Linear Complexity Stereo Matching Based on Region Indexing

Marco Antonio Floriano de Oliveira, Raul Sidnei Wazlavick
Federal University of Santa Catarina (UFSC)
88040-900 Florianopolis – SC, Brazil – PO Box 476
{marco, raul}@inf.ufsc.br

Abstract

This paper presents a linear complexity method for real-time stereo matching, in which the processing time is only dependent on the image resolution. Regions along each epipolar line are indexed to produce the disparity map, instead of searching for the best match. Current local methods have non-linear complexity, as they all rely on searching through a correlation space. The present method is limited to a parallel camera setup, because all disparities must occur in the same direction. A continuity constraint is applied in order to remove false matches. The resulting map is semi-dense, but disparities are well distributed. Experimental results on standard datasets reach around 90% of accuracy using the same parameters in all tests.

1. Introduction

Stereo matching is the process of calculating disparities between images with a displacement in the scene viewpoint. This process can be restricted to one dimension with the epipolar constraint, as the disparities always occur along the epipolar lines [6].

Methods for stereo matching may have a local or global approach [9]. In local methods, the disparity calculation is based on a correlation that depends only on the information of a local region. In global methods, this calculation is based on the minimization of a global cost function, producing more accurate disparity maps, with a higher processing time.

Local correlation between the stereo pair can be based on intensity values [4], feature attributes [13], or both [2]. Intensity-based correlation produces dense disparity maps, but tends to generate errors in areas with low or repetitive texture and depth discontinuities. Feature-based correlation is more reliable, but needs feature extraction preprocessing and produces sparse and irregularly distributed disparities.

Current local methods calculate disparities with an inherent combinatorial complexity, as a search is done to find the best match for a point among the n points in the corresponding epipolar line or, at least, a subset of it (the search range). Because of that, they have non-linear time complexity.

In order to produce a disparity map with reasonable quality and a relatively low processing time, recent methods use hardware parallel processing [5], optimized implementations [3], or try to reduce the search range with a number of approaches, such as coarse-to-fine scheme [11] and disparity windows prediction in stereo image sequence [7]. But if a linear complexity, $O(n)$, is desired for processing each epipolar line (which is the case in real-time systems), a search for the best match should be avoided.

In this paper, a method for stereo matching is presented, in which the time complexity is linear to the number of points in the image, $O(mn)$, where m is the number of lines and n is the number of columns. It is based on a region indexing process instead of searching through a correlation space, so that the disparity range between the stereo pair has no impact in the processing time.

2. Region indexing

In current local methods, the correlation function explicitly compares two regions, whether a region is taken as intensity values or feature attributes. This comparison results in an indication on how good is the match. In order to avoid combinatorial complexity, a search for the best match cannot be done. In this case, comparing two regions explicitly is pointless, because a region in the left image can be compared to no more than one region in the right image.

Searching for the best match for each region in a line is $O(n^2)$ for any possible disparity, or $O(ns)$ if the search range is restricted to s. When s = 1, the complexity is $O(n)$, but no search is possible. Thus, if no criterion is used to select a region among others in the same line, the probability of matching successfully a region is $1/n$ and the expected number of successful matches in a line is 1.

An indexing process makes possible to establish a criterion for selecting a correlated region based on how it is evaluated for the index calculation. Assuming that $f(I_{i\,j})$ calculates a discriminant value for the region surrounding the point $(i\,,j)$ in the image *I*, and that *f* maps highly correlated regions to the same value and uncorrelated regions to different values, the matching can be done by indexing each region in both images, taking *f* as an indexing function.

If the cameras are parallel, the projective planes are coplanar and a region $L_{i\,j}$ in the left image can match only a region $R_{i\,j-d}$ in the right image with a disparity $d \geq 0$, which means that all disparities in the line occur in the same direction (i.e. with the same sign). It makes all match candidates to be processed and indexed before reaching $L_{i\,j}$, allowing the matching process to be done in constant time for each region.

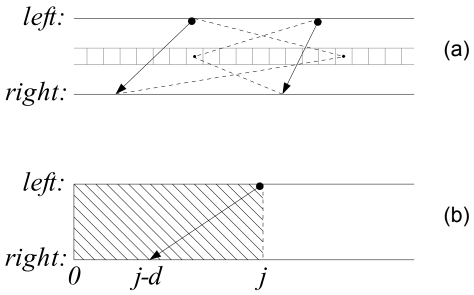


Figure 1. Region indexing: (a) two matches; (b) match limitation (positive disparity).

