Analysis of Color Space and Similarity Measure Impact on Stereo Block Matching

Nelson Yen-Chung Chang, Yu-Cheng Tseng, and Tian Sheuan Chang
Department of Electronic Engineering
National Chiao-Tung University
Hsinchu, Taiwan
e-mail: {ycchang, tyucheng, tschang}@twins.ee.nctu.edu.tw

Abstract—The impact of color space and similarity measure on complexity, speed, and performance of stereo matching is especially important to applications adopting stereo vision. This work analyzed the complexity of several most commonly considered color space and similarity measure. In addition, the execution speed and performance of color space and similarity measure combination are also compared on the same basis. The comparison result suggests that the Y-only rank provides the best combination under speed and performance trade-off.

I. INTRODUCTION

Stereo vision estimates the depth of a scene from two scene images through stereo matching and triangulation. It has been widely adopted in many applications such as robotics[1], autonomous vehicle[2], multi-view coding[3], and intelligent surveillance [4]. Most of these applications have real-time constraint as well as depth estimation performance requirement. The real-time constraint varies from application to application, but usually requires the depth information to be generated above a given frame rate. The depth estimation performance requirement asks for dense and reliable depth information.

To satisfy the constraint and requirement, stereo block matching methods are usually chosen. The performance of stereo block matching methods is affected by factors including color space, similarity measure, and block size. One of the most important issues is to know the impact of these factors on performance and complexity. Szeliski and Scharstein [5] identified the performance impact due to different block size for block matching methods. Their work also briefly compared the performance and execution time of different matching methods. Hua [6] compared the impact of using different color space on depth estimation performance. However, their testing stereo image pair is less complicated than the scene encountered by a real application. Corke [7] compared the performance of different similarity measures. They also proposed two non-parametric similarity measures called rank and census. Hirschmuller and Scharstein[8] later compared the performance of different similarity under different lighting changes. Although these works had compared the performance, their comparison results are evaluated by different metrics on different stereo image pairs. Hence, it is difficult to determine the impact of different factors from previous works due to the inconsistency of evaluation metrics and stereo image pairs.

Despite the lack of a consistent performance comparison, most of the mentioned works did not address the complexity and execution time in detail. Although some works [9][10] have conducted analysis on execution time on some individual methods, their results were obtained based on different machines and settings. Besides, the impact of different color space and similarity measure on the complexity and execution time are also absent.

Motivated by the absence of a complete and consistent comparison, this work contributes on the comparison and analysis of the impact due to different factors on complexity, execution time, and performance based on the same evaluation settings. To evaluate the complexity and execution time, we optimized the implementation to give a more accurate profile. To evaluate the performance, the most commonly used Middleburry evaluation scheme [5] is adopted to give quantitative result. The comparison result suggests the most suitable color space and similarity measure combination depending on the need for speed, performance, or trade-off on both.

The rest of the paper is organized as follow. Section II briefly introduces our stereo block matching implementation used in this work. Section III presents the factors considered in this work and their corresponding computation complexity. Section IV compares the execution time and performance impact due to different factors. Finally, a brief conclusion is given in Section V.

II. STEREO BLOCK MATCHING

Stereo matching determines the displacement between each corresponding pair of pixels in two camera images. The displacement is referred as disparity and the process is referred as disparity estimation. The set of disparity of all the pixels in an image is termed as disparity map or disparity image. The disparity is inversely proportional to depth. Further detail of stereo matching can be found in [11].
Once all the individual matching costs are computed, the aggregated matching cost is computed from the individual matching costs. For each pixel position, the aggregated block matching cost is computed from the individual matching costs. For each pixel, transform from color space into designated color space representation. For each pixel, compute individual matching cost at each candidate disparity and store into a matching cost table. For each pixel position, compute the block matching cost at each candidate disparity from the individual matching costs. For each pixel position, select the candidate disparity with the minimal aggregated matching cost as disparity. The disparity error rate is averaged from the overall error rate of all stereo image pairs.

