Geometric Models Reconstruction Using Inverse Perspective Projection

Antônio Wilson Vieira
DCC - UFMG - Belo Horizonte - MG
CCET - UNIMONTES - Montes Claros - MG
awilson@dcc.ufmg.br

Mário Fernando Montenegro Campos
DCC - UFMG - Belo Horizonte - MG
mario@dcc.ufmg.br

Abstract

This work proposes to study the IPP, Inverse perspective projection, and some techniques for solid modeling in order to deal with the point cloud from stereo vision procedure and to represent the objets in a data structure more accurate for post processing like mesh simplification that avoids dense redundant vertex representation for planar regions. In our experiments, we used a feature-based strategy to obtain a partial disparity map from correspondent edges in two stereo images and constructed a dense disparity map interpolating the previous partial map. This disparity map and a known camera calibration system leads to a 3-D scene reconstruction in a regular ranger image matrix. Then we write this output triangular mesh in a topological data structure and apply a mesh simplification strategy to obtain an optimized representation to the scene model.

1. Introduction

The problem of Inverse Perspective Projection, IPP, is related to the problem of stereo vision. The goal is to obtain 3-D information from a set of, at least two, digital images of a scene. In general, the problem is treated separately in tree specific problems: Calibration, Correspondence and Reconstruction. The calibration process intend to get the projection matrix or the intrinsic and extrinsic camera parameters. The correspondence problem relies on reaching points in different images that are associated to the same point in the scene. Finally, the reconstruction problem deals with the problem of building the 3D scene geometry combining the disparity map from correspondence results and camera calibration parameters. A big challenge in all process of reconstruction is to solve the correspondence problem to build the disparity map, that serves as input for the well resolved algorithms of reconstruction. However, the correspondence, or matching, algorithms usually takes a point in one image and tries to find its correspondence in the other image using some similarity criterion based on a neighboring region of the two image points, witch is an ambiguous match since many regions of the same image may attend this similarity. Some knowledge about the cameras calibration during the images acquisition can reduce the problem of searching a point from one image over the entire other image to the problem of searching over a reduced region. In fact, the epipolar geometry allows to reduce the search region to a line, or a 1-D segment, that is, anyway, an ambiguous problem, specially for points that belong to the flat regions without texture. The ocorrence of texture, in fact, reduces ambiguity, giving a stronger signature to each point in the so called Correlation-based methods. Other approach, called Feature-based methods tries to match specific set of points in the images that are related to some significant feature. This method require a pre-processing step of feature-detect in both images and, then, search for corresponding feature in the images according to the feature descriptors. In this work we assume images from a structured scene with easy geometry description, without texture to allow good results in edge detection algorithms for, then, use a Feature-based matching algorithm to build a disparity map interpolating results from feature matching. The principal contribution with this work is a post-processing to the range image, after the scene reconstruction, in order to obtain an optimized representation to the scene with a mesh simplification step that resume large planar regions from the scene to a parametric patch description. This optimized polygon based representation is extremely efficient and largely used in solid modeling for CAD, game development, math research and medical 3-D image processing because of its hight expressiveness, low storage cost and hight processing performance for rendering and motion modeling.

2. Prior Works

The problem of stereo vision has a definition in the articles from Barnard and Fischler [1]. The work of Koschan [9] presents some techniques for treating the problem and the evolution of stereo vision, and the work of Brown et al [2] presents a comparison of stereo vision methods that, usually, give more attention to correspondence problem then to calibration an reconstruction.

The problem of calibration is treated with algorithms which are able to estimate, explicitly, the intrinsic and extrinsic parameters or to estimate, directly, the projection matrix as discussed from Trucco and Verri [13] and is well treated in literature as in works from Faugeras and Luong [5] and Hartley and Zisserman [7].