The indexing function acts as a heuristic function allowing sufficient valid matches to be archived in a lower complexity algorithm. The complexity reduction to $O(n)$ makes the processing time linear and independent on the disparity range, while the amount of valid matches becomes dependent on the indexing function definition.

2.1. Matching algorithm

With a defined indexing function *f*, the regions along corresponding epipolar lines in both rectified images can be indexed in order to calculate disparities. From the beginning to the end of each line, region pairs in each column are processed. A region in the right image is indexed if the position in the vector is free, and a region in the left image match a previously indexed region if the position is taken.

```
for line $i$ from $0$ to $m-1$
    $p \leftarrow 0$
    for column $j$ from $0$ to $n-1$
        if $(vector\,(f(R_{i\,j}))$ is empty$)$
            $vector\,(f(R_{i\,j})) \leftarrow j$
            $list\,(p) \leftarrow f(R_{i\,j})$
            $p \leftarrow p+1$
        end
        if $(vector\,(f(L_{i\,j}))$ is not empty$)$
            $disparity\,(i\,,j) \leftarrow j - vector\,(f(L_{i\,j}))$
            empty vector $(f(L_{i\,j}))$
        end
    end
    for position $j$ from $0$ to $p-1$
        if $(vector\,(list\,(j))$ is not empty$)$
            empty vector $(list\,(j))$
        end
    end
end
```

Figure 2. Basic algorithm for region indexing.

It is possible for $R_{i\,j-d}$ to be incorrectly matched to $L_{i\,j-d\,'}$ before the corresponding region $L_{i\,j}$ is reached, where $0 < d\,' \leq d$, if the area between $j-d$ and j has low or repetitive texture (the discriminant value will be repeated). Often, *d'* is close or equal to *d*.

Those false matches may be hard to detect, but most of them can be avoided in the indexing process by equally displacing all left regions, so that $L_{i\,j}$ can match regions up to column $j+h$, where *h* is a positive displacement value. This way, matches resulting in negative disparities (opposite direction) can be ignored, as disparities always have the same direction due to the restriction shown in Figure 1. The displacement makes possible to check whether a match is done because of a lack of texture variation. Figure 4 shows the algorithm modification regarding this displacement.

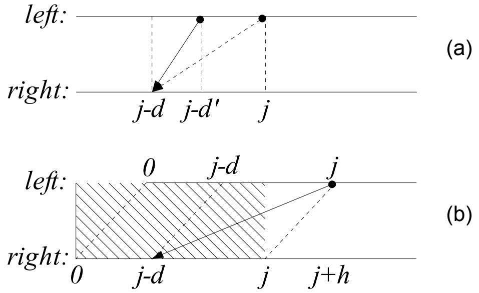
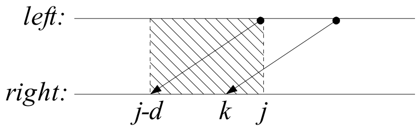


Figure 3. Low or repetitive texture problem: (a) false match; (b) horizontal displacement.

```
for column $j$ from $-h$ to $n-1$
    if $(j+h < n) \wedge (vector(f(R_{i\,j+h}))$ is empty$)$
        $vector(f(R_{i\,j+h})) \leftarrow j+h$
        $list(p) \leftarrow f(R_{i\,j+h})$
        $p \leftarrow p+1$
    end
    if $(j \geq 0) \wedge (vector(f(L_{i\,j}))$ is not empty$)$
        if $(j - vector(f(L_{i\,j})) \geq 0)$
            $disparity(i,j) \leftarrow j - vector(f(L_{i\,j}))$
        end
        empty $vector(f(L_{i\,j}))$
    end
end
```

Figure 4. Indexing regions in a displaced line.

This matching algorithm imposes the restriction that the index $f(R_{i\,j-d})$ is unavailable after $R_{i\,j-d}$ was indexed and before either a false match occurs, or the corresponding region $L_{i\,j}$ is reached and matched. This way, regions in the right image with the same discriminant value cannot be indexed at same time, so the first one locks the position until it is matched.



$$f(R_{i\,j-d}) \neq f(R_{ik}) \qquad j-d \leq k \leq j$$

Figure 5. Index unavailability.

A false match can occur when two uncorrelated regions have the same discriminant value. Even if the indexing function is defined in a way to try to reduce this occurrence, the index contains little information about the region because it is a value in a relatively short interval, as the indexing vector cannot be too big due to memory constraints.

