Real-Time Extraction of 3D Dynamic Environment Description Using Multiple Stereovision Sensors

Sergiu NEDEVSCHI, Radu DANESCU, Dan FRENTIU, Tiberiu MARITA, Florin ONIGA, Ciprian POCOL
Computer Science Department, Technical University of Cluj-Napoca
3400 Cluj-Napoca, ROMANIA

ABSTRACT

A method of environment description based on stereovision sensors will be presented. The 3D environment is composed of industrial objects depicted as cuboids. The objects can be stationary or moving, and a distinction should be made between these two classes. The system is structured in a distributed fashion. One sensor is composed of a pair of video cameras and an image processing device, which is able to perform real-time stereo processing. The output of one stereo sensor is a list of cuboids, describing the part of the environment that it sees. All the sensors must report the cuboids in the same coordinate system. The cuboids are communicated using a symbolic representation and a standard network communication protocol. Each sensor output is sent to a fusion computer, which assembles the complete description of the environment. The result of the fusion process is in two formats: a concise format which can be used by a remote control algorithm, and a standard 3D description format which can be used by remote visualization standard programs. Both these formats are communicated through standard networking protocols. As possible employment of the system we can enumerate: warehouse activity planning, surveillance of harbors, parking lots, etc.

Keywords: stereovision, sensor fusion, symbolic environment description, communication, distributed computation, remote control.

1. INTRODUCTION

Having a good 3D description of an environment is essential if we want to employ any kind of automated control over it. Stereovision is becoming more and more popular as a 3D measurement tool, having the advantage of being a passive method and also of providing a rich amount of 3D data. Due to the fact that a single sensor covers a limited area, and because of the presence of occlusions in the environment, the fusion of multiple sensors becomes imperative. Also, the accuracy of a sensor reading is not uniform in any point of its working range, and therefore by using multiple sensors and integrating their readings one can improve the uniformity of the reconstruction resolution over the whole working space.

For the sensor fusion algorithm there are two options to consider, namely fusion of the 3D points reconstructed by basic stereovision or fusion of the high-level objects resulted from grouping of the 3D points at sensor level. Choosing one of the available approaches determines the structure and functionality of the whole system. The point fusion approach has the advantage of algorithm simplicity and the need of a unique point grouping process, but the disadvantage of a higher communication burden between the sensors and the fusion module. The high-level object fusion algorithm requires low communication bandwidth, but the point grouping must be performed by each sensor. Our approach is a fusion of the objects, represented as cuboids.

2. DEFINITION OF THE ENVIRONMENT MODEL

The scene has associated a unique 3D coordinate system. The 3D data from all the stereovision sensors is relative to this unique coordinate system. This is ensured by proper sensor calibration.

The environment is described as a list of objects. Each sensor outputs its own list of objects, which corresponds to the sensor's view of the world. The fusion algorithm will join all object lists into a final one.

An implicit model for the object's representation was chosen. The objects will be represented as a list of eight points. We have chosen to represent the objects as: *Object = Object(LowerFacet, UpperFacet, SensorID, DynamicInformation)*, where the facet is a set of four points, *Facet = F (P1, P2, P3, P4)*. The point's structure is composed of its 3D coordinates and a confidence measure: *Point = P(X,Y,Z, Confidence)*. The first point of a facet will be the closest point to the origin of the system of coordinates, and then the following points will follow counter-clockwise. This way, each point of the object will have a unique identity, thus simplifying the fusion algorithm. This representation allows the determination of other important parameters, such as position of the center of mass, the size of the object and its orientation.

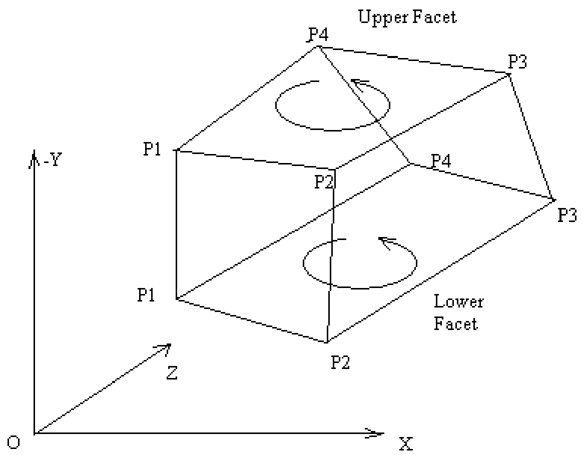


Fig. 1. Object representation

The dynamic information is $DynamicInformation = (V_x, V_y, V_z, \omega_x, \omega_y, \omega_z)$. The first three components are the components of the velocity of the object's center of mass along the coordinate axes, and the last three are the angular velocities of the object.

