



HAL
open science

A Dynamic Programming Algorithm Applied to Omnidirectional Vision for Dense 3D Reconstruction

Rémi Boutteau, Xavier Savatier, Jean-Yves Ertaud

► **To cite this version:**

Rémi Boutteau, Xavier Savatier, Jean-Yves Ertaud. A Dynamic Programming Algorithm Applied to Omnidirectional Vision for Dense 3D Reconstruction. 2013 2nd IAPR Asian Conference on Pattern Recognition (ACPR), Nov 2013, Naha, Japan. 10.1109/ACPR.2013.170 . hal-01710409

HAL Id: hal-01710409

<https://hal.science/hal-01710409>

Submitted on 15 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A dynamic programming algorithm applied to omnidirectional vision for dense 3D reconstruction

R. Boutteau, X. Savatier, J.Y. Ertaud

Institut de Recherche en Systèmes Electroniques Embarqués (IRSEEM)

boutteau@esigelec.fr, savatier@esigelec.fr, ertaud@esigelec.fr

Abstract—In this paper, we present a dense 3D reconstruction algorithm adapted to stereoscopic omnidirectional sensors. Our main contributions are the generalization of global constraints to central systems and the use of dynamic programming to take them into account. Experimental results show the interest of our algorithm on real data.

Index Terms—omnidirectional vision, catadioptric sensor, stereovision, 3D reconstruction, dynamic programming

I. INTRODUCTION

Interest in mobile robotics have constantly been increasing in recent years, especially to replace humans in dangerous environments. These hostile environments can be natural (seabed, planets, areas of natural disasters) or linked to human action (nuclear, battlefields). For interventions in such environments, many robots have been developed, but they are generally controlled by a remote operator who receives the necessary data via sensors (ultrasonic sensors, cameras, etc.), interprets this data, and then sends commands to the robot based on these different elements. The communication between the robot and the control station raises real problems, either with wired communication (cable length, impossibility to close a hermetic door behind the robot, etc.) or wireless communication (wave propagation delay over long distances, presence of an interference system, etc.).

Solving these problems requires the design of robots capable of adapting by themselves in real-time to their environment. Robots must be able to accomplish two main tasks: to perceive their environment and be able to locate and navigate within it. Our works concern the first point, i.e. the perception of the environment of a mobile robot without *a priori* knowledge. Our research focuses on vision systems since they can get a lot of information on the observed scene in a very short time. A single image is nevertheless not sufficient to reconstruct a three-dimensional scene. This is due to the image formation process, which involves the projection of the 3D scene onto a two dimensional image plane, leading to the loss of depth information. To avoid this problem, several points of view are needed.

We describe in Section II the system we have designed to solve the problem of 3D reconstruction of the environment of a mobile robot, and its modelling. Section III is dedicated to our 3D reconstruction method which is possible without motion, thanks to the stereoscopic structure of our system. We present our main contribution, the adaptation of dynamic programming

to an omnidirectional stereoscopic system. Section IV is devoted to experimental results. We highlight the improvements of our method compared to conventional methods. Finally, in the last section, we summarize the key points discussed, and we propose future works.

II. SYSTEM OVERVIEW

Although it is possible to reconstruct the environment with a single camera, a stereoscopic sensor can produce a 3D reconstruction instantaneously (without displacement) and will give better results in dynamic scenes. For these reasons, we developed a stereoscopic system dedicated to mobile robot applications using two catadioptric sensors as shown in Figure 1. These sensors are composed of two hyperbolic mirrors and two cameras. They are placed coaxially to simplify the epipolar geometry since in this configuration, the epipolar lines correspond to the radii of the omnidirectional images.

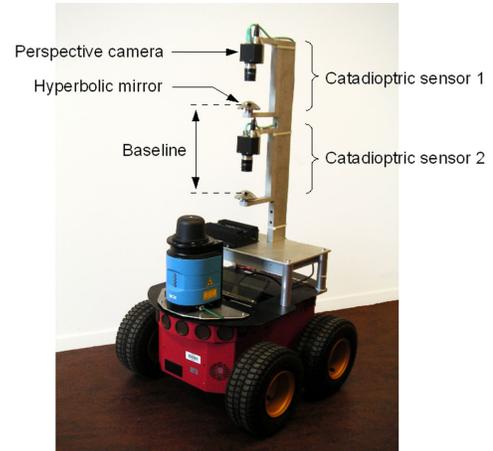


Fig. 1. View of our catadioptric stereovision sensor mounted on a Pioneer robot. Baseline is around 20cm. The overall height of the sensor is 40cm.