2.2. Indexing function

Valid matches can be calculated if the indexing function maps similar regions to the same value and non-similar regions to different values, where similar means highly correlated or close to a common pattern. In order to remove false matches using continuity constraint, it is important that the disparities resulting from them compose a higher entropy set compared to the disparities resulting from valid matches.

Supposing $nc(L,R)$ is a correlation function normalized to a value in $[0;1]$ that estimate the degree of similarity between the regions *L* and *R*, these three characteristics above can be formally defined as follows:

(i) if two regions $L_{i\,j}$ and $R_{i\,j-d}$ are similar, then there is a high probability that the discriminant values of these regions are the same (i.e. similarity implies equal discriminant, in most cases):

$$nc(L_{i\,j}, R_{i\,j-d}) \approx 1 \quad \rightarrow \; P(f(L_{i\,j}) = f(R_{i\,j-d})) \approx 1 \tag{1}$$

(ii) if two regions $L_{i\,j}$ and $R_{i\,j-\tilde{d}}$ are not similar, then there is a high probability that the discriminant values of these regions are different (i.e. non-similarity implies different discriminant, in most cases):

$$nc(L_{i\,j}, R_{i\,j-\tilde{d}}) \approx 0 \quad \rightarrow \; P(f(L_{i\,j}) \neq f(R_{i\,j-\tilde{d}})) \approx 1 \tag{2}$$

(iii) if two regions $L_{i\,j}$ and $R_{i\,j-\tilde{d}}$ are not similar and their discriminant values are the same, then this false match results in a disparity value $\tilde{d} \in [0;j]$ following an uniform probability distribution:

$$nc(L_{i\,j}, R_{i\,j-\tilde{d}}) \approx 0 \;\; \wedge \;\; f(L_{i\,j}) = f(L_{i\,j-\tilde{d}}) \quad \rightarrow \; P(\tilde{d} = x) \approx \frac{1}{j+1} \qquad 0 \leq x \leq j \tag{3}$$

This last characteristic provides the basis for robust false match detection through continuity constraint. It states that, if the match is not valid, the resulting disparity can assume any possible value with nearly the same probability, as any change in the discriminant value may lead to a completely arbitrary match.

2.3. Intensity-based indexing

In the experimental results shown in this paper, the indexing function $f(I_{i\,j})$ is based on some of the intensity values of $I_{i\,j}$. Each region has a fixed size of *M* rows and *N* columns. This function calculates a discriminant value based on the region mean and how selected points differ from it.

The main term of function *f* is the region index y_{reg}. Each bit in this index indicates whether the corresponding intensity value has reached the region mean:

$$y_{reg}(I_{i\,j}) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \Big[2^{\sigma_K(u,v)-1} \cdot K(u,v) \cdot \delta(I(i+u, j+v), \mu(I_{i\,j})) \Big] \tag{4}$$

The kernel K defines what region points will be used in the index calculation, where $K(u,v) \in \{0, 1\}$, $0 \le u < M$, $0 \le v < N$. The function $\sigma_K(u,v)$ gives the number of points set to 1 in the kernel K, line by line, down to the position (u,v):

$$\sigma_K(u,v) = \sum_{w=0}^{u \cdot N+v} K(w \backslash N, r) \qquad (5)$$
$$w \equiv r (mod\, N)$$

The threshold function $\delta(x,t)$ outputs 1 if $x \ge t$ and 0 otherwise:

$$\delta(x,t) = \begin{cases} 1 & x \ge t \\ 0 & x < t \end{cases} \qquad (6)$$

The function $\mu(I_{ij})$ gives the intensity mean of I_{ij}, which is used as the threshold value in (4):

$$\mu(I_{ij}) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} I(i+u, j+v) \qquad (7)$$

This way, a bit in the region index is set to 1 if the corresponding intensity value is greater or equal to the region mean, and set to 0 otherwise. Only points set to 1 in the kernel are used for this calculation.

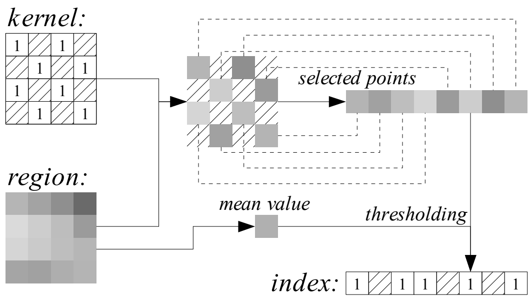


Figure 6. Example of region index calculation.

