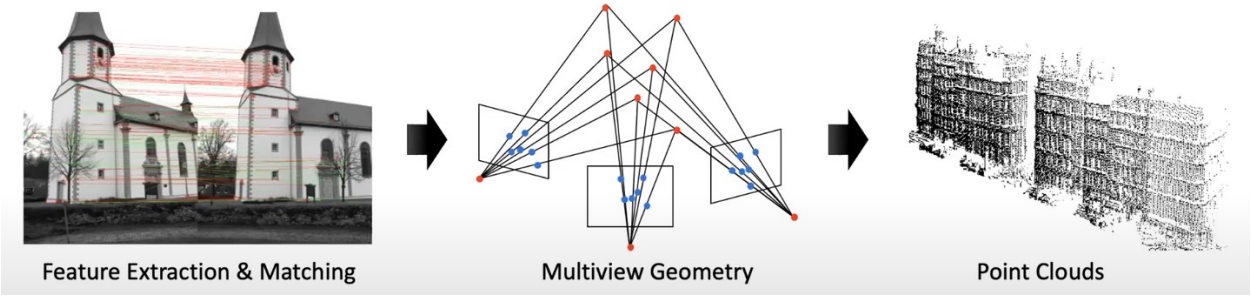
EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2020**

Lecture 18: (Applications of 3D vision (AR/VR))

Scribes: Scott Lim, Matthew Tran

18.1 Detecting Geometric Structures from Images for 3D Parsing

18.1.1 Traditional 3D Reconstruction



The current and best practice in the industry has the above 3D reconstruction mechanism. Starting from local features, we establish point correspondences and use multiple view geometry. Then, we can get point clouds and camera pose.

However, the traditional 3D reconstruction has several issues with the following features:

- Textureless Objects
- Reflection/Transparency
- Repetitive Patterns
- Medium/Large Baseline
- Moving Objects

In particular, it fails in areas where humans are very good, such as dealing with few features and reflections.

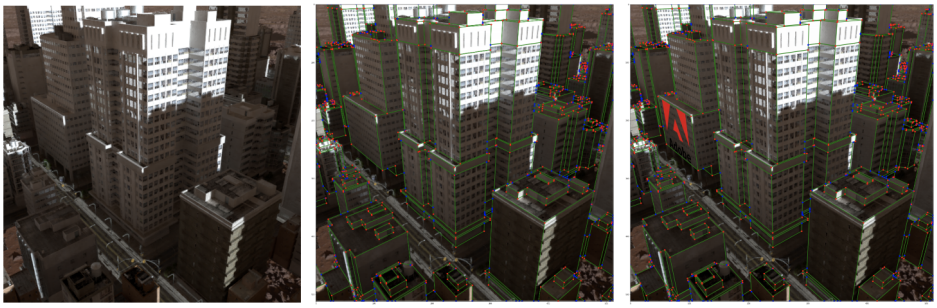
18.1.2 Deep Learning (Data-Driven) Approaches

Thousands of papers suggested numerous deep learning approaches (i.e. pose estimation, voxels, point clouds, 3D bouncing cube, meshes, implicit surfaces, etc.) to replace the traditional 3D reconstruction.

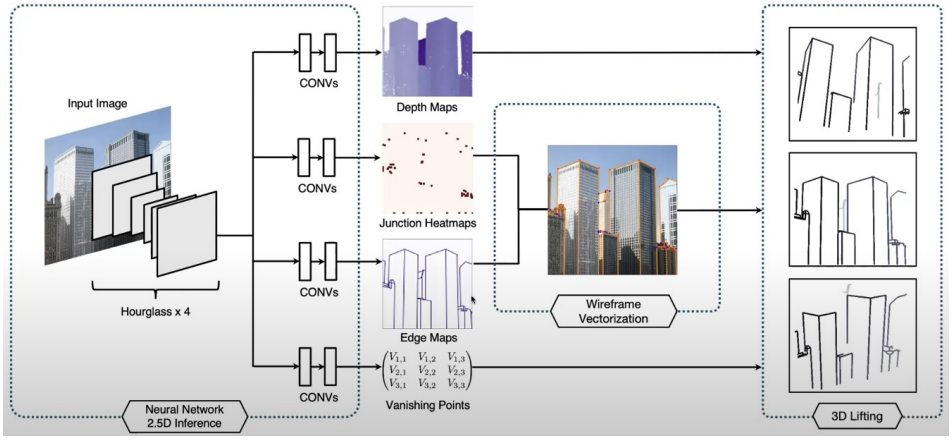
However, all these deep learning methods are just as good as nearest neighbors at 3D reconstruction. They don't actually learn the underlying geometric structures. We live in a structured world full of straight lines, smooth curves, parallelism, orthogonality, and symmetry. We ask how to incorporate this knowledge into our data-driven approach?