The method to respect the Single Viewpoint Constraint [1], the modelling and the calibration of this system are described in [2]. The model we used is the unified projection model introduced described in [3].

III. 3D DENSE RECONSTRUCTION

A. Introduction

The work presented in this section deals with 3D dense reconstruction of the environment. It aims to match a maximum

of pixels in order to have a 3D model as complete as possible. A disparity map obtained by local methods (see Fig. 7) has many artifacts that lead to a wrong 3D model if mismatches rejection techniques are not implemented. These artifacts are due to ambiguities caused by a lack of texture, by variations of illumination or by the presence of occlusions. If rejection techniques are used, the disparity map is not dense enough to provide an interpretable 3D model.

To improve matching algorithms, it is necessary to impose additional constraints called global constraints as they are applied to all matches. The cost function that will validate the matching is then evaluated on the whole image or on the epipolar line, and not only on a pixel and its neighborhood.

We first present the global constraints commonly used and illustrate them for the case of spherical sensor. Then, we expose the method we have adapted to the omnidirectional vision to take into account these constraints during the matching step.

B. Global constraints applied to omnidirectional sensors

Using similarity criteria such as correlation is generally not sufficient to match primitives unambiguously. A primitive of the first image can indeed resemble to several primitives of the second image, especially when the scene does not present a sufficient texture or when the texture is repetitive.

To overcome these ambiguities, it is necessary to take into account global constraints. There are many global constraints in the literature [4]. We will present only those providing significant improvement, i.e. the uniqueness constraint, the ordering constraint and the disparity continuity. To the best of our knowledge, these constraints have always been used for perspective cameras. Therefore, we propose to illustrate these constraints to spherical sensors.

Uniqueness constraint

The uniqueness constraint expresses the fact that a primitive of an image has at most one match in the other image and *vice versa*. Thus, in Figure 2a, the point X_{S1}^1 of the first image has only one corresponding point in the second image. This constraint implies that no pair of 3D points are aligned with the optical center of one of the sensors.

The uniqueness constraint can be justified by assuming the opacity of points, which leads to consider only the physical point closest to the camera (X_1 in Fig. 2a) if several points are aligned with the optical center.

Ordering constraint

The ordering constraint implies the conservation of the corresponding point order along two epipolar curves. Figure 2b illustrates this constraint. If three points X_1 , X_2 and X_3 are projected respectively in X_{S1}^1 , X_{S1}^2 and X_{S1}^3 onto the first image, the projections of these points into the second image will retain the same order: X_{S2}^1 , X_{S2}^2 and X_{S2}^3 .

Continuity of the disparity constraint

The continuity constraint implies that the observed scene is composed of physical objects that can be approximated by

primitives, which means that the disparity of the same object varies continuously. The discontinuities of the disparity appear at the boundaries of physical objects.

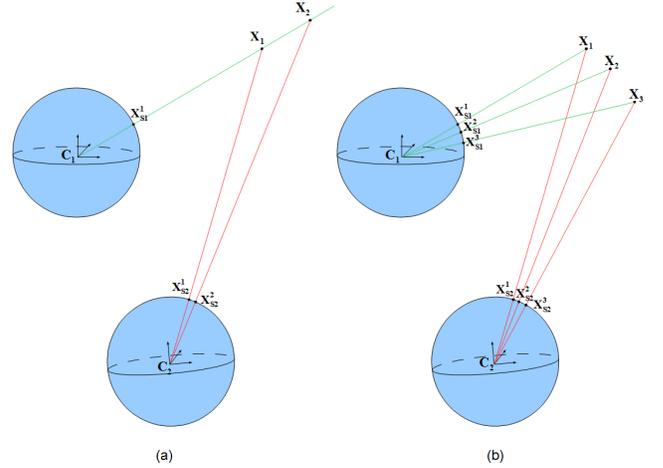


Fig. 2. Global constraints applied to omnidirectional sensors. (a) Uniqueness Constraint - (b) Ordering constraint

C. Proposed method

Dynamic programming (DP) is a well-known technique in conventional vision to take into account the global constraints in the matching step. This is an optimization method under constraints invented by Richard Bellman in 1953, which consists in deducing the optimal solution to a problem from the optimal solutions of simpler problems.