Increasing or decreasing all intensity values proportionally does not affect the region index. This way, non-similar regions may have the same index, but they can be separated using the mean value. If the indexing vector is segmented based on this value, only regions with the same index and an approximate intensity mean will match. For 2^S desired segments, S is the size, in bits, of the segment index y_{seg}.

The indexing function f concatenates the bits of these two indices:

$$f(I_{ij}) = y_{seg}(I_{ij}) \cdot 2^{\sigma_K(M-1, N-1)} + y_{reg}(I_{ij}) \qquad (8)$$

Where $\sigma_K(M-1, N-1)$ gives the size, in bits, of the region index y_{reg} (it equals the number of points set to 1 in the kernel K, which are the points used for the calculation of this index), and $y_{seg}(I_{ij})$ is the segment index of I_{ij}:

$$y_{seg}(I_{ij}) = \mu(I_{ij}) \backslash 2^{D-S} \qquad (9)$$

Where D is the size, in bits, of an intensity value (i.e. the grayscale depth, so 2^D is the number of possible intensities).

This indexing function is intensity-based only. The indexing vector is segmented to increase the potential amount of indexed regions.

3. False match detection

A continuity constraint for false match detection is applied as a regularization process based on the entropy difference between disparities resulting from valid and false matches. In the indexing process, false matches result in disparities with higher entropy than those resulting from valid matches because of the uniform probability distribution shown in (3) (a disparity can assume any possible value with nearly the same probability if it results from a false match).

Each disparity d has a weight w_d based on the map histogram. This weight is an arithmetic mean of the frequency of similar disparities (any disparity s, where $|s-d| \le 1$) in the map. Thus, the weight vector is a smoothed version of the histogram.

A match is considered valid if the weight sum of similar disparities is above a minimum threshold relative to the weight sum of all disparities in the verification window:

$$\sum_{s=d-1}^{d+1} v_s \cdot w_s \;\ge\; \left(\sum_{a=0}^{n-1} v_a \cdot w_a \right) \cdot (1-\tau) \qquad (10)$$

Where τ is a discontinuity tolerance value in $[0;1]$, V is the histogram of the verification window, and n is the number of columns in the image (disparities may range up to $n-1$). As an additional criterion, a match can be removed if the amount of disparities that are equal to d is below a certain absolute minimum q.

This continuity constraint is applied for each point in the disparity map. If there is no disparity calculated for a point, the last evaluated disparity is used. This reduces sparse areas in the map based on the assumption that a near disparity may still be correctly approved for the new point (facing the new neighborhood), as if resulting from a valid match.

The resulting map can be equalized by replacing each approved disparity by the weighted mean of similar disparities in the verification window:

$$\bar{d} = \frac{\sum_{s=d-1}^{d+1} v_s \cdot w_s \cdot s}{\sum_{s=d-1}^{d+1} v_s \cdot w_s} \qquad (11)$$

4. Experimental results

In results from real images, nearly half of the regions in the right image are indexed (regions with the same discriminant value cannot be indexed at the same time, as shown in Figure 5). Also, nearly half of this indexed regions are matched to a region in the left image. Figures 7, 8, and 9 show intermediate results on "shrub" dataset [1].

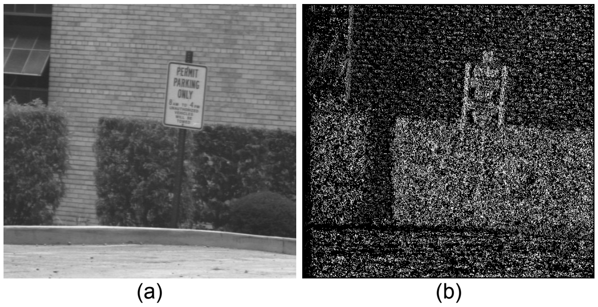

(a) (b)

Figure 7. Results on "shrub" dataset: (a) left image; (b) disparities from region indexing (regions: 51% indexed, 27% matched).

After the false match detection through continuity constraint, the map is semi-dense, but disparities are well distributed and almost all false matches were removed, as shown in Figure 8.

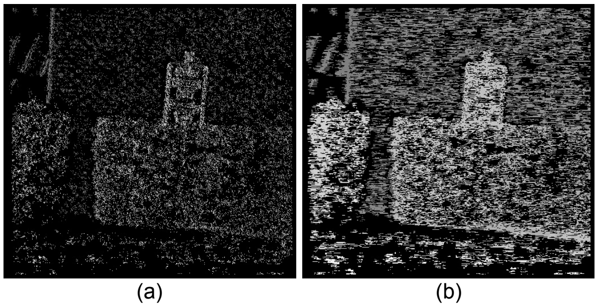

(a) (b)

Figure 8. Continuity constraint: (a) valid disparities (13% density); (b) using last evaluated disparity (40% density).