18.1.3 Learning to Reconstruct 3D Wireframes from Single Images

Traditionally, the world is represented using point clouds which are not ideal due to high resource requirements. Inspired by CAD, the wireframe approach essentially breaks down the 3D scene into straight lines for the edges. This is particularly useful in AR/VR applications due to their low resource requirements.



Now we ask how to create this wireframe? The following pipeline describes one method.



This approach generates heat maps for the different junction types and the lines to create a wireframe. Then it uses depth maps and vanishing points to lift the wireframe into 3D.

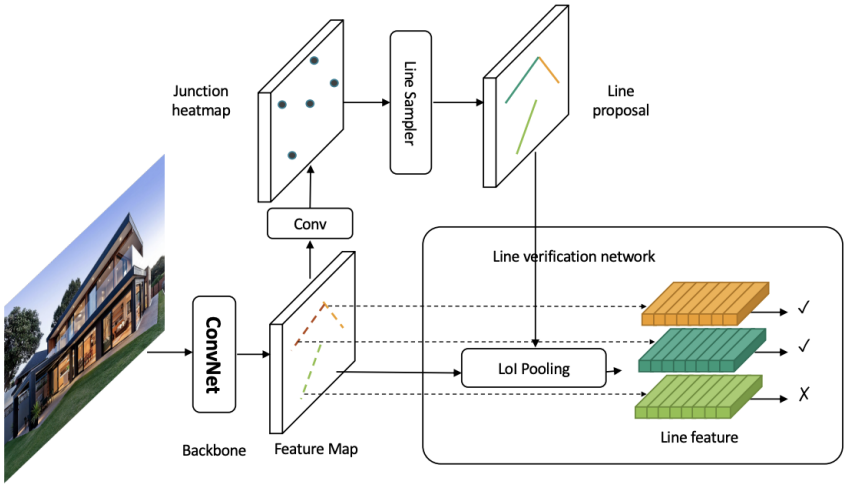
This early wireframe representation approach was purely learning based and had two major issues:

- Predicted vanishing points are not always accurate
- Line detection is challenging, especially when crowded

18.1.4 L-CNN: End-to-End Wireframe Parsing

Existing line detectors usually generate a line heat map and use heuristic search algorithm to make vectorized lines and junctions. However, this is difficult to implement and have unsatisfactory performance.

This algorithm attempts to solve all the problems of existing detectors in an end-to-end trainable neural network. It has the following pipeline.

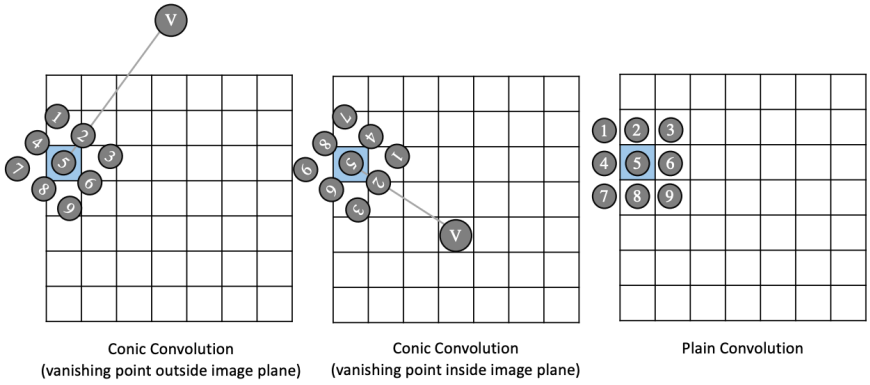


This approach uses a backbone network, junction proposal network, line sampler, and line verification network. Qualitatively, it approaches human level performance and can run in real time.