This technique can be applied to the matching problem by searching a path with minimum cost through a matrix composed of all possible matches. The matching will then consider the global constraints, since the cost function is evaluated on the whole sequence to match, and not on the primitives independently.

The first works on the application of dynamic programming to the problem of matching are those of Baker [5], followed by those of Ohta and Kanade [6]. The matching is achieved not on the pixels, but on the edges, which does not provide a dense 3D reconstruction of the environment. [7] and [8] propose to use dynamic programming to perform a dense matching, i.e. by using pixels as primitives for matching, and obtain results with fewer errors than in the case of local methods.

However, the number of works on the application of dynamic programming to omnidirectional images is very small. This can be explained by the fact that the matching must be constrained by the epipolar constraint (the problem must be bi-dimensional). The epipolar geometry of catadioptric sensors is indeed more complex than perspective cameras because the epipolar lines become epipolar conics.

In [9], they have applied dynamic programming to a catadioptric sensor using a hyperboloidal mirror in motion. As the displacement is unknown, the research must be done along the epipolar curves which are, in the general case, conics. The extraction of the pixel values along the epipolar curve

and those of their neighbors is consequently a complex stage. Interesting results are highlighted but a dense disparity map is not presented.

Lee [10] uses dynamic programming to match the vertical lines of a structured environment (doors, edges of walls, etc.). His method consists in moving a conical mirror. However, it only takes into account the case of a translation without rotation, which is not feasible with a mobile robot. Moreover, it assumes the robot's movements are known to perform the triangulation, which poses problems of accuracy when this information is obtained by odometry.

Finally, [11] provides a stereoscopic system based on a two-lobe hyperboloidal mirror. This configuration has the advantage of simplifying the search along the epipolar curves, but the results of matching using dynamic programming are presented only on very simple synthetic images since it consists only of two cubes.

We therefore propose to apply dynamic programming to our omnidirectional images by taking into account the specific configuration of our sensor that simplifies the epipolar geometry. We also use color in a particular color space: $H_1H_2H_3$ to improve performance. It has indeed been demonstrated in [12] that this color space works best for matching.

Extracting pixels of the epipolar lines

Dynamic programming can only be applied to two-dimensional problems. It is therefore necessary to restrict the search for corresponding pixels with the epipolar constraint: the problem is then reduced to the mapping of pixels of a line in the first image with those of the second image.

The structure of our stereoscopic sensor allows to determine the epipolar lines easily since they correspond to the radii of omnidirectional images. The treatment of the two images will therefore be done by scanning the image angularly and extracting the radii defined by the angle θ (see Figure 3). The angle step is determined by the resolution and/or the desired computation time.

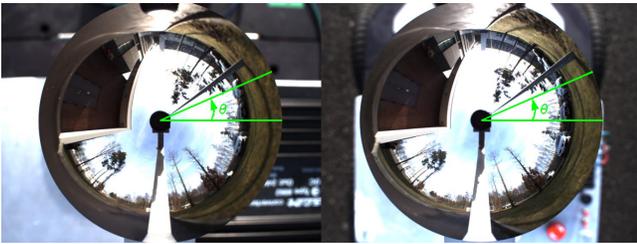


Fig. 3. Extraction of the radii in the bottom and up sensors images.

For each value of θ , the coordinates of the m pixels of the radius of the bottom sensor image and those of the n pixels of the up sensor image are determined by the Bresenham algorithm.

Building the DP matrix

The principle of dynamic programming is to find a path

of optimal cost (minimum or maximum depending on the similarity measure used) in a matrix called DP matrix (see Figure 4). A grid node is a pair of candidates which may have a match, and the optimal path consists of a sequence of nodes.

The first step is to determine the coefficients of this matrix. They are calculated from the first item $\mathbf{A}[0,0]$ to the last element $\mathbf{A}[m,n]$. The path cost is defined as the sum of the costs of the nodes in the path, the component $\mathbf{A}[i,j]$ is calculated as follows:

- the maximum of the three previous elements (left, upleft, and up) is looking for:
 $maximum = \max(\mathbf{A}[i-1,j], \mathbf{A}[i,j-1], \mathbf{A}[i-1,j-1])$
- the local cost, i.e. the correlation score of the combination of points (i,j) is added to this maximum:
 $\mathbf{A}[i,j] = maximum + \text{NCC}(\text{BottomSensorPixel}(i), \text{UpSensorPixel}(j))$

To compute the correlation score, we use the $H_1H_2H_3$ color space and the Normalized Cross Correlation (NCC) while another quicker measure could have been chosen, but at the expense of results. The filling of the DP matrix elements can be summarized by the algorithm 1.