Remaining sparse areas in the disparity map are filled by nearest interpolation. For fast processing, each point in the map have the value of the nearest valid disparity in the same or adjacent line or column.

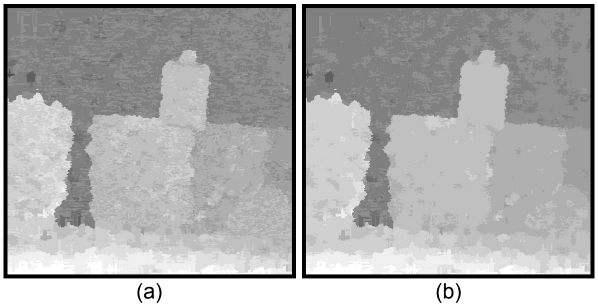
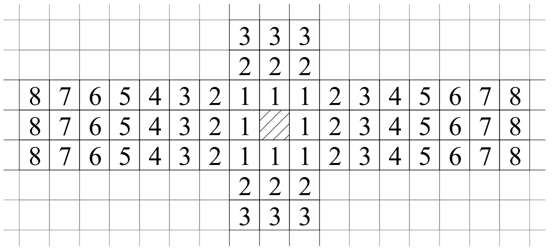

(a) (b)

Figure 9. Nearest interpolated disparities: (a) non-equalized map; (b) equalized map.

This nearest interpolation is limited to orthogonal directions, but can be reasonably uniform, even in cases where the nearest point is considerably distant, because the map is continuous almost everywhere (with the exception of depth discontinuities) after the continuity constraint is applied.



							3	3	3							
							2	2	2							
8	7	6	5	4	3	2	1	1	1	2	3	4	5	6	7	8
8	7	6	5	4	3	2	1	▨	1	2	3	4	5	6	7	8
8	7	6	5	4	3	2	1	1	1	2	3	4	5	6	7	8
							2	2	2							
							3	3	3							

Figure 10. Lines and columns for nearest interpolation.

The disparity map is processed to generate an intermediary map, where each point contains the distances from the two nearest disparities in orthogonal directions and their respective values (one is the last visited in the same or adjacent row, and the other is the last visited in the same or adjacent column). The final map is then calculated by running the original map backwards, keeping only the disparity with the smaller distance in all four directions.

4.1. Parameters

All experimental results shown in this paper were produced with the same parameters. It shows that the method can take different input and produce good results without requiring any adjustment.

Table 1. Region indexing parameters.

stereo displacement, *h*	8 columns
segment index size, *S*	4 bits
region size, *M*×*N*	4×4
used region points, *K*	$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$

Using this set of default parameters, the indexing vector will have 4096 positions (12-bit indices).

Table 2. False match detection parameters.

verification window	15×15
discontinuity tolerance, τ	0.6
min. equal disparities, *q*	8

An arithmetic mean filter with 2×2 window size is applied before the region indexing, because the method shows better results when the input image pair is smoothed. Individual intensities may vary slightly between correlated regions due to differences in the way the same surface is projected on the two cameras. This variation may affect the region index if the intensity value is close to the region mean. Smoothing the image pair causes each bit in the region index to be based on more than one point from the original image, reducing the impact of such variations.

4.2. Standard datasets

The present method produces, in current personal computers, more than 30 frames per second with a resolution of 256×256 (less than a microsecond for each point). This is done for any possible disparity information between the stereo pair, while current methods have a time penalty because of the search range. It has been observed empirically that the processing time depends only on the image resolution, as proposed in theory. The method was tested with the following standard datasets with ground truth and occlusion map:

- “tsukuba”, “sawtooth”, “venus” and “map” [9];
- “cones” and “teddy” [10].

The error is calculated by subtracting the resulting map from the ground truth and counting all absolute differences greater than 1, ignoring points in occluded areas (i.e. areas where a match cannot be done) and in the image borders (18 pixels for “tsukuba” dataset and 10 pixels for the others), as proposed by Scharstein et al. to evaluate the results from stereo methods [9].

