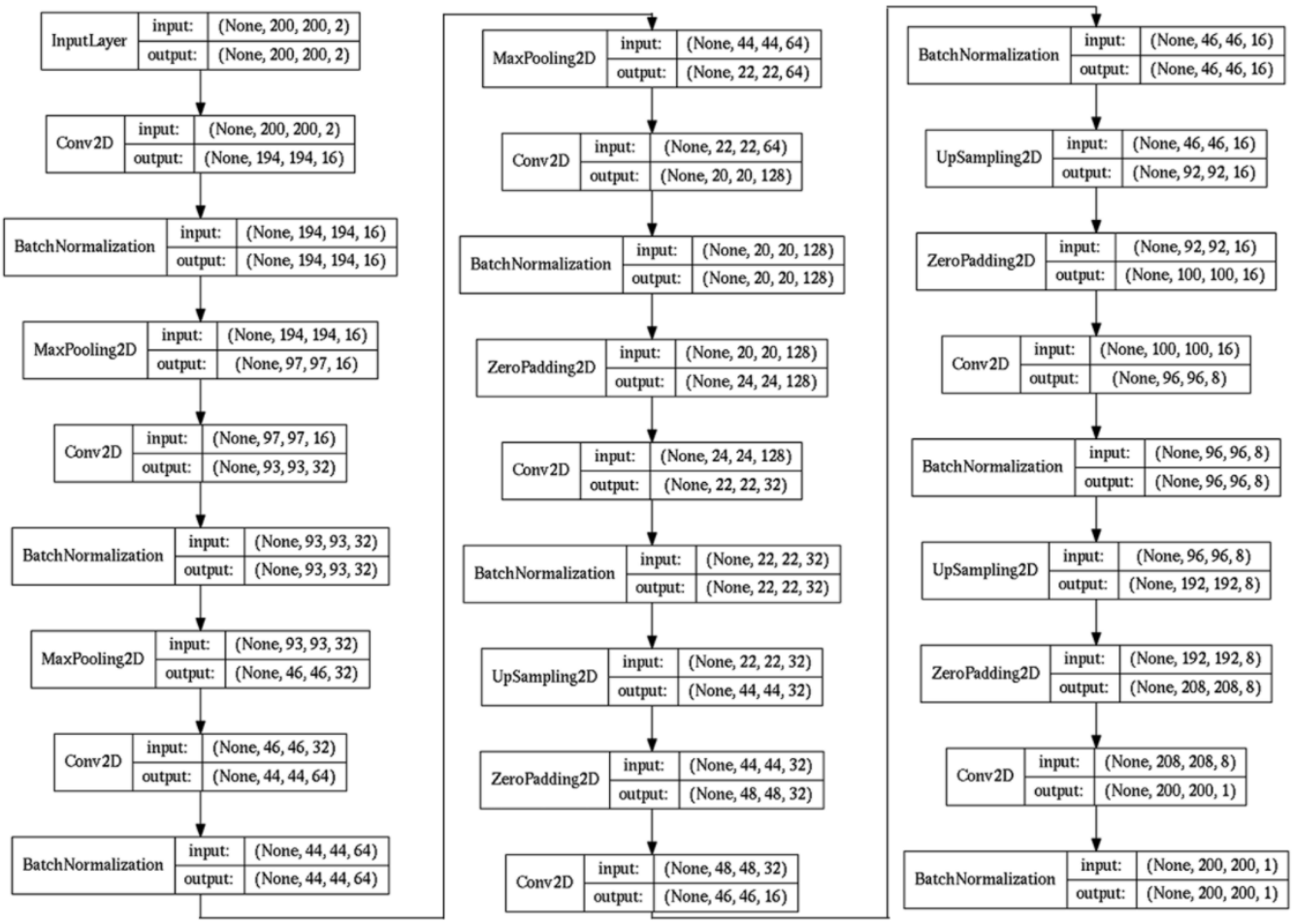

Fig. 8. Network Architecture