The same object format is used in all stages of the process, however, some components are employed only in a specific stage. The objects resulted directly from the stereovision sensor will have no valid dynamic information, only coordinates and confidences. After the sensor fusion step the coordinates will be refined and the confidences updated – this allows the cascading of more fusion steps. After the fusion step, the tracking step will compute the dynamic parameters, making the description complete.

3. THE SENSORIAL SYSTEM ARCHITECTURE AND FUNCTIONS

The architecture of the sensorial system is presented in fig. 2.

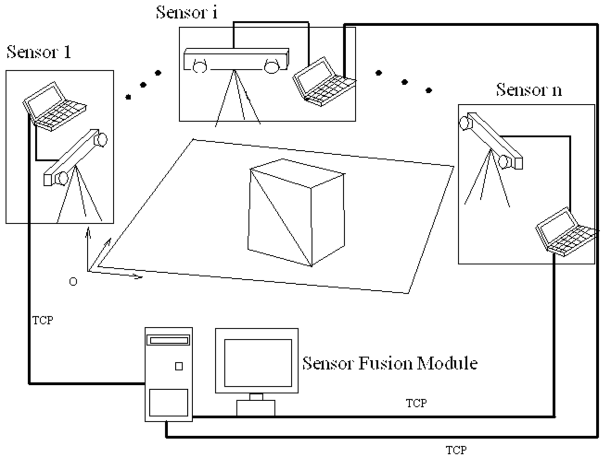


Fig. 2. The architecture of the sensorial system.

The system consists of "*n*" Stereovision Sensors linked by TCP connection to the Sensor Fusion Module (SFM). The Stereovision Sensors must be placed around the space of interest in such a way that a good coverage of the scene is accomplished. This way, each sensor has a different view of the 3D scene, and issues as hidden object facets or object occlusions are easier to treat.

A Stereovision Sensor consists of a pair of cameras, mounted on a rig, linked to its image processing computer. The image processing computer performs stereo 3D reconstruction cycles on the synchronously acquired image pairs. The reconstructed 3D points grouped into 3D objects (cuboids) represent the sensor's output.

Calibration of the Stereovision Sensor is required to completely determine the geometry of the cameras. In order to provide the results relative to the same system of coordinates, each of the sensors must be calibrated relative to the global 3D coordinate system.

The SFM has processing and communication responsibilities. The processing responsibilities consist of fusion of the sensorial objects and tracking of the fused objects. The communication responsibilities consists of handling the connections with the Stereovision Sensors and with the clients, which can be a viewing application, another SFM, etc.

The communication between SFM and the Stereovision Sensors is bi-directional. The Stereovision Sensors have to acquire simultaneously the image pairs with an acquisition rate in accordance to the dynamic characteristics of the objects of interest. The key role in this synchronization is played by the SFM, which issues this synchronization signal in the form of a multicast request for an object list. Having received this request, each sensor performs a reconstruction cycle and sends back to SFM the list of detected objects.

4. STEREOVISION SENSORS CALIBRATION

In order to reconstruct and measure the 3D environment using stereo cameras, the cameras must be calibrated. The calibration process estimates the camera's intrinsic parameters (which are related to its internal optical and geometrical characteristics) and extrinsic ones (which are related to the 3D position and orientation of the camera relative to a global world coordinate system).

The intrinsic parameters of each camera are calibrated individually. The estimated parameters are the focal length and the principal point coordinates and the lens distortions. The parameters are estimated by minimizing the projection error from multiple views of a set of control points placed on a coplanar calibration object with known geometry. For a stereo system of two cameras, the obtained intrinsic parameters can be refined by inferring the stereo information available. This is done by introducing a new constraint in the estimation process which considers also the projection error of the control points image coordinates from one image to another [1].

The extrinsic parameters of the cameras are estimated by minimizing against the extrinsic parameters the projection error for a set of 3D control points with measured coordinates in a world reference system [2,3]. For the specific setup of the current application having multiple stereovision sensors, each stereo pair of cameras is calibrated using a set of control points measured in a unique world coordinate system - the coordinate system of the scene (fig . 3).