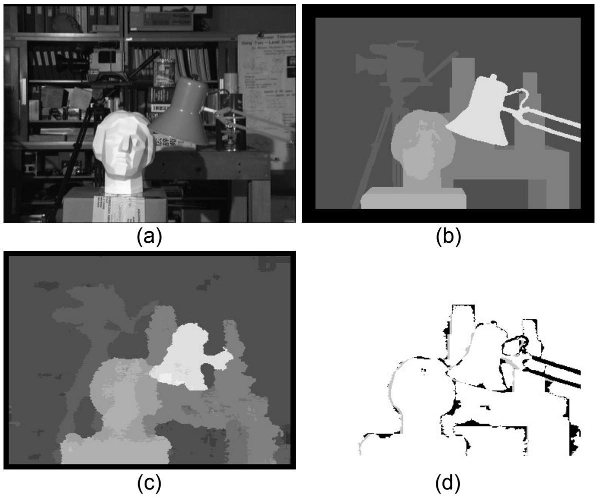


Figure 11. Results on “tsukuba” dataset: (a) left image; (b) ground truth; (c) calculated map; (d) bad pixels (occlusion in gray).

This way, the error is the percentage of bad pixels in non-occluded areas. The “tsukuba” dataset was first introduced by Nakamura et al. [8], and the “map” dataset by Szeliski et al. [12].

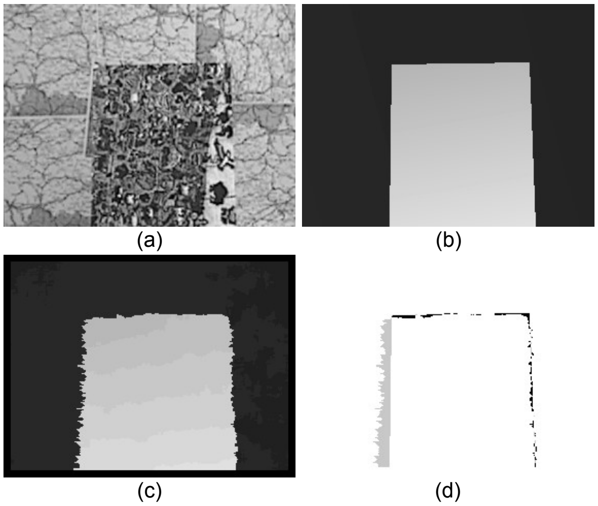


Figure 12. Results on “map” dataset: (a) left image; (b) ground truth; (c) calculated map; (d) bad pixels (occlusion in gray).

Most of the disparity error concentrates near the object borders and thin objects are lost, as can be seen in the error map (d) in Figure 11. Most of the bad pixels in Figure 12 are due to occlusion. Disparities in these areas have high entropy and can be removed, but the respective borders are affected because of the resulting sparse areas.

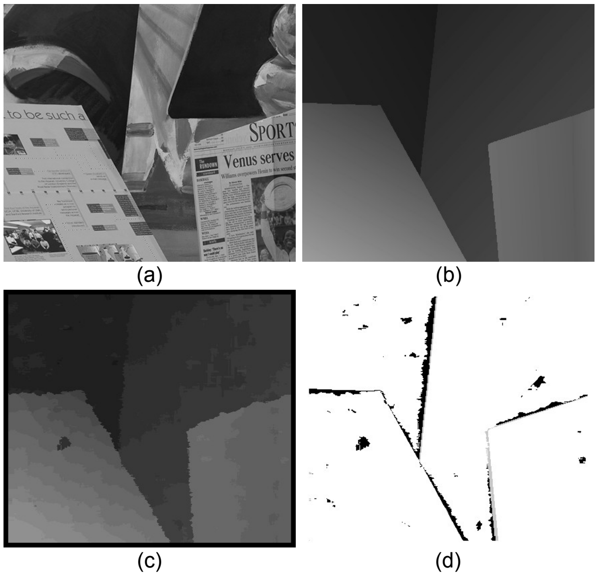


Figure 13. Results on “venus” dataset: (a) left image; (b) ground truth; (c) calculated map; (d) bad pixels (occlusion in gray).

Some error spots in Figure 13 are due to repetitive texture. As the region indexing is based on a local assumption of similarity (regions with the same discriminant value are probably correlated), repetitive texture may form dense groups of false matches which cannot be removed applying the continuity constraint, but can be mostly ignored in the region indexing phase, using the algorithm modification shown in Figure 4.