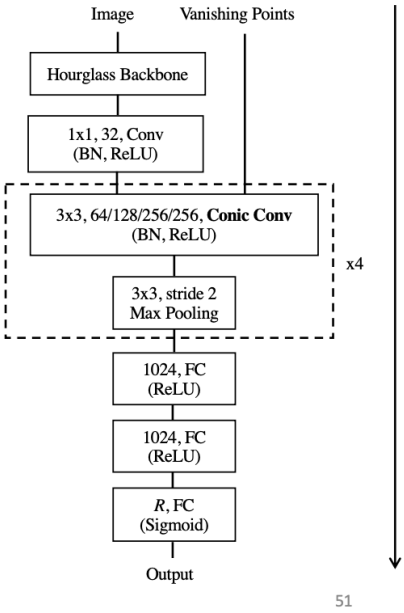
18.1.5 NeurVPS: Neural Vanishing Point Scanner via Conic Convolution

Vanishing points are points that parallel lines in 3D intersect at after projection. This gives lines direction in 3D. Traditional approaches with two-stage algorithms are accurate but not robust to outliers. Neural networks are robust but inaccurate. NeurVPS attempts to get the best of both worlds and be end-to-end trainable with robustness and accuracy.

One key part of this is the conic convolution which are better adapted for detecting vanishing points.



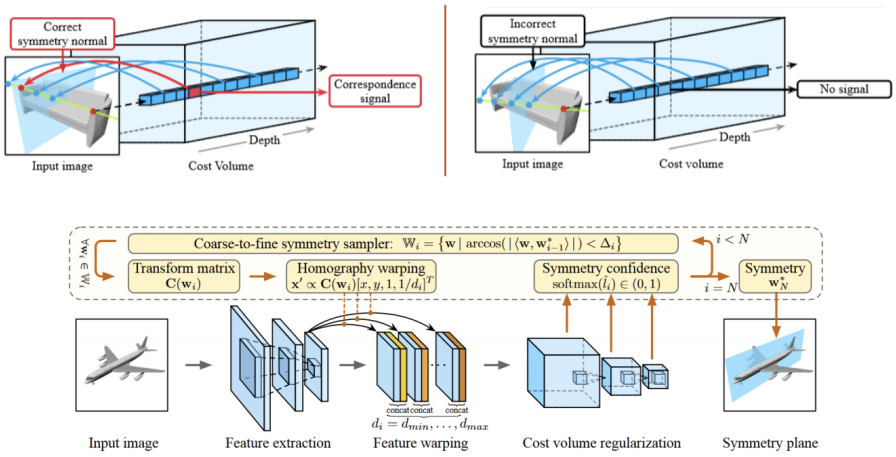
It then applies a network to classify vanishing points.



On a synthetic benchmark, NeurVPS had a median error of 0.1° which approached the floating point limit.

18.1.6 NeRD: Neural 3D Reflection Symmetry Detector

This algorithm exploits reflective symmetry to detect object poses from a single image. It uses reflective symmetry to detect vanishing points.



The performance compared to other traditional methods is exceptionally good.

18.1.7 HoliCity - A City-Scale Data Platform for Learning Holistic 3D Structures

The above four methods were combined to generate a city-scale 3D reconstruction.

18.2 Theory and Applications of VR and Immersive Computing (by Dr. Allen Yang)

The concepts of augmented reality (AR) and virtual reality (VR) existed for more than 60 years, but the computer vision industry really adopted AR/VR just recently. Especially during the past year, the pandemic situation really accelerated AR/VR's development.

