A FPGA-Based Real-Time Stereo Vision System with Improved SGM Path Cost Propagation Method

Chia-Pin Tseng, Boyang Wang
Advisor: Dr. Chang Choo

5/2015
Objective

* Real-time FPGA-based stereo vision system
* Semi-Global Matching
* Improved design
Survey

* Stereo Vision on robotic vision and autonomous vehicle [1]
* SGM vs local matching and global matching [2]
* FPGA vs computer system [3]
Design Flow

1. Literature survey
2. Software implementation (C/C++)
3. Simulation and evaluation
4. Hardware implementation (Verilog)
5. Simulation and evaluation
6. Simulation and evaluation
7. Implement on FPGA
8. Real life experiment
9. Analysis
Description of Algorithm (Census Transform)

- Census Transform
- Adder tree for Hamming distance
Description of Algorithm (SGM)

\[ L_r(p, d) = C(p, d) + \min( \]
\[ L_r(p-r, d), \]
\[ L_r(p-r, d-1) + P_1, \]
\[ L_r(p-r, d+1) + P_1, \]
\[ \min_i L_r(p-r, i) + P_2) - \min_k L_r(p-r, k) \]

\[ S(p, d) = \sum_r L_r(p, d) \]
\[ \text{disparity}(p) = \arg\min_d (S(p, d)) \]
Description of Algorithm (SGM)

\[ L_r(p, d) = C(p, d) + \min( \]
\[ L_r(p - r, d), \]
\[ L_r(p - r, d - 1) + P_1, \]
\[ L_r(p - r, d + 1) + P_1, \]
\[ \min_i L_r(p - r, i) + P_2 - \min_k L_r(p - r, k) \]

* Example: \( L_r(p - r, d) \)
along 180° path,
green one is the
previous pixel
(P1 < P2)

\[ S(p, d) = \sum_r L_r(p, d) \]

\[ \text{disparity}(p) = \arg \min_d (S(p, d)) \]
Description of Algorithm (SGM)

\[ L_r(p, d) = C(p, d) + \min( \]
\[ L_r(p-r, d), \]
\[ L_r(p-r, d-1) + P_1, \]
\[ L_r(p-r, d+1) + P_1, \]
\[ \min_i L_r(p-r, i) + P_2 - \min_k L_r(p-r, k) \]
\[ S(p, d) = \sum_r L_r(p, d) \]
\[ \text{disparity}(p) = \arg \min_d (S(p, d)) \]

* Sum the path costs in all directions \( r \)
* Search the disparity with the minimal cost for each image pixel \( p \)
\[ L_r(p, d) = C(p, d) + \min( \]
\[ L_r(p-r, d), \]
\[ L_r(p-r, d-1) + P_2, \]
\[ L_r(p-r, d+1) + P_2, \]
\[ \min_i L_r(p-r, i) + P_2) - \min_k L_r(p-r, k) \]
\]

disparity = arg min \left( \sum_d L_r(p, d) \right)
Branch SGM

\[ L'_r (p, d) = C(p, d) + \min( \]
\[ L_r (p - r, d), \]
\[ L_r (p - r, d - 1) + P_1, \]
\[ L_r (p - r, d + 1) + P_1, \]
\[ \min_i L_r (p - r, i) + P_2 ) - \min_k L_r (p - r, k) \]
\[ \text{disparity} = \arg \min_d \left( \sum_r L'_r (p, d) \right) \]

\[ L_r (p, d) = \alpha L'_r (p, d) + (1 - \alpha) \max(L'_r (p, d), L'_{r+1} (p, d), L'_{r-1} (p, d)) \]

Share the neighboring maximum cost

Only few changes
Branch SGM

\[ L_r(p, d) = \alpha L'_r(p, d) + (1 - \alpha) \max(L'_r(p, d), L'_{r+1}(p, d), L'_{r-1}(p, d)) \]
Branch SGM

\[ L_r(p, d) = \alpha L'_r(p, d) + (1 - \alpha) \max(L'_r(p, d), L'_{r+1}(p, d), L'_{r-1}(p, d)) \]
Let’s try
Let’s try (cont’d)

SGM

disparity = db

disparity = df
Let’s try (cont’d)

Branch SGM ($\alpha=0.8$)

disparity = db

disparity = df
Let's try (cont'd)

\[ \text{disparity} = \text{db} \quad (\alpha = 0.8) \]

Branches will try to cover strong features
disparity = db (α=0.5)
Branches will try to cover strong features
disparity=db ($\alpha=0.2$)

Branches will try to cover strong features
What’s the improvement?

To see the improvement, we can see the final cost aggregation result $S(p,d)$.

Branch SGM provides very clean $S(p,d)$
Branch SGM vs SGM

SGM

(Varance=0.001)

Branch SGM
What’s the improvement? (cont’d)

Under variance equals to 0.0005, the average error reduces 33%
What’s the improvement?(cont’d)

Under variance equals to 0.0005, the average error reduces 21%
Hardware requirement
Branch SGM vs SGM

<table>
<thead>
<tr>
<th></th>
<th>register</th>
<th>LUTs</th>
<th>Block RAM (18Kb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGM</td>
<td>6519</td>
<td>10625</td>
<td>130</td>
</tr>
<tr>
<td>Branch SGM</td>
<td>6907</td>
<td>11748</td>
<td>130</td>
</tr>
</tbody>
</table>

Our branch SGM uses 6% more slices and 10% more LUTs.
Hardware requirement
Branch SGM vs SGM

Branch SGM is almost as fast as SGM
Conclusion and Future Work

- Our branch SGM reduced up to 30% average errors under noisy input
- The design required 10% more registers and LUTs
- Future: Implement accurate field (entropy) to guide branches to collect more features information
Reference