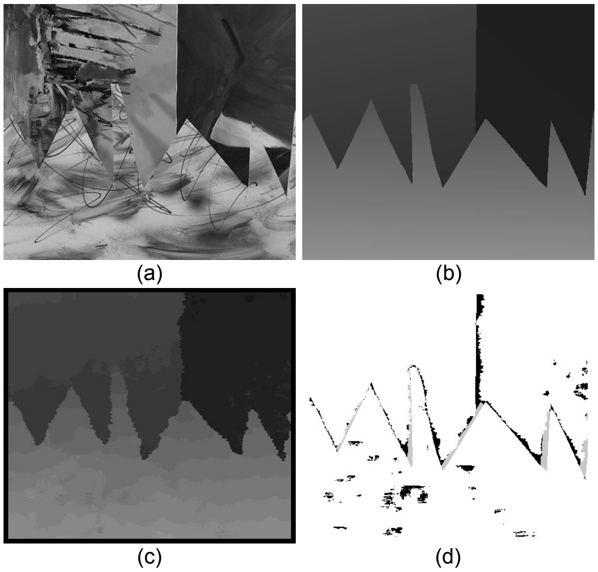


Figure 14. Results on “sawtooth” dataset: (a) left image; (b) ground truth; (c) calculated map; (d) bad pixels (occlusion in gray).

Table 3. Results on standard datasets.

	map	***tsukuba***	***sawtooth***	***venus***	***cones***	***teddy***
resolution	284×216	384×288	434×380	434×383	450×375	450×375
indexing						
regions	59853	108585	162487	163780	166284	166284
indexed	79%	67%	72%	72%	71%	68%
matched	35%	35%	37%	34%	35%	33%
time	10 ms	17 ms	27 ms	27 ms	27 ms	27 ms
continuity						
valid disp.	25%	25%	27%	22%	22%	20%
final density	66%	59%	64%	55%	54%	51%
time	15 ms	26 ms	42 ms	41 ms	42 ms	41 ms
interpolation						
time	2 ms	4 ms	6 ms	6 ms	6 ms	6 ms
error	**0.63%**	**4.07%**	**3.33%**	**3.23%**	**5.68%**	**9.91%**
total time	**27 ms**	**47 ms**	**75 ms**	**74 ms**	**75 ms**	**74 ms**

The data within the images has negligible influence on the processing time (Figure 15), because the method complexity is not dependent on the disparity range, and each disparity is calculated in constant time.

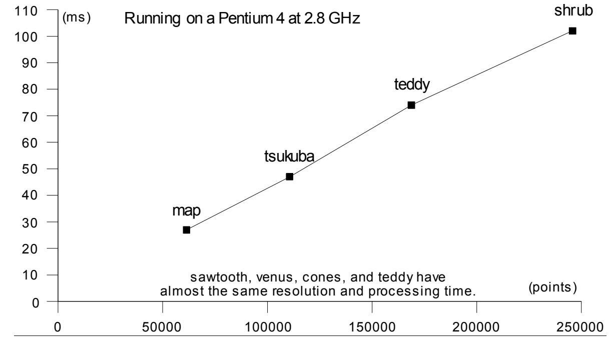


Figure 15. Processing time for datasets with different resolutions.

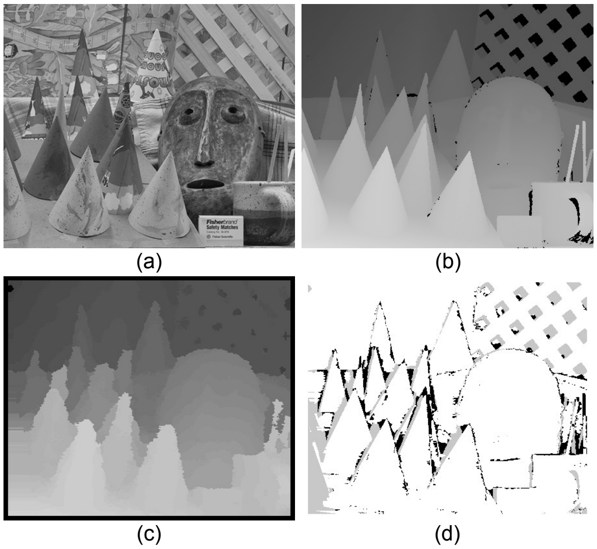


Figure 16. Results on “cones” dataset: (a) left image; (b) ground truth; (c) calculated map; (d) bad pixels (occlusion in gray).