The problem of correspondence is a key in stereo vision and, because its ambiguity [4], several assumptions are usually made for particular solutions and make it hard to find a general solution to the problem. The assumptions on correspondence may be geometric, about the projection point of the camera or objects shape, or radiometric, like assuming that the light model and surfaces in the scene are lambertians. Such assumptions, by one side, simplify the solutions but, by other side, bring doubts to the results as long as the real scene diverges from the assumptions. Brown et al [2] classifies the correspondence methods in local and global and gives some references of work in each method.

The problem of reconstruction, or 3-D perception, is related to the understanding of difference, or disparity, between correspondent points in distinct images from the same scene. The disparity calculated in all image points, define the disparity map [13] that, in this case, intend to establish a relationship among the projection points and their corresponding point in 3-D space. This process and its result is strongly depending of the results from the problems of calibration and correspondence [2].

Most work in stereo vision doesn't take in account the influence of environment, assuming ideal vacuum light propagation but, in practice, different environment like underwater require a special treatment to deal with light refraction, as shown in the work of Neto at al [11]. Another research area, complementary to this, called solid modeling develops data structure to store and process geometrical models far better than the point clouds structure gotten from stereo vision. Such data structures, like the simple $Corner - Table$ presented by Lopes and Rossignac in [10] allows compression, re-mashing and mesh simplification operations. Vieira at al presented, in [14], an application of $Corner - Table$ data structure to implement the mesh simplification algorithm proposed by Heckbert and Garland in [6]. This work intend to go trough the stereo vision process using density images as input and output a scene representation more forehanded then the usual range image.

3. Methodology

To develop the work that we propose for this article, we need to obtain 2 stereo images from a semi-structured scene and process them in order to obtain a 3-D description of the scene in a simplified mesh data structure. The methodology we will use for such work is described in the following steps: image acquisition, edge detection, correspondent edge matching, interpolated dense disparity, 3-D reconstruction from disparity map, topological mesh description and mesh simplification. In order to detail each step, we treat them separately in a subsection for each.

3.1. Image Acquisition

The images used in our experiments are acquired from static scene of objects with well defined geometry and without dense texture. We consider a Lambertian surface reflectance model and assume a simple calibrated stereo system composed of two pinhole cameras with parallel optical axes and the same focal length. The Figure 1 illustrates this model as presented in [13].

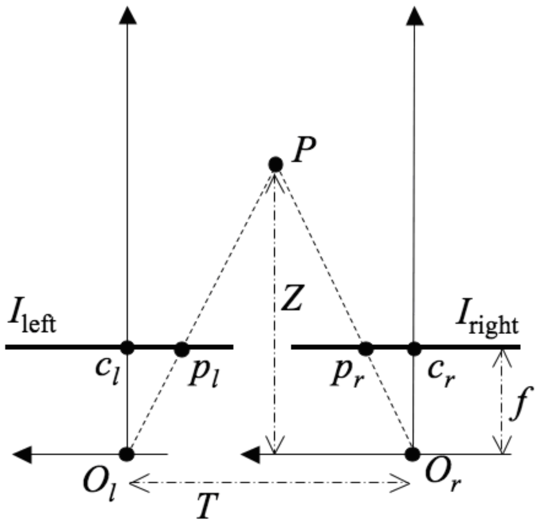


Figure 1. Simple Stereo System.

The left and right images I_l and I_r will be the input for the next steps.

3.2. Edge Detection

In the edge detection step, we apply a gaussian noise filtering to suppress as much of the image noise as possible and use the Canny edge detector, as in [3], in both images in order to output two binary images E_l and E_r with border pixel sets.

3.3. Correspondent Edge Matching

As stated in the literature of stereo vision, global matching is a widely difficult task, as shown for Scharstein and

Szeliski in [12], specially for images I with large flat regions poorly textured. Such images can be characterized for large regions R with null gradient.

$$\nabla I(p) = (0,0), p \in R$$

However, incident regions with null gradient use to have sharp transitions with strong gradient that lead to good edge detection. Then we'll assume as input, in this step, the preprocessed binary images E_l and E_r with border pixel sets gotten as output of prior step Edge Detection.

$$E(p) = \begin{cases} 0 & , \quad p \notin border \\ 1 & , \quad p \in border \end{cases}$$