The obtained extrinsic parameters for each camera *"j"* are a translation vector of the camera in the world coordinate system (Tj) and a rotation vector (Rj) relative to the same coordinate system. This approach in the calibration process allows us to measure the coordinates of the reconstructed 3D object in the same world coordinate system, which is essential for the sensor fusion algorithm.

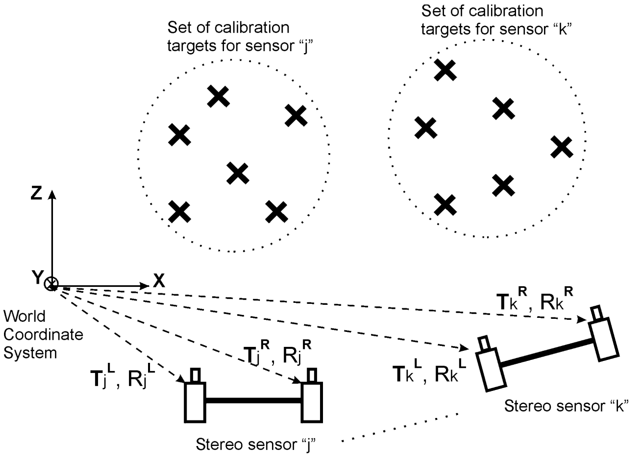


Fig. 3. Calibration setup for calibrating the extrinsic parameters.

5. STEREO 3D RECONSTRUCTION

The stereo reconstruction algorithm used is mainly based on the classical stereovision principles available in the existing literature [4]: find pairs of left-right correspondent points and map them into the 3D world using the stereo system geometry determined by calibration.

Constraints, concerning real-time response of the system and high confidence of the reconstructed points, must be used. In order to reduce the search space, only edge points of the left image are correlated to the right image points. For robust detection of the image edges, a Canny-based [5] edge detector was implemented. By focusing to the image edges, not only the response time is improved, but also the correlation task is easier, since these points are placed in non-uniform image areas. The sum of absolute differences (SAD) function [6] is used as a measure of similarity, applied on a local neighborhood. Parallel processing features of the processor are used to implement this function. For a given left image point the search is performed along the epipolar line computed from the stereo geometry.

After this step of finding correspondences, each left-right pair of points is mapped into a unique 3D point [4]. Using the camera geometry, two 3D projection rays are traced, one for each point of the pair. By computing the intersection of the two projection rays, the coordinates of the 3D point are determined.

The result of reconstruction is a set of 3D points that must be clustered into objects. The grouping is performed mainly based on the local density of the points and the vicinity criteria: a local group of points must be dense enough to be considered as candidate and two points are considered to be in the same group if they are close to each other. Both these criteria are adapted to the fact that the density of reconstructed points per object decreases with the distance (due to the perspective projection) and their positioning error increases with the same distance. When we are dealing with known objects shapes (ex. surveillance of a warehouse: containers have parallelepiped shape), additional shape-constraints can be imposed to have a better grouping. For each cluster of points, the circumscribing cuboid is built, as specified in the environment model. For each vertex of a cuboid the confidence factor is evaluated based on the density of neighboring 3D points. The orientation of each object is also inferred.

6. SENSOR FUSION ALGORITHM

Each Stereovision Sensor will provide a list of cuboids in the environment model format. When attempting to fuse the results of the sensors into a global result, we must make the difference between the case when an object is detected by only one sensor, and the case when an object is detected by two or more sensors. In the first case, the act of fusion is simply to add this object to the global result set. In the second case, the result must be a combination of the sensor readings, taking into consideration the confidence measures of each sensor.

The main simplification of the problem comes from the fact that the cuboids are defined in the same coordinate system, and therefore no geometrical transformations are necessary in order to compare their position. In order to define a fusion criterion, we must define the following measures:

Center of mass:

$$C_m = \frac{\sum_{i=1}^{4} LowerFacet.P_i + \sum_{i=1}^{4} UpperFacet.P_i}{8}$$

Approximate radius:

$$R = \frac{1}{8}\sum_{i=1}^{4}(Dist(C_m, Lowerfacet.P_i) + Dist(C_m, UpperFacet.P_i))$$

The function *Dist* can be the Manhattan distance or the Euclidean distance, depending on how much we want to balance the speed versus the quality of the match.

