--

Our algorithm

* Speed (hardware improvements)
  + Calibration

Given more time, there are many avenues for improvement of our solution. So, the amount of time and resources we assume will be available shape our future plans. In order to focus this scope, we stratify our future goals into three sections: our current software algorithm, our overall current solution, and potential future solutions that we can pursue.

Our most direct focus within these levels of hierarchy is our current software algorithm. In it’s current form it functions with basic proven timeliness and accuracy – our two primary metrics. As these are our focus in terms of performance, we present here the most important and eminent goals of improvement to these two topics.

The goal of our current algorithm is to work satisfactorily on an Intel embedded system, and so processing time is paramount to a useful real-time algorithm. Our algorithm is currently working at the minimum speed to be considered satisfactorily real-time. Our first next step would be to begin trialing our solution in hardware. We currently have our Altera Cyclone IV FPGA fully set-up and communicating with our Atom processor using a custom kernel driver. Our next step in this area would be to implement an algorithmic step in hardware description language, synthesize, test, and include it in our processing loop.

Pareto chart of performance here

Looking at our current performance metrics, it is clear that the longest processing time is incurred by the calibration step. However, this step is currently not utilized in the final snapshot of our algorithm given its very large negative effect on performance. Implementing it in hardware can only be technically considered an improvement in quality then. In order to actually improve processing time, the algorithmic step that logically follows next in line to implement in hardware is the homography transformation.

With this in mind, we propose that our next steps in this direction would be to implement both the calibration and homography transformations. This would deliver an incremental improvement in measurement accuracy, as well as a clear improvement to our bottom line processing time performance. Given relevant research into hardware implementations of calibration and plane-to-plane transformations, we see it reasonable that solutions could achieve approximate performance of YY and ZZ respectively. We believe that we have the current skillset and resources to accomplish this. Both of the algorithms are not very complex in nature. They are both composed of lots of repetitive calculation. So we would expect the timeline for this development to be about several weeks. This would allow extra time for debugging the hardware PCIe link between the FPGA and Intel Atom incase any problems are incurred.

-----

Our software algorithm is very integrated in nature. In order to improve it, we believe that it is important to make it more robust overall. After working with and testing our original algorithm, we think it would be helpful to develop a deeper model of the road environment. Many roads, even fairly straight interstates, have at least some minor curve after a given distance. This variably limits our range and increases the dynamism of the environment we are trying to understand.

We are aware of various algorithms, like that of Nieto [cite], that track the curve in the road, delivering a straightened version of it. Developing this kind of model has manifold improvements: it allows us to better understand the edge of the road, thereby giving limits to the scope of our algorithm and reducing false positives from non-road objects; it increases the range of our solution; and overall it provides us a more consistent, straight model of cars moving behind us. This would give an improvement to accuracy of detection as well as accuracy in world model itself. In the beginning we discounted the relevance of lane models. Having implemented an algorithm from start to finish we can appreciate the robustness it would provide to our solution.

This kind of lane model system is proposed in Nieto [,.,.]. It is fairly straightforward and similar to the development we have been doing. It doesn’t require any large change to any of our existing resources whether it be our codebase or our skills. We would just need time to program and test the solution. We believe it to be something we can implement within a couple of weeks with the team that we already have.

With our current solution, development of a real prototype consisted of utilization of different vehicles: a truck, motorcycle, and bicycle. We know the capabilities of our physical hardware system, but realized the lack of hard skill set in the mechanical arena left us short of a concrete incarnation of our prototype. Given the technology to mount a camera to a vehicle is mostly trivial, we did not worry about perfectly reconciling these issues and prioritized a working software algorithm – or the core of what we do. Given more resources one of our next priorities would be to develop a more permanent prototype. Ideally, this would involve financial help so that a permanent vehicle could be procured – preferably a cheap, small-size, small-bore, motorcycle. This helps us to avoid the transience of personal vehicle ownership. We would still need to collaborate with a mechanical engineering team to develop a rigorous, vibration tolerant mounting system for a motorcycle. We could also use this collaboration to aid us in designing a more robust gimbal system. The smaller engine would hopefully prove an easier task to develop a dampening mount for, in stark contrast to the 649cc, single-cylinder motorcycle we have access to now.

The task of developing a permanent physical prototype would be of a longer duration. We would need more resources in terms of money and engineering skillset. There would be a substantial amount of work to be done: modify a motorcycle; design a mount and gimbal. This would be something we would expect to finish in a minimum period of several more months. The gain would be great. We would have a unified testing system and better prototype to exemplify our algorithm to the world.

These aforementioned changes would be our direct and immediate next steps. It is also useful to looking further into future. Comparing our current solution with that of our original proposal, it is clear that there is also more functionality we can implement. Looking in the further into the future we would develop a solution that is more robust in regards to the environment. This means adapting our algorithm for rain or bad weather such as low sunlight. It also means researching and building wholly new algorithmic solutions for nighttime situations.

We have done some preliminary research into night situations. There is a great deal of research that shows it is fairly simple to detect objects based on their headlights. We believe this would be fairly straightforward to implement but it would require modifying our algorithmic worldview. Said simpler, it would require some new fundamentals to our new algorithm. At the worst case, it may require a whole new algorithm altogether. We have done less research on working solutions in poor visibility situations – like heavy rain, or low sunlight. These appear to remain something that modern advanced driver assistance systems struggle against. We have some small algorithmic proposals in mind to make our solution more robust, but no overarching solutions to these longstanding problems.