In addition to the matching cost reuse, our implementation optimization techniques also include index computation minimization, dynamic allocation elimination, local buffering, and function inlining. In case of floating point values, we shifted them into fixed point values to increase computation speed. Finally, the compiler optimization is set to O2 to maximize speed.

### III. FACTORS: COLOR SPACE AND SIMILARITY MEASURE

#### A. Color Space

Color space affects the matching performance and computation complexity due to its model definition and characteristics. TABLE I lists the color spaces considered in this work. The corresponding computation complexity to transform from RGB color space into other color spaces is listed in TABLE II. From TABLE II, Lab color space transformation requires the highest computation. Lab color conversion also involves division. One way to reduce the color transform computation is to use only single color component, such as Y-only and L-only. To compute the total matching cost of a pixel, the matching costs from each individual color component are summed together except Lab color space.

#### B. Similarity Measure

TABLE III lists the definition of the similarity measures. In which, $I$ represents color value; the subscripts $curr$ and $cand$ represents the current and the candidate image, which can be left and right images or vice versa. $B$ represents the block size for matching cost aggregation and $R$ represents the window size used in rank and census transform. The block size $B$ and $R$ are 9x9 and 5x5 pixels respectively in our implementation. \(u, v\) are the coordinates of the current pixel; the \(x, y\) and \(i, j\) are the coordinates of the pixels within $B$ and $R$ respectively. The complexity of different similarity measures is listed in TABLE IV. Note that the complexity in the table is derived for one color component. For color space with three components, the complexity should be tripled.

### IV. SPEED AND PERFORMANCE COMPARISON

To evaluate speed and performance, the implementations are run on an Intel Core2 Duo 2.99GHz machine with 4 GB memory. The performance is evaluated using the average execution time and the overall disparity error rate. The average execution time is normalized to one pixel and disparity so that it can be compared. The average overall disparity error rate is averaged from the overall error rate of all implementations.
TABLE III. DESCRIPTION AND DEFINITION OF DIFFERENT SIMILARITY MEASURES

<table>
<thead>
<tr>
<th>Similarity measure</th>
<th>Description</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAD</td>
<td>Sum of absolute difference.</td>
<td>[ f(u,v,d) = \sum_{x,y \in B(u,v)}</td>
</tr>
<tr>
<td>SAD (_\mu)</td>
<td>Zero-mean SAD. The value of each pixel is first shifted with respect to the mean of the block. Then it is computed in the same way as SAD. This improves SAD robustness to radiometric distortion.</td>
<td>[ f(u,v,d) = \sum_{x,y \in B(u,v)}</td>
</tr>
<tr>
<td>SADm</td>
<td>Zero-median SAD. It is similar to ZSAD except that mean is replaced with median of the block.</td>
<td>[ f(u,v,d) = \sum_{x,y \in B(u,v)}</td>
</tr>
<tr>
<td>NCC</td>
<td>Normalized cross correlation. The cross correlation between the current block and the candidate block are computed. The cross correlation is normalized by the mean value in the block.</td>
<td>[ f(u,v,d) = \sum_{x,y \in B(u,v)} \frac{(I_{\text{curr}}(x,y) - \mu_{\text{curr}}(u,v)) \cdot (I_{\text{cand}}(x + d,y) - \mu_{\text{cand}}(u+d,v))}{\sqrt{\sum_{x,y \in B(u,v)} (I_{\text{curr}}(x,y) - \mu_{\text{curr}}(u,v))^2 \cdot (I_{\text{cand}}(x + d,y) - \mu_{\text{cand}}(u+d,v))^2}} ]</td>
</tr>
<tr>
<td>Rank</td>
<td>Rank difference. The current block and candidate block is transformed into rank, which represents how many pixels have values larger than the block center pixel does.</td>
<td>[ f(u,v,d) = \sum_{x,y \in B(u,v)}</td>
</tr>
<tr>
<td>Census</td>
<td>Census difference is similar to rank except that instead of rank transform, a census transform which uses a bitstream representation is applied first. The bitstream encodes which pixels have values larger than the block center pixel does. The hamming distance between the two censuses is the matching cost.</td>
<td>[ f(u,v,d) = \sum_{i,j \in B(u,v)} \text{Hamming}(\text{Census}<em>{\text{curr}}(x,y) - \text{Census}</em>{\text{cand}}(x + d,y)) ] [ \text{Census}(u,v) = \text{bitstring}(I(i,j) \geq I(u,v)) ]</td>
</tr>
</tbody>
</table>

