Occupancy Grid Computation

March 18, 2014

Contents

1 V-disparity Map Computation

Project all pixels based on their disparity value and v-coordinate value. Then we can set a threshold value to eliminate some noise pixels which accumulate results are not big enough.

Fig. 1 shows the input disparity map. Fig. 2 shows the result v-disparity map result.

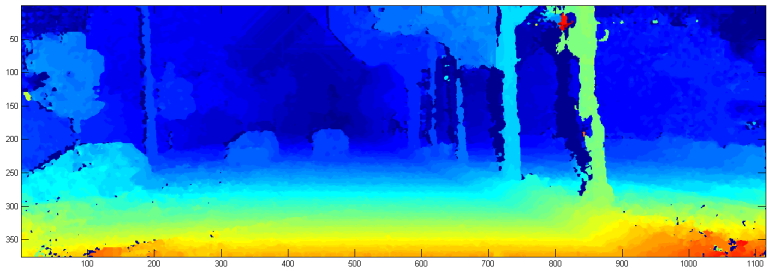


Figure 1: Disparity Map

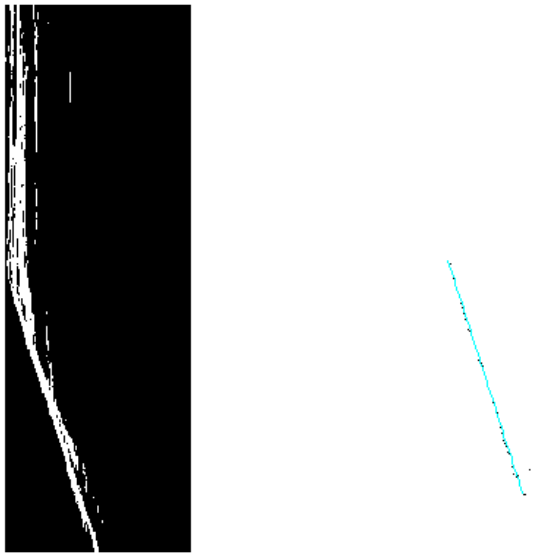
Figure 2: V-disparity Map

2 Ground Computation

The straight line formed by the lowest points for each columns represent ground information between disparity value and v-coordinate. We need to extract the lowest points for each columns and then use curve fitting algorithm to find the equation for that straight line.

In this program, we use MATLAB routine 'robustfit' to achieve this goal. Information about the algorithm used for 'robustfit' can be found here.

Fig. 3 shows the fitting result(blue line).



(a) V-disparity Map (b) Ground

Figure 3: Ground Computation

3 Separate Ground and Obstacle

After we get the ground information, we can use that to separate ground with obstacles. The idea is that if one pixel's v-coordinate value is not larger than the v-coordinate value of ground point which share the same disparity value with our target pixel, then that point belong to ground pixel, otherwise, that point belongs to obstacle pixel.

Fig. 4 shows ground disparity map result. Fig. 5 shows obstacle disparity map result.

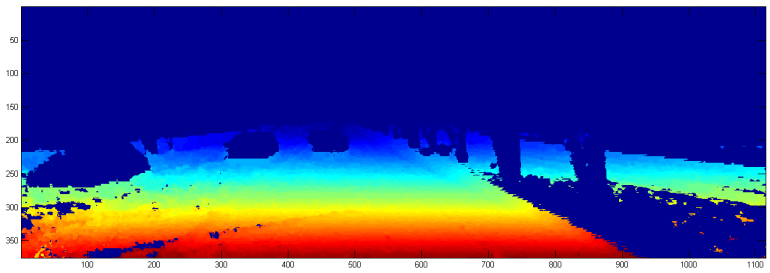


Figure 4: Ground Disparity Map

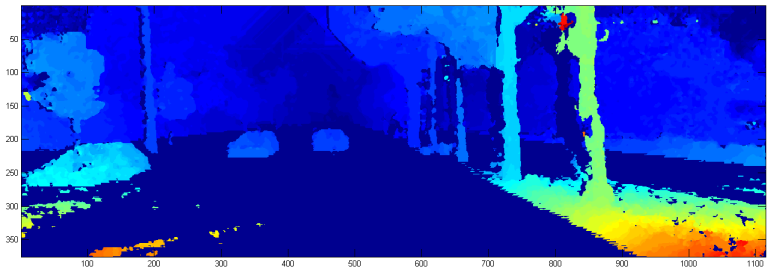


Figure 5: Obstacle Disparity Map

4 Obstacle U-disparity Map Computation

Like V-disparity Map computation, we just project all pixels in Obstacle Disparity Map based on their disparity value and u-coordinate. Also we can set a threshold value to eliminate some noise pixels which accumulate result are not big enough.

Fig. 6 shows U-disparity map result.

Figure 6: Obstacle U-disparity Map

5 Occupancy Grid Computation

We use the method in [1] to compute Occupancy Grid.

We only need to care about the occupancy probability between ground plane and the other plane has a fixed height, so we need to define this plane with fixed height. In here we use another line equation to represent that plane.

Fig. 7 shows occupancy grid result.

Some parameter:

$N_P(U)$: Number of pixels

$N_O(U)$: Number of obstacle pixels

$N_V(U)$: Number of visible pixels

$P(V_U = 1) = \dfrac{N_V(U)}{N_P(U)}$: Visibility of a cell

$P(C_U = 1) = 1 - e^{\frac{r_O(U)}{r_O}}, r_O(U) = \dfrac{N_O(U)}{N_V(U)}$: Confidence of observation

$P(O_U) = P(V_U = 1)P(C_U = 1)(1 - P_{FP}) + P(V_U = 1)(1 - P(C_U = 1))P_{FN} + \dfrac{(1 - P(V_U = 1))}{2}$:

Occupancy probabilityP_{FN} and P_{FP} are false negative probability and false positive probability.

Figure 7: Obstacle U-disparity Map

6 Source Code Commitment

We have provided a MATLAB code for computing occupancy grid based on disparity map, and the source code can be found at `Src/Occupancy Grid Computation - MATLAB`.

Input files are .txt files which are just the raw data output getting from disparity map computation program based on OpenCV.

References

[1] M. Perrollaz, J.-D. Yoder, A. Spalanzani, and C. Laugier, "Using the disparity space to compute occupancy grids from stereo-vision," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 2721–2726, IEEE, 2010.