Algorithm 1 Pseudo-code to evaluate the elements of the DP matrix.

```

 $\mathbf{A}[0,0] \leftarrow 0$ 
for  $i = 0 \rightarrow m$  do
  for  $j = 0 \rightarrow n$  do
     $maximum \leftarrow \max(\mathbf{A}[i-1,j], \mathbf{A}[i,j-1], \mathbf{A}[i-1,j-1])$ 
     $\mathbf{A}[i,j] \leftarrow maximum + \text{NCC}(\text{BottomSensorPixel}(i), \text{UpSensorPixel}(j))$ 
   $j \leftarrow j + 1$ 
  end for
   $i \leftarrow i + 1$ 
end for

```

Finding the optimal path

The search for the optimal path is the key step of dynamic programming because it is here that some of the global constraints outlined above are imposed. The optimal path is evaluated from the last element of the DP matrix $\mathbf{A}[m,n]$. For each node (i,j) of the path, we seek to determine the element among its neighbors that will achieve a maximum cost, the cost of the path being the sum of the costs of the nodes.

The ordering constraint is imposed by forcing the path to be monotonous. Thus, the search for the next element of the node (i,j) is done among the three neighbors $(i-1,j)$, $(i,j-1)$ and $(i-1,j-1)$, which prevents any backsliding. The order of the corresponding primitives will be kept along the two epipolar lines. An example of optimal path is given in Figure 4.

Computation of the disparity

The disparity is obtained by taking the difference between i and j for each node of the path. However, this disparity is not

necessarily valid since horizontal or vertical transitions tend to associate a pixel of the bottom sensor image to several pixels of the up sensor image (and *vice versa*), which is in contradiction with the uniqueness constraint. To impose the uniqueness constraint, only the diagonal transitions of the path are considered as valid. The constraint of continuity is also imposed by rejecting matches that introduce a strong gradient of disparity, i.e. long horizontal or vertical transitions. Figure 4 is an example of optimal path obtained where the valid matches are green and where the invalid ones are red. Horizontal or vertical transitions of the optimal path, i.e. invalid matches, appear when there are occlusions in one of the images.

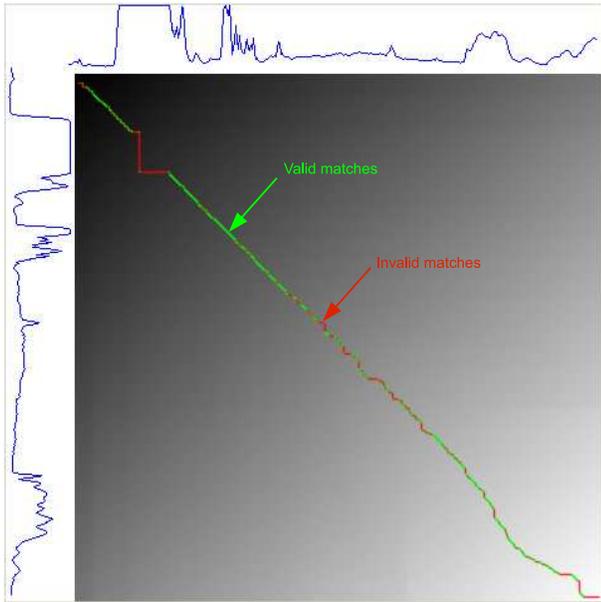


Fig. 4. An example of DP matrix. The optimal path is represented in green for valid matches and in red for invalid ones.

IV. EXPERIMENTAL RESULTS

A. Matching results

The experimental results presented in this section are intended to justify the choices we made regarding correlation. These choices were guided by the results of existing works and are consolidated by our experiments. In Figures 5 and 6, the matches are illustrated by a green line connecting the projection of a same 3D point in both images. The long green segments represent matching errors since the motion between two images is short.

Grayscale vs RGB

To illustrate the impact of the use of color in the matching process, we have compared the results of a matching in grayscale and RGB space on several kind of scenes. Figure 5 illustrates the results obtained in an indoor environment.