Then, using the Hough transform for lines [8], we give a signature to each line $L_l \in E_l$ and $L_r \in E_r$, so that each line $L \in (E_l \cup E_r)$ has a feature descriptor $L = (l, o, x, y)$ where:

- **l** is the lenth
- **o** is the orientation
- **(x,y)** are the midpoint coordinates

Using such descriptor for two lines L_l and L_r in the distinct images, we compute their similarity using the sum of weighted square distance, S, among the proprieties in the descriptors:

$$S = a(l_l - l_r)^2 + b(o_l - o_r)^2 + c(x_l - x_r)^2 + d(y_l - y_r)^2$$

where a, b, c, d are weights, and subscripts l and r refer to the left and right image, respectively.

Finally, we use this similarity measure to run a feature matching algorithm that outputs a list of corresponding lines in both images. The partial disparity map D, is than calculated from the corresponding lines.

3.4. Interpolated Dense Disparity

Using the partial disparity map D, calculated for each border line in the previous step, we assume flat regions defined by the borders and than estimate the disparity in each point as a linear interpolation of border disparity. The process describe a scan-line that, given a line i of D, start from the pixel $(i, 0)$, get the disparities $d_1 = D(i, j_1)$ and $d_2 = D(i, j_2)$ from the first pair of border pixels (i, j_1) and (i, j_2) to calculate the disparity $D(i, k)$ as

$$D(i,k) = d_1 + \frac{k - j_1}{j_2 - j_1}(d_2 - d_1)$$

for each $k \in [j_1, j_2]$.

By repeating this process for each line of the partial disparity map, we construct, as output, a dense disparity map D assuming flat regions defined by the borders.

3.5. 3-D Reconstruction

In our simple stereo model used for acquisition, the left and right images were coplanar with parallel optical axes, common focal length f and known baseline with length T. Hence, using the disparity map D, as input, we can compute the depth Z for each point using triangular similarity that leads to the equation

$$\frac{T - d}{Z - f} = \frac{T}{Z}$$

where d is the point disparity and, solving for Z, gives

$$Z = f\frac{T}{d}$$

Now we have a range map, that we call M, representing a regular 6×6 triangular mesh with vertices geometry $(i, j, M(i, j))$ and easy regular topolgy descriptor as output.

3.6. Topological Mesh Description

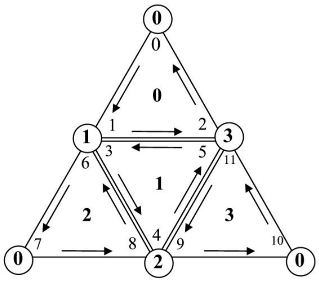
In order to resize the mesh, removing redundant information, especially in large planar regions of the reconstructed surface, we should model the mesh in a special data structure to deal with triangles of arbitrary size instead of regular matrix representation. A very concise data structure for triangular meshes, like the Corner–Table (CT), servers to our goals. The CT uses the concept of *corner* to represent the association of a triangle with one of its bounding vertices, or equivalently the association of a triangle with its opposite bounding edge to that corner.

In this data structure, the corners, the vertices and the triangles are indexed by non–negative integers. Each triangle is represented by 3 consecutive corners that define its orientation. For example, corners 0, 1 and 2 correspond to the first triangle, the corners 3, 4 and 5 correspond to the second triangle and so on… As a consequence, a corner with index c is associated with the triangle of index $trig(c) = floor\,(c/3)$.

The Corner–Table data structure represents the geometry of a surface by the association of each corner c with its geometrical vertex index V$[c]$.

Assuming a counter–clockwise orientation, for each corner c, the $next(c)$ and $prev(c)$ corners on its triangle boundary are obtained by the use of the following expressions: $next(c) = 3 \times trig(c) + [(c + 1) \bmod 3]$, and $prev(c) = 3 \times trig(c) + [(c + 2) \bmod 3]$.