This way, the criterion that two objects occupy the same space (and therefore they must be joined) is:

$$Join(O_1, O_2) = (Dist(C_m(O_1) - C_m(O_2)) < \\ Max(R(O_1), R(O_2)))\ and\ (O_1.SensorID \neq O_2.SensorID)$$

This condition is a little different from the condition commonly used to check that two objects intersect (the distance between their centers of mass is less than the sum of the radii). The reason for this condition is that we are trying to find whether the objects detected from different sensors are the *same* object, and not if they share a common space.

The sensor fusion algorithm is presented bellow:

```
1. Build a list of objects from all sensors
2. For each pair of objects $O_i$ and $O_j$ do
        If ($Join\ (O_i,\ O_j)$)
                $O_k = Fusion\ (O_i,\ O_j)$
                Insert $O_3$ in the object list
                $O_k$.SensorID = $O_i$.SensorID
                Remove $O_i$ and $O_j$ from list
        End if
3. Repeat 2 until no more objects to be fused
4. Return object list
```

The fusion function will combine the corresponding vertices of the two objects into a resulting vertex, exploiting the strict ordering of the representation points imposed by the environment description model. In this way, there is no need to search for the correspondence between the points of the objects.

```
Function Fusion (Object $O_i$, Object $O_j$), returns Object $O_k$
For each F in LowerFacet, UpperFacet
        For Each P in P1, P2, P3, P4
                $O_k$.F.P = Combine ($O_i$.F.P , $O_j$.F.P)
        End For
End For
```

The combination function *Combine* can be written in two ways, each one having its reason. We can make the combination of the object's vertices as a weighted sum, using the confidence level as the weight, or we can take as valid coordinate the coordinate with the highest confidence. For the first variant, the motivation is that each observation adds some information, and should not be disregarded. For the second variant the reason is that we presume that we distribute the vision sensors in such a manner that they cover the scene as best as possible, and thus each point of one object is best seen by one of the sensors (and this is expressed by a high degree of confidence of that object point reconstructed by that particular sensor), and therefore its observation is accurate enough, and there is no need to add other information, which could be in fact measurement noise.

```
Function Combine_average (Point P1, Point P2), returns
Point P3
        P3.(X,Y,Z) = (P1.(X,Y,Z)*P1.Confidence +
        P2.(X,Y,Z)*P2.Confidence) / (P1.Confidence +
        P2.Confidence)
        P3.Confidence = P1.Confidence +
        P2.Confidence
```

```
Function Combine_maximum(Point P1, Point P2), returns
Point P3
        P3 = MaxConfidencePoint (P1, P2)
```

7. OBJECT TRACKING

Tracking is employed in order to estimate the dynamic parameters of the object. The tracking algorithm views the object slightly different than the reconstruction algorithm. The object will be recorded as the position of its center of mass, the size components along each of the axes and the rotation angles around the same axes. These components can be easily deduced from the coordinates of each object's vertex. However, we need to assume that the object is a parallelepiped, otherwise this representation is incomplete.

The position and speed of the center of mass will be tracked through a linear Kalman filter, using the uniform motion model (assumption of constant speed). The rotation angle and the angular speed are also tracked through a linear Kalman filter, assuming a uniform rotation model.