Among the 420 detected Harris points, only 110 points were matched in grayscale *versus* 227 points in color (after a back-correlation step [13]). The use of color considerably improves the matching process.

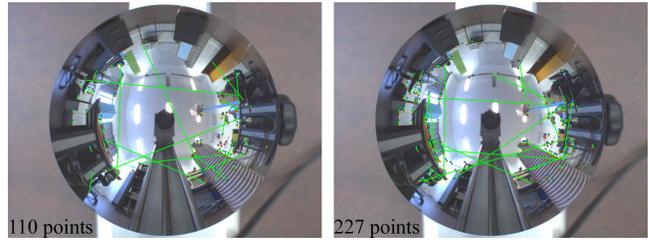


Fig. 5. Comparison of the matching using grayscale (left) and color (right). The number of correct matches is multiplied by two using color information.

Comparison between RGB and $H_1H_2H_3$ color spaces

The choice of the color space used for correlation has an impact on performance. The number of points considered as correctly matched varies from 227 using the RGB space to 221 using the $H_1H_2H_3$ color space (see Fig. 6). The six points not matched with $H_1H_2H_3$ correspond to mismatches in the RGB space. The $H_1H_2H_3$ color space is better than the RGB space since there are as many correct matches as before for fewer mismatches.

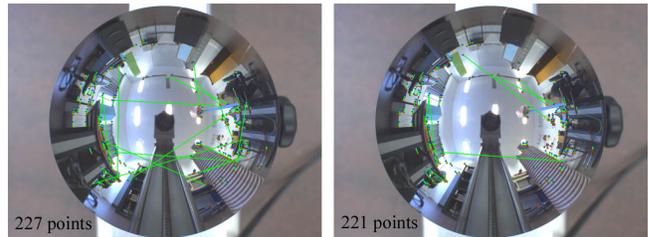


Fig. 6. Comparison of the matching in RGB (left) and in $H_1H_2H_3$ (right) color spaces. The use of the $H_1H_2H_3$ color space reduces the number of mismatches.

Rejection of mismatches by back-correlation

The ultimate objective of the matching step is to obtain the disparity map reflecting the distance between objects and the camera. The objects closest to the camera are represented by the warmer colors and more distant objects by cold colors. Figure 7 shows the disparity map obtained by a Normalized Cross Correlation using the $H_1H_2H_3$ color space, but without applying the methods of mismatches rejection. The disparity map is very dense, but also very noisy because of the presence of a lot of mismatches.

When the back-correlation is implemented, the disparity map obtained has few matching errors, as illustrated in Figure 8. The map is however less dense, leading to a 3D model composed of too few points.

Local vs global methods

Our method was evaluated on the pair of images presented in Figure 3. The environment is very diversified since there are both natural elements (trees, grass) and highly structured elements such as buildings.

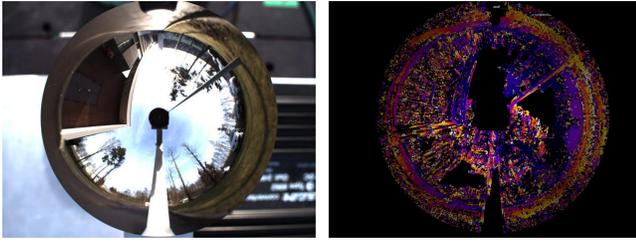


Fig. 7. Disparity map obtained with local methods without mismatches rejection. (Left) Omnidirectional image - (Right) Disparity map.

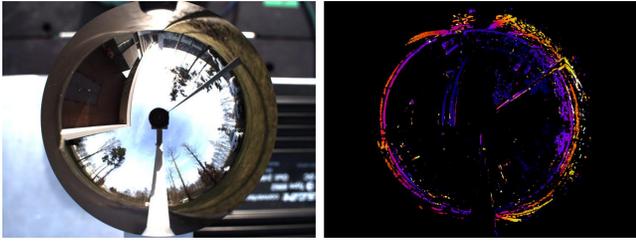


Fig. 8. Disparity map obtained by local methods with mismatches rejection. (Left) Omnidirectional image - (Right) Disparity map.

We have evaluated the contribution of dynamic programming by comparing the disparity map obtained by local methods (Fig. 9a) and imposing global constraints (Fig. 9b) by the method described above.