The edge–adjacency between the neighboring triangles is represented by associating to each corner c, its opposite corner O$[c]$, which has the same opposite edge. This information is stored in two integer arrays, called the V and O tables. Figure 2 shows an example of a Corner–Table representation for a tetrahedron.



V[0]	= 0	O[0]	= 4
V[1]	= 1	O[1]	= 9
V[2]	= 3	O[2]	= 8
V[3]	= 1	O[3]	= 10
V[4]	= 2	O[4]	= 0
V[5]	= 3	O[5]	= 7
V[6]	= 1	O[6]	= 11
V[7]	= 0	O[7]	= 5
V[8]	= 2	O[8]	= 2
V[9]	= 2	O[9]	= 1
V[10]	= 0	O[10]	= 3
V[11]	= 3	O[11]	= 6

Figure 2. Tetrahedron example.

The CT concisely represents the connectivity of a triangular mesh using only the arrays O and V. To represent the mesh geometry, an array G is used to store the geometry of the vertices (coordinates, normals,...).

3.7. Mesh Simplification

The mesh simplification process uses topological operators to remove vertices from the mesh while maintaining the mesh structure consistence during simplification. The decision of witch vertex to remove is based on a local cost estimator that evaluates the geometrical distortion caused by a vertex removal operation. In this section we describe the topological operator Edge–Collapse and a geometrical cost estimator called Quadric Error Metric from [6].

The Edge-Collapse operator consists in removing an edge $e = (u, v)$ from a surface S, identifying its vertices to a unique vertex $\overline{v}$. From a combinatorial viewpoint, this operator will remove 1 vertex, 3 edges and 2 faces from original mesh, thus preserving its Euler characteristic. From a geometric viewpoint, the new position of the vertex $\overline{v}$ can be computed with the geometry around u and v.

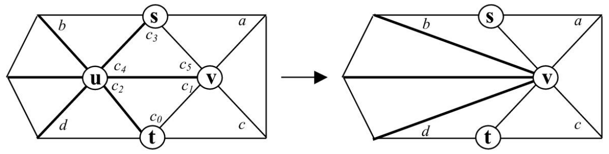


Figure 3. Edge Collapse.

The figure 3 illustrates the collapse of an edge $e = (u, v)$, identified by the corner c_0 and the following algorithm, from [14] details de process.

```
EdgeCollapse($c_0$)
  $c_1 = next(c_0)$; $c_2 = prev(c_0)$; $c_3 = O[c_0]$;
  $c_4 = next(c_3)$; $c_5 = prev(c_3)$;
  $a = O[c_4]$; $b = O[c_5]$;
  $d = O[c_1]$; $c = O[c_2]$;
  $i = c_2$;
  Do
    $V[i] = V[c_1]$;
    $i = next(right(i))$;
  $While(i! = c_2)$;
  $O[a] = b$; $O[b] = a$;
  $O[c] = d$; $O[d] = c$;
```

The Quadric Error Metric presented in [6] is an efficient geometrical cost estimator widely used in mesh simplification algorithms. This method evaluates a cost C_e for collapsing an edge $e = (u, v)$ in a resulting vertex w as

$$C_e = w^t(Q_u + Q_v)w$$

where,

$$Q_v = \sum Q_i$$

with Q_i defined, for each triangular face f_i incident to v, as

$$Q_i = n.n^t = \begin{pmatrix} n_x^2 & n_xn_y & n_xn_z & n_xd \\ n_xn_y & n_y^2 & n_yn_z & n_yd \\ n_xn_z & n_yn_z & n_z^2 & n_zd \\ n_xd & n_yd & n_zd & d^2 \end{pmatrix}$$

and $n = (a, b, c, d)$, the implicit parameters to the plane $(ax + by + cz = d)$ support of a triangular face f_i.