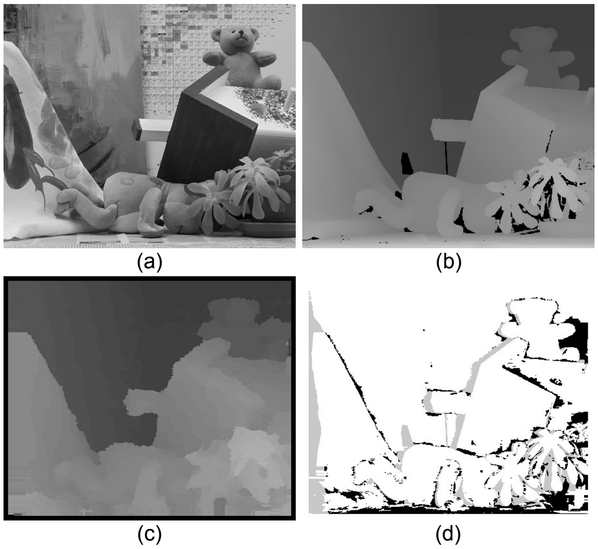


Figure 17. Results on "teddy" dataset: (a) left image; (b) ground truth; (c) calculated map; (d) bad pixels (occlusion in gray).

5. Conclusion

The main contribution of this work is the definition of a linear complexity method for stereo matching. The use of an indexing process makes possible for each epipolar line to be computed in $O(n)$ for calculating disparities, which results in linear complexity to the number of points in the image. Current local methods make a search through a correlation space, which results in non-linear complexity.

In order to match a region, all match candidates in the corresponding epipolar line must be indexed first. This makes all disparities to occur in the same direction (i.e. to have the same sign), so the projective planes must be coplanar, limiting the method to a parallel camera setup.

The resulting map is not dense and must be interpolated, but disparities are well distributed, with some sparse areas due to low texture and occluded points. As a consequence of occlusion and simplicity of the indexing function, object borders are not well defined and concentrate most of the error when comparing with the ground truth. Nevertheless, the accuracy is nearly 90% on standard datasets, using the same parameters in all tests.

6. Future work

The proposed method is suitable for real-time systems that require stereo vision, such as robot vision and augmented reality.

An interesting extension of this work is the application of region indexing to motion estimation. In this case, the complexity remains $O(mn)$, but the indexing vector should be used for the whole image instead of each separate line, as the disparities occur in 2D, and not only horizontally.

Other possible extension is the use of an indexing function based on feature attributes.

7. References

[1] R. C. Bolles, H. H. Baker, and M. J. Hannah, "The JISCT Stereo Evaluation" *DARPA Image Understanding Workshop*, Apr. 1993, pp. 263–274.

[2] S. D. Cochran and G. Medioni, "3-D Surface Description from Binocular Stereo", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, Oct. 1992, pp. 981-994.

[3] L. Di Stefano, M. Marchionni, S. Mattoccia, and G. Neri, "A Fast Area-Based Stereo Matching Algorithm", *Image and Vision Computing*, vol. 22, Oct. 2004, pp. 983-1005.

[4] H. Hirschmuler, "Improvements in Real-Time Correlation-Based Stereo Vision", *IEEE Workshop on Stereo and Multi-Baseline Vision*, Dec. 2001, pp. 141-148.

[5] H. Jeong and S.-C. Park, "Trellis-based Systolic Multilayer Stereo Matching", *IEEE Workshop on Signal Processing Systems*, Aug. 2003, pp. 257-262.

[6] Q.-T. Luong and O. D. Faugeras, "The Fundamental Matrix: Theory, Algorithms, and Stability Analysis", *International Journal of Computer Vision*, vol. 17, Jan. 1996, pp. 43-75.

[7] J. Mulligan and K. Daniilidis, "Predicting Disparity Windows for Real-time Stereo", *Proceedings of the 6th European Conference on Computer Vision*, vol. 1, Jun. 2000, pp. 220-235.

[8] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta, "Occlusion Detectable Stereo – Occlusion Patterns in Camera Matrix", *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 1996, pp. 371–378.

[9] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", *International Journal of Computer Vision*, vol 47, Apr. 2002, pp. 7-42.

[10] D. Scharstein and R. Szeliski, "High-Accuracy Stereo Depth Maps Using Structured Light", *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, Jun. 2003, pp. 195-202.

[11] C. Sun, "Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques", *International Journal of Computer Vision*, vol. 47, May 2002, pp. 99-117.

[12] R. Szeliski and R. Zabih, "An Experimental Comparison of Stereo Algorithms", *International Workshop on Vision Algorithms*, Sep. 1999, pp. 1-19.

[13] O. Veksler, "Dense Features for Semi-Dense Stereo Correspondence", *International Journal of Computer Vision*, vol. 47, Apr. 2002, pp. 247-260.