Scene understanding to aid in limited-bandwidth remote driving

Project Proposal - CS 231

Hayk Martirosyan
Brady Jon Quist

January 2015

1 Introduction

Progress on autonomous driving promises to make our transportation infrastructure more efficient. In particular, huge efficiency gains could be made in the transportation of goods by using vehicles without humans in them. These vehicles would have a fraction of the components of passenger vehicles, and could be designed without crumple zones, seats, steering wheels, infotainment, airbags, door and window controls. Instead, they could consist of a simple electric drive-train, a sensor cluster, and a cargo bay. In this type of logistics infrastructure, the hardware is lighter and cheaper, there are no human drivers, and the system can be intelligently automated and running 24/7, 365.

However, fully autonomous vehicles are not yet ready to fulfill this vision. Existing solutions operate well in pre-mapped regions and good conditions, but break down in adverse weather and in the presence of unexpected events or unknown terrain. One alternative which acts as a gateway to this ultimate goal is to use remotely-driven vehicles. The hardware remains the same, but the vehicle is operated by a human in a driving simulator at a remote location. This approach retains many of the efficiency benefits of the autonomous version while keeping the flexibility and judgement of human drivers.

One of the key challenges with this approach is maintaining a robust communication stream that enables remote driving. In order to reliably operate a remote vehicle, the driver must have a low-latency, high-fidelity, and wide-coverage stream of audio and video. Existing infrastructure has good coverage in many areas, but bandwidth can be very limited. To accommodate this, we propose that many parts of a typical stream have no importance to driving capability. For example, the driver does not care about details of the sky, trees, or objects very far away. A nearby object moving quickly across the view, however, is extremely important.

We propose to rank the components of a vehicle's stereo vision stream in terms of their importance to remote driving capability, with the goal of enabling limited-bandwidth transmission of the stream.

2 Technical

As this project focuses on the Computer Vision component of this problem, we will primarily use stereo camera simulations and real world data that are publicly available and are well calibrated[1][2][3]. This will alleviate some of the potential difficulties associated with the setup (e.g., alignment, syncing the image frames from two separate cameras, etc.) and allow us to get further in the computer vision aspects of our proposal.

There are several ways in which the bandwidth could be reduced depending on what we feel is important.

One approach is to use a 3D image to help the algorithm determine which regions of the image are not important. In this situation a 3D scene of what appears before the driver would be estimated using a stereo view camera. With this 3D information, we can determine which pixels are less important (based on distance to the driver or distance to the drivers path). Furthermore, if we compare the 3D scene between frames, we can potentially determine which objects are moving (due to their own motion and not the vehicles) and send the pixels corresponding to that motion at lower compression levels or perhaps even higher frame rates.

An alternative method would rely on image segmentation and classification to determine which portions of the image should be compressed more than others. In this scenario, video from a single camera would be used to detect common objects observed on a road. These objects might include the sky, road, median, other vehicles, trees, buildings, etc. Based on the object classification, we could determine the compression level and frame rate that different portions of the video need to be transmitted at. For example, if we know that a significant portion of the image is the sky, then we can send that infrequently and at a very high compression level without compromising the safety of the remotely controlled vehicle. Conversely, if we know that there is a car directly in front of the vehicle, that information should be sent at a lower compression level and at a much higher frame rate to ensure safe operation of the vehicle.

Finally, we could potentially combine the 3D scene information with the image segmentation and classification to better determine which objects that we can compress.

If time permits, we would like to implement the image compression as a means of testing how useful the 3D point estimation and image segmentation/classification are. It would be ideal for this whole process to work in real time (whether in our own setup or at the real-time speeds of the datasets we obtained online), but recognize that to do so may be beyond the scope of this project.

3 Milestones

1. 3D point estimation (Primarily Hayk, but Brady involved)
    - Determine the 3D points location of images (Preliminary results by 2/20/2015)
    - Detect how far away the objects are and the angle to target to determine compression level (Completed by the final project deadline)
2. Image segmentation/classify (Primarily Brady, but Hayk involved)
    - Segment the image and classify the segments into common types of objects (sky, road, clouds, trees, median, cars) (Preliminary results, on a subset of objects, by 2/20/2015)
    - Determine based on object classification, location, motion what the compression level should be (Completed by the final project deadline)
3. Video compression (time permitting)
    - Compress the video based on the 3D scene detection and image segmentation to test the performance and benefits of the the two preceding items.
4. Make the whole process work in real time (moonshot - time permitting)
    - Perform 3D scene detection, image segmentation/classification, and compression work in real time so that a vehicle could be driven remotely.

References

[1] Albert S. Huang, Matthew Antone, Edwin Olson, David Moore, Luke Fletcher, Seth Teller, and John Leonard. A high-rate, heterogeneous data set from the darpa urban challenge, 2010.

[2] Tobi Vaudrey, Clemens Rabe, Reinhard Klette, and James Milburn. Differences between stereo and motion behavior on synthetic and real-world stereo sequences. In *23rd International Conference of Image and Vision Computing New Zealand (IVCNZ '08)*, pages 1–6, 2008.

[3] Andreas Wedel, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers. Efficient dense scene flow from sparse or dense stereo data. In *10th European Conference on Computer Vision (ECCV '08)*, pages 739–751, Berlin, Heidelberg, 2008. Springer-Verlag.