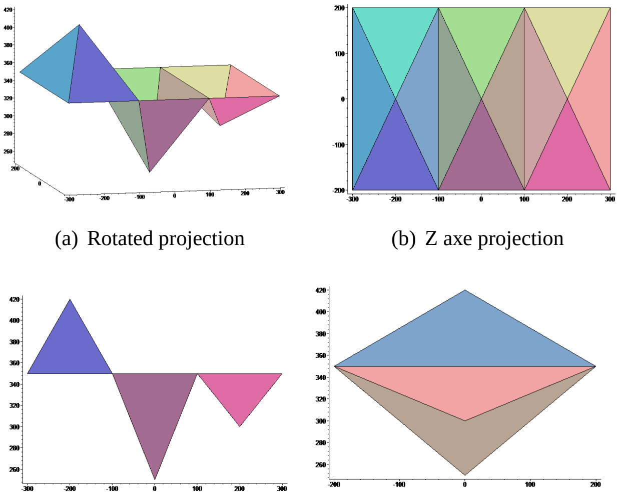
This cost is an estimation of the total slop suffered for planes incidents to the vertices u and v if they were collapsed and allows a greedy simplification algorithm that removes vertices from the mesh until a stop condition is reached. This stop condition may be based in the number of vertices to be removed or a threshold limit to the accumulated geometrical error.

4. Experiments and Results

Our experiments were made using a computer with the following configuration:

- **Hardware:** Notebook Dell Inspiron 1525 with an Intel Core 2 Duo CPU of 2.00 GHz and RAM memory of 2.00GB.
- **Software:** Operational system of 32 bits MS Windows Vista, compiler Visual C++ and graphic library resources of OpenGL.

We built a geometrical model of a polygonal surface with 12 nonparallel triangular faces and used a virtual camera to take front parallel stereo image pairs with different known baselines in order to verify the correctness of our method to obtain the disparity map. Figure 4 details this model.



(a) Rotated projection

(b) Z axe projection

(c) Y axe projection

(d) X axe projection

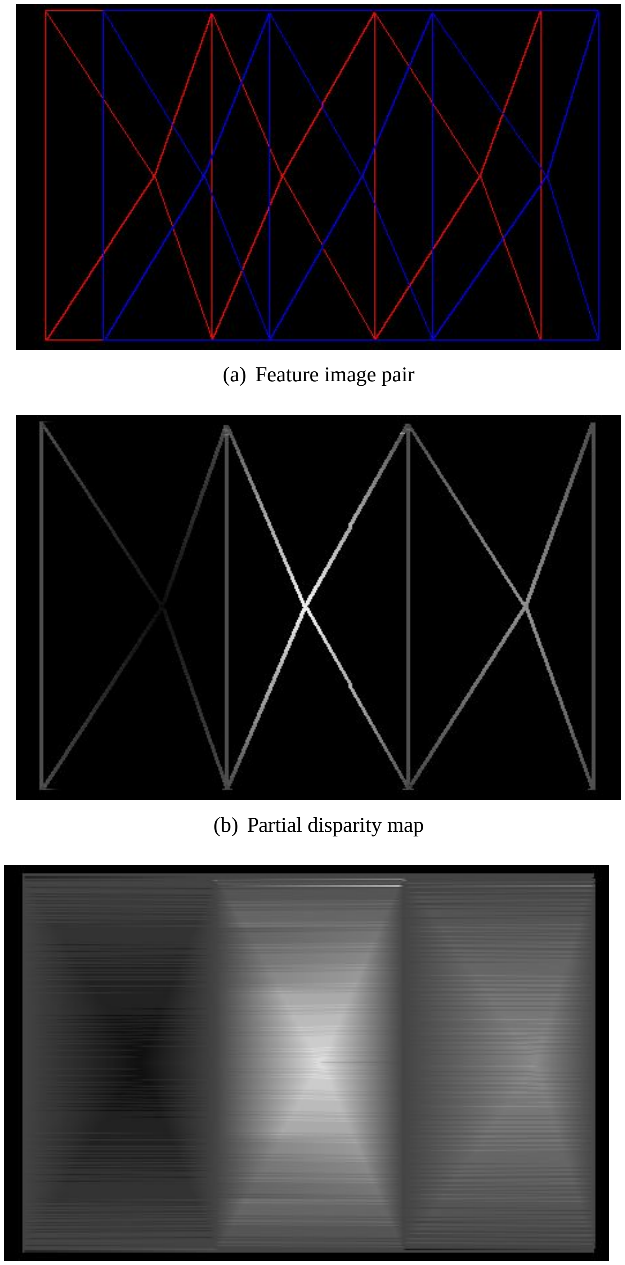
Figure 4. Geometrical model.