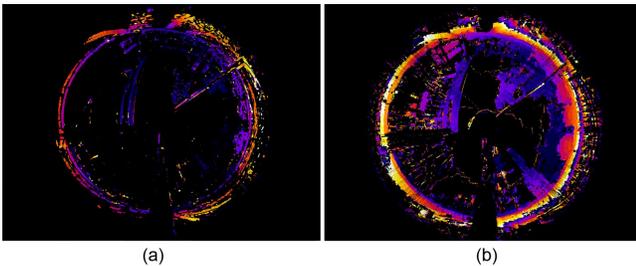


Fig. 9. Disparity maps using local (a) and global (b) methods.

The disparity map obtained by dynamic programming is far more complete than the one provided by local methods. For example, the trees highlighted in red in Figure 10 were not reconstructed with local methods. It is the same for the building structure (in green in Figure 10) even with a weak texture. The matching of the pixels from this building was made possible by dynamic programming that allows the matching of weakly textured regions bounded by remarkable edges. Some items that were already visible in the disparity map obtained with local methods, such as cars highlighted in blue in Figure 10, have surfaces more complete and are artifact-free thanks to the constraint of disparity continuity.

V. CONCLUSION AND FUTURE WORK

This article has been devoted to methods for 3D dense reconstruction of the environment without motion of the sensor. Local methods commonly used in stereovision have



Fig. 10. Highlighting of the objects of the disparity map. By using the global method, the 3D reconstruction of the building (in green), trees (in red) and cars (in blue) is more complete.

limitations, restricting their use in a large environment. As a result, we proposed a generalization of global constraints to omnidirectional sensors. The application of these constraints through dynamic programming greatly improves the density of the disparity map, while guaranteeing a relatively low mismatch rate. Moreover, we demonstrated that the use of color in the $H_1H_2H_3$ space improves the results significantly. Dense 3D reconstruction could be improved by using a correlation window in adequation with the geometry of the sensor. Indeed, the square correlation window used on classical images has no meaning on omnidirectional images. An interesting approach could be to work directly on the sphere for the matching step.

REFERENCES

- [1] S. Baker and S. Nayar, "A theory of single-viewpoint catadioptric image formation," *International Journal of Computer Vision (IJCV)*, vol. 35, no. 2, pp. 175–196, November 1999.
- [2] R. Boutteau, X. Savatier, J. Ertaud, and B. Mazari, "An omnidirectional stereoscopic system for mobile robot navigation," *Sensors and Transducers Journal, Special Issue on Robotic and Sensors Environments*, vol. 5, pp. 3–17, March 2009.
- [3] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *IEEE International Conference on Robotics and Automation (ICRA)*, April 2007, pp. 3945–3950.
- [4] M. Pollefeys, "Tutorial on 3d modeling from images," Tutorial presented at European Conference on Computer Vision, Tech. Rep., 2000.
- [5] H. Baker and T. Binford, "Depth from edge and intensity based stereo," in *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, 1981, pp. 631–636.
- [6] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 139–154, 1985.
- [7] I. Cox, S. Hingorani, and S. Rao, "A maximum likelihood stereo algorithm," *Computer Vision and Image Understanding (CVIU)*, vol. 63, no. 3, pp. 542–567, 1996.
- [8] A. Bobick and S. Intille, "Large occlusion stereo," *International Journal of Computer Vision (IJCV)*, vol. 33, no. 3, pp. 181–200, September 1999.
- [9] S. Wei, "Stereo matching of catadioptric panoramic images," Technical Report, Czech Technical University, 2000.
- [10] Y. Lee and M. Chung, "A reliable feature matching method in omnidirectional views for autonomous map generation of a mobile robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, vol. 2, 2001, pp. 895–900.
- [11] F. Correa, V. Guizilini, and J. Okamoto, "Omnidirectional stereovision system with two-lobe hyperbolic mirror for robot navigation," in *International Congress of Mechanical Engineering*, vol. 2, 2005, pp. 653–660.
- [12] S. Chambon and A. Crouzil, "Color correlation-based matching," *International Journal of Robotics and Automation*, vol. 20, no. 2, pp. 78–85, 2005.
- [13] P. Werth and S. Scherer, "A novel bidirectional framework for control and refinement of areabased correlation techniques," in *International Conference on Pattern Recognition (ICPR)*, vol. 3, 2000, pp. 730–733.