Fig. 2 marks the average execution time and the overall disparity error rate of each color space and similarity measure combination on an execution time vs. error rate graph. We consider the combination closest to the origin as the best speed-performance trade-off combination, which is Y-only rank (Rank-Y). For Y-only rank, the per pixel-disparity execution time is 3.93 x 10^{-7} sec. while the error rate is 14.81%. The best performance and the fastest combinations are Y-only census (Census-Y) and Y-only SAD (SAD-Y). Y-only census achieved an error rate of 12.59% whereas Y-only SAD has a per pixel-disparity execution time of 3.63 x 10^{-7} sec.. The reason that census performs well is because it can better reflect the probability distribution of a correct match. Although census can achieve better performance, census requires twice the execution time of SAD because of the additional census transform. One way to improve SAD’s performance by about 2% is using SAD \(_\mu\), which takes block average value into consideration. From the above result, we can also see that Y-only and L-only achieved better performance than other 3-component color spaces. The reason might be that stereo block matching relies more on intensity or luminance pattern than chroma color information.

TABLE IV. COMPLEXITY OF DIFFERENT SIMILARITY MEASURE

<table>
<thead>
<tr>
<th>Similarity measure</th>
<th>Per pixel computation complexity</th>
<th>Total operations per image pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAD</td>
<td>Addition: dB + d \text{Absolute:} dB</td>
<td>( d(2B+1)wh )</td>
</tr>
<tr>
<td>ZSAD</td>
<td>Addition: 2dB + d+2B \text{Absolute:} d</td>
<td>((d+1)(2B+2)wh)</td>
</tr>
<tr>
<td>ZMSAD</td>
<td>Addition: dBlogB+ BblogB+ 2dB + d \text{Absolute:} d</td>
<td>((d+1)BlogB+(d+1)B+2)wh)</td>
</tr>
<tr>
<td>NCC</td>
<td>Addition: dB^3+3B \text{Multiplication:} B^3+3B \text{Division:} 2 \times d \text{Square root:} d</td>
<td>((d+1)(4B^2+2)wh)</td>
</tr>
<tr>
<td>Rank</td>
<td>Addition: dB^4+4R \text{Absolute:} d</td>
<td>((dB^2+1)^4Rwh)</td>
</tr>
<tr>
<td>Census</td>
<td>Addition: dB^2+2R \text{Absolute:} d</td>
<td>((dB^2+2)Rwh)</td>
</tr>
</tbody>
</table>
In summary, both color space and execution affect the resulting depth estimation performance. According to the result, using single color component usually yield faster speed and better performance. However, color space had much less impact on execution time than similarity measure had.

V. CONCLUSION

This work studies the impact of color space and similarity measure on complexity, speed, and performance of stereo block matching. The evaluation is performed using the same evaluation stereo image pairs, metric, and computation platform to yield consistent and objective comparison. The complexity comparison shows that similarity measure dominates the computation complexity in stereo block matching. The preliminary execution speed and performance comparison result suggests that Y-only rank (rank-Y) as being the best combination under the trade-off between speed and performance. Further study of the storage requirement as well as bandwidth requirement should be valuable to both software and hardware implementation of stereo matching methods.

REFERENCES