Using such model, we got a sequence of 8 stereo image pairs with our virtual camera whose baselines were 2, 4, 8, 16, 32, 64, 128 and 256, respectively. Then, for each stereo pair, we used a feature detect filter to find the edges from both images and, using a feature match algorithm as presented in section 3.3, we got the correspondent edge pairs, for baseline $T = 64$, as shown in Figure 5 (a) in red color for left image and blue for right image.

The disparity were, firstly, calculated for the edges, using the correspondence already known for the edges. The Figure 5 (b) shows, in gray scale for intensity, the disparity calculated for the border images in (a).

The full disparity map were, than, calculated using this partial map, as the interpolated disparity value for each image point according to the algorithm presented in section 3.4. The Figure 5 (c) shows, in gray scale for intensity, the the disparity calculated for all points interpolating edge values as shown in (b).

As we measure disparity in pixels, we observed that, relatively small value of baseline causes the disparity to have too strong discontinuities and, in order to have a more continue and smooth disparity map, we need a greater value for baseline. Though, we have also observed that, relatively large values of baseline causes the algorithm to mismatch corresponding features. This occurs because as long as we enlarge the baseline, we also enlarge the space for ambiguities that lead to false positive match. To illustrate this effect we show in Figures 6 and 7 the full disparity map calculated for our sequence of 8 stereo pair with exponential increasing value for baseline.



(a) Feature image pair

(b) Partial disparity map

(c) Full disparity map

Figure 5. Constructing disparity map.

To develop the 3D reconstruction we used the expression presented in section 3.5 that uses only the fixed baseline and focal length that, in our virtual camera, were set to 200 since we put the projection center to $z = 50$ and the image plane to $z = 250$. Hence we had a focal length $f = 200$ and, in each experiment, we saved a range image reconstructed using the corresponding disparity map. Since we knew the range value for each real point in the scene, we than calculated an error associated to each stereo pair using accumulated absolute value of difference for each point as shown in Table 1.

Baseline	**Error**	$\frac{\textbf{Error}}{\textbf{Pixel}}$	**Error %**
2	48.193.845	201	47.81
4	31.602.195	132	31.35
8	17.403.090	73	17.26
16	11.760.110	49	11.67
32	8.236.862	34	8.17
64	3.486.456	15	3.46
128	9.948.274	41	9.87
256	26.614.962	111	26.40

Table 1. Error comparison.

In table 1, the second column have accumulated error in all pixels, third column have average error per pixel and last column have a percentage error in the real range of each pixel.

Using the range image representation for the reconstructed scene we need a 600×400 integer matrix for the entire surface representation. Applying the mesh simplification as presented in section 3.7 we were able to reduce this representation to only 57 vertices taking 3 integer coordinates each and a topology representation that takes 106 integers. In practice, this representation could be reduced to only 11 vertices and a topology representation taking only 36 integers if the disparity map gave us a 3D reconstruction without any error, so that each planar face could be correctly delimited for the simplification algorithm.

5. Conclusion and Future Work

We have studied, in this work, the IPP, Inverse perspective projection. In our experiments, we used a feature-based strategy to obtain a partial disparity map from correspondent edges in two stereo images and constructed a dense disparity map interpolating the previous partial map. This disparity map and a known camera calibration system leads to a 3-D scene reconstruction in a regular ranger image matrix. Then we write this output triangular mesh in a topological data structure and apply a mesh simplification strategy to obtain an optimized representation to the scene model.