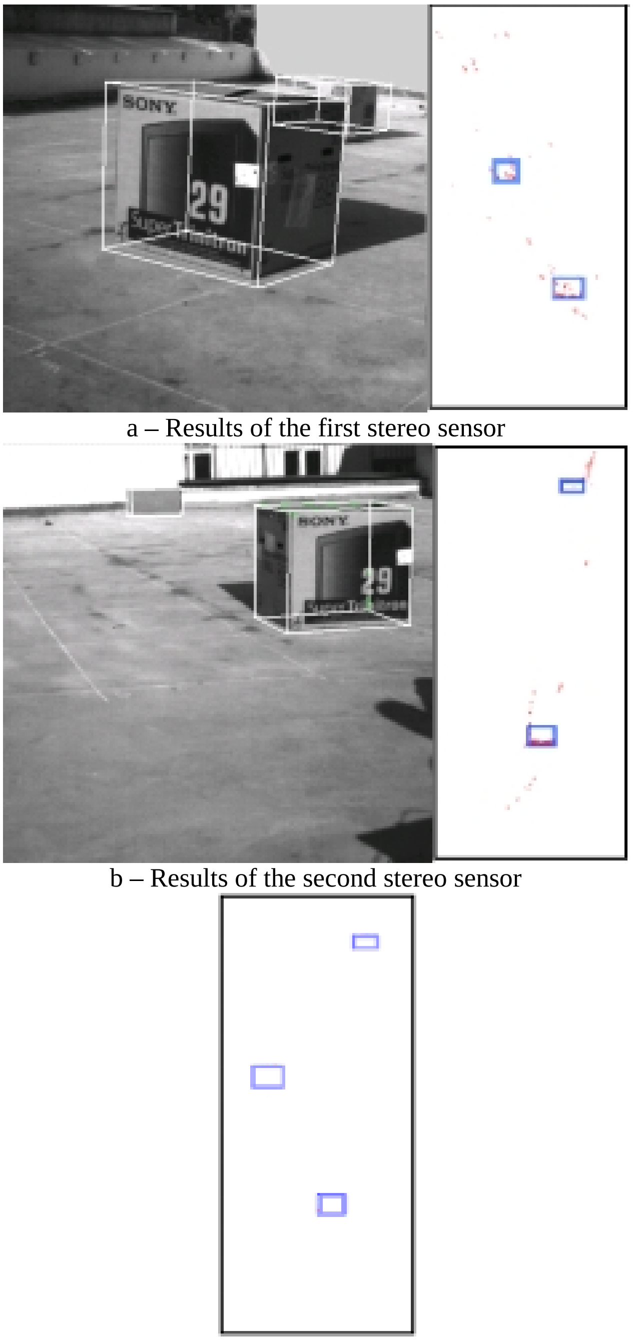
The size of the object is tracked up to a certain point, through a simple averaging of the individual measurements. After a certain number of frames, the size of the object is considered to be established, and the tracker will modify it no more.

The tracker will output the objects in the environment model format, for further processing.

8. RESULTS

For testing of the algorithm we have used two stereovision setups. The two setups were calibrated using the method described in the calibration section, using a common coordinate system. The perspective views of the scene for each stereovision sensor are presented in the left side of (fig. 4.a and 4.b). The reconstruction results for each stereovision sensor is presented as a bird-eye view of the scene in the right part of the same images, and as white cuboids projected on the original perspective image.

The sensor results were sent to the fusion unit, which integrated the data into a fused scene description. The function used to combine the objects was the one that selects the coordinate of the higher confidence. The weighted average function was also tested, but the results seem of lower quality. The results are displayed in fig. 4.c as a bird-eye view. The final result corresponds to the aim of the algorithm: combining together the scene description of different sensors and refining the measurements of each sensor against each other, in the case where the same object is viewed by more than one sensor.



a – Results of the first stereo sensor

b – Results of the second stereo sensor

c – Results of the sensor fusion

Fig. 4. Stereo reconstruction and object fusion results

9. CONCLUSIONS

A method for extracting the 3D scene description from multiple stereovision sensors has been presented. The stereovision sensors are able to perform real-time image pair processing and extract 3D points of the environment, points which they subsequently group into high-level cuboids. The communication of the cuboids to a fusion system is performed using a minimum bandwidth. Fusion is performed at cuboid level, and a complete description of the scene is obtained. The fused description has the advantage of increasing global field of view by uniting the fields of view of each sensor, and the advantage of refining the description of individual objects, if they are viewed by more than one sensor.

A point-level fusion approach is to be considered as an alternative approach, and the results compared to the current method, both in terms of reconstruction accuracy and overall time performance.

10. REFERENCES

[1] Jean-Yves Bouguet, **Camera Calibration Toolbox for Matlab**, MRL - Intel Corp., http://www.vision.caltech.edu/bouguetj/calib_doc/, 2003.

[2] S. Nedevschi, T. Marita, M. Vaida, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, D. Moga, **Camera Calibration Method for Stereo Measurements**, *Journal of Control Engineering and Applied Informatics (CEAI)*, Vol.4, No. 2, pp.21-28, 2002, Bucuresti, Romania.

[3] S. Nedevschi, T. Marita, R. Danescu, F. Oniga, D. Frentiu, C. Pocol, "**Camera Calibration Error Analysis in Stereo Measurements**", *microCAD International Scientific Conference, March 2003, Miskolc, Hungary*, pp. 51-56.

[4] Trucco E., Verri A, **Introductory techniques for 3D Computer Vision**, New Jersey: Prentice Hall, 1998.

[5] J. Ramesh, R. Kasturi, B. G. Schunk, **Machine Vision**, New York: McGraw-Hill Inc., 1996.

[6] Todd A. Williamson, **A High-Performance Stereo Vision System for Obstacle Detection**, Ph.D. Thesis. Carnegie Mellon Technical Report, September 1998