- Virtual Reality: A computer technology that uses head mounted displays, sometimes in combination with other sensory devices, to generate realistic images, sounds, and other sensations (touch, smell, motion, etc.) that simulate a user's physical presence in a virtual environment
- Augmented Reality: A computer tehcnology that augments a physical, real-world environment directly or through its indirect view computer-generated sensory information, including graphics, video, and sound. AR may alter a user's view of reality, and may also enhance one's perception of reality.

18.2.1 AR/VR Form Factors

There are 4 main form factors when it comes to implementing AR/VR applications:

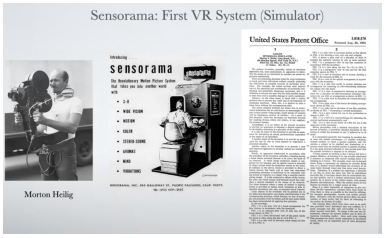
- Simulated Environment - recreating parts of the environment and simulating the rest (i.e. flight simulator, sport simulator, VR arcade room/amusement ride, etc.)
- Tethered to PC - usually involves a headset that is tethered to a static computer (i.e. Oculus Kickstarter Kit Display, HTC Vive Pro with Lighthouse Bases, Virtuix Omni Treadmill, etc.)
- All-in-One - generally an untethered headset (i.e. HoloLens, Oculus Quest, Google Glass, etc.)
- Smartphone - uses a smartphone, optimized for low cost (i.e. Google Cardboard, Google Daydream, AR Kit/AR Core, etc.)

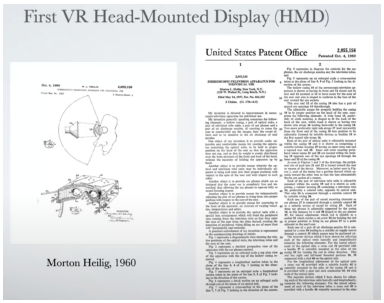
The different approaches balance different properties.

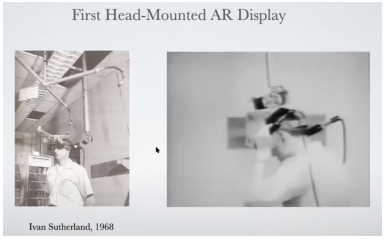
Form Factor	Simulated Environment	Tethered to PC	All-in-One	Smartphone
Cost	Very High	Medium	Low	Very Low
Fidelity	Best	High	Medium	Low
Motion-Tracking	High	High	Medium	Low
User Input	Very Realistic	High	Low to Medium	Very Low

18.2.2 History of AR/VR

AR/VR has taken an interesting path throughout history. One big application is in defense where it's used to train paratroopers and astronauts and even used in pilot's helmets.







18.2.3 How to Trick the Senses?

In order to create an immersive experience, the senses need to be tricked. Here are the current technology gaps on how we approach different sides of the problem.

	Near-Eye Display	Binaural Audio	Haptic Controller	Natural Language Processing	3D Localization
What works	Matured and inexpensive smartphone LCD screen technologies	Head-Related Transfer Function (HRTF) simulates spatial 3D sound using two channels	Reliable tracking using infrared patterns and outside-in tracking algorithms	Successful commercialization of voice-based AI assistant solutions	Low-power depth cameras make tracking more reliable in low-light, low-feature environments
What hasn't work	• Very bulky in volume • Narrow field-of-view • Screen-door effect • High latency	• Personalization of Binaural Head-Related Transfer Function (HRTF) • High-fidelity playback vs spatial resolution	• Localization beyond line-of-sight • Natural gesture recognition • Support for typing in 3D	• Challenges to answer daily questions • Scenarios speaking is not appropriate • GPT-3 a breakthrough?	• Indoor/Outdoor • mm Accuracy • Large-scale re-localization (loop closure)

18.2.4 Self-Driving Tech

The autonomous driving industry is very costly, especially going from level 3 to 4 self-driving. Also, people generally lack trust of autonomous systems even though human's manual driving causes drastically more fatal crashes.

ROAR is an open-source project designed for low cost, high compatibility, high performance, and high safety to train self-driving cars. It's designed in such a way that it can be directly interchanged between simulation and actual hardware.