We had to dedicate special effort to implement the correspondence step of our pipeline, so that we realized, as confirmed in the dense literature in the issue, that correspondence is a key in stereo vision and a difficult problem to solve. Our strategy was to explore the process of stereo vision applying to a structured geometrical model without texture in order to use a feature based match algorithm that was able to build a partial disparity map in the edges that was than extended to a dense disparity map using interpolation. This method as shown efficient in our mathematical model and we intend to extend this to real stereo image pairs using arbitrary camera position instead the simple front parallel used here. Although our results didn't bring any innovative development to the stereo vision research at all, is was a useful workout to absorb the principal concepts and challenges of this area.

References

[1] S. T. Barnard and M. A. Fischler. Computational stereo. 14:553–572, 1982.

[2] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.

[3] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[4] O. Faugeras, B. Hotz, H. Mathieu, T. Vieville, Z. Zhang, P. Fua, E. Theron, G. Berry, J. Vuillemin, P. Bertin, and C. Proy. Real time correlation based stereo: algorithm implementations and applications. *The International Journal of Computer Vision*, 1996.

[5] O. Faugeras and Q. T. Luong. The geometry of multiply images. *The MIT Press*, 2001.

[6] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. *Computer Graphics*, 31(Annual Conference Series):209–216, 1997.

[7] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. *Cambridge Univ. Press*, 2000.

[8] J. Illingworth and J. Kittler. A survey of the hough transform. *Comput. Vision Graph. Image Process.*, 44(1):87–116, 1988.

[9] koschan1. What is new in computational stereo since 1989: A survey of current stereo papers. Rapport de recherche 93-22, Universit de Columbia, Allemagne, august 1993.

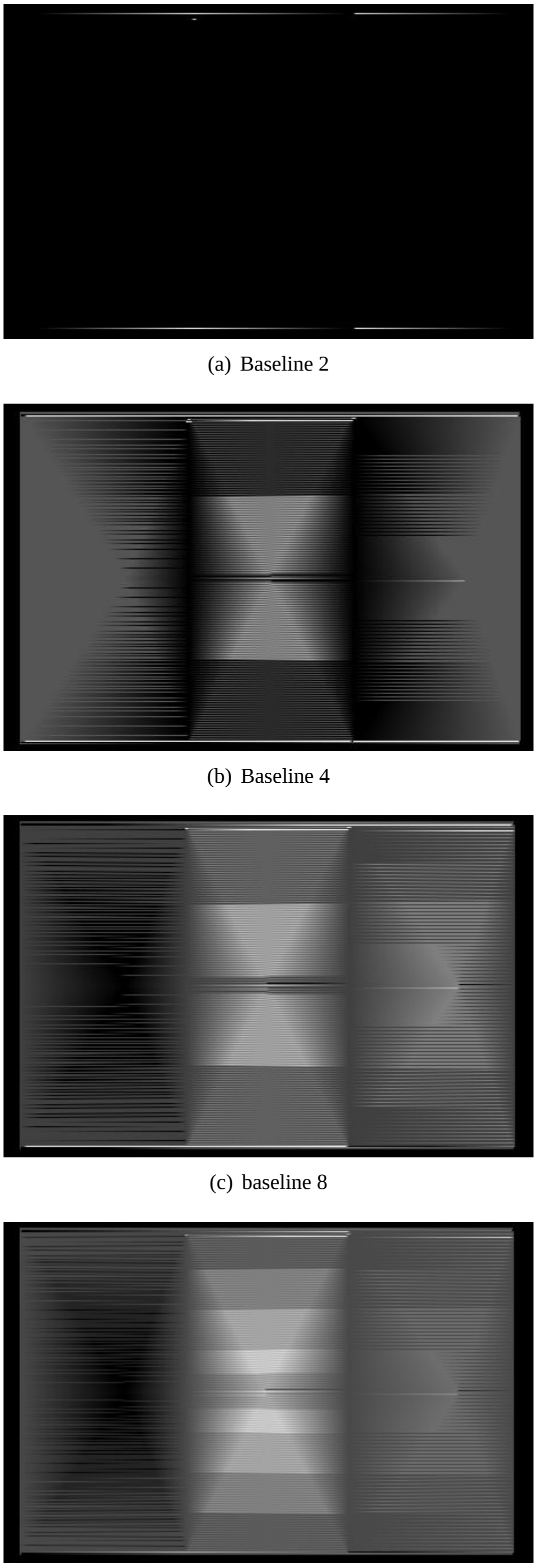
[10] H. Lopes, J. Rossignac, A. Safonova, A. Szymczak, and G. Tavares. Edgebreaker: a simple compression for surfaces with handles. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 289–296. ACM Press, 2002.

[11] Q. Neto, R. Carceroni, W. Barros, and M. Campos. Underwater stereo. *Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing*, 1:170–177, 2004.

[12] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 47:7–42, 2001.

[13] E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision*. Prentice Hall, USA, 1998.

[14] A. W. Vieira, T. Lewiner, L. Velho, H. Lopes, and G. Tavares. Stellar mesh simplification using probabilistic optimization. *Computer Graphics Forum*, 23(4):825–838, october 2004.



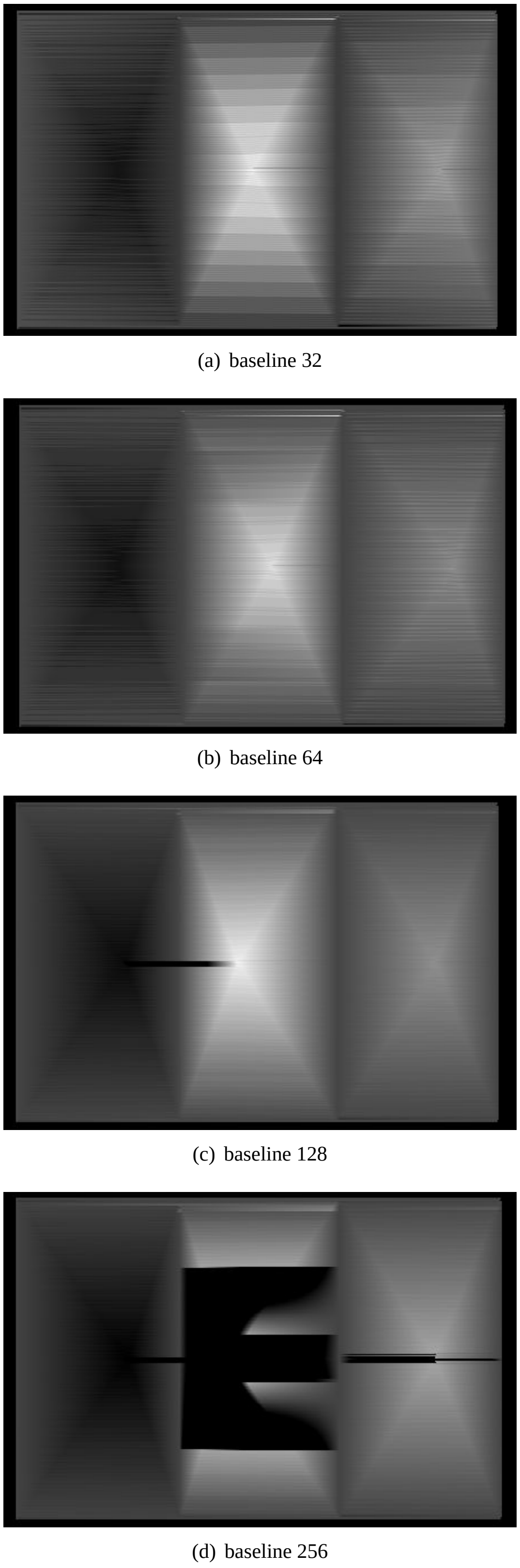
(a) Baseline 2

(b) Baseline 4

(c) baseline 8

(d) baseline 16

Figure 6. Influence of baseline from 2 to 16.



(a) baseline 32

(b) baseline 64

(c) baseline 128

(d) baseline 256

Figure 7. Influence of baseline from 32 to 256.