3D Motion Tracking Based on Probabilistic Volumetric Reconstruction and Optical Flow

Gisele M. Simas*, Guilherme P. Fickel*, Lucas Novelo*, Rodrigo A. de Bem* and Silvia S. C. Botelho*
Center for Computational Science C^3
Federal University of Rio Grande - FURG
Rio Grande - RS - Brazil
Email: gisele_simas@yahoo.com.br, guilhermefickel@gmail.com, lucasnovelo@gmail.com, rodrigo.bem@gmail.com, silviacb@furg.br

***Abstract*—This paper proposes a method for motion tracking of objects without a pre-defined shape, the main aspect of this method is the use of a probabilistic volumetric reconstruction that incorporates motion information. First, a volumetric reconstruction of the objects of interest is obtained by the 3D Probabilistic Occupancy Grid method, which was recently proposed for to be applied in environments sensed by multiple cameras. Then, we originally propose to add Optical Flow information to this reconstruction. Next, a method similar to the Expectation-Maximization (EM) algorithm is used to identify and track the body parts of objects of interest. It was noted that the proposed information of velocity vector fields are a good option to improve the perception of motion in 3D reconstruction, providing the best results in the tracking.**

***Keywords*-probabilistic volumetric reconstruction; optical flow; motion tracking.**

I. Introduction

Several investigations have been undertaken to develop tracking systems capable to provide robust information about the motion of people and objects through methods based on visual, mechanical, acoustic or magnetic equipment [1]. This interest stems from the numerous applications that are made possible by such a task, for example: human-machine interface, teleoperation, anthropological studies, virtual reality, entertainment and surveillance.

In this scope, the scientific community has given special attention to methods of motion tracking from computer vision [2], [3], [4]. More specifically, the visual motion tracking methods that do not employ optical marks have a number of advantages over other methods, such as: a) they are not intrusive (do not require extra equipment attached to the objects of interest); b) there is no need of redundant equipment to tracking multiple objects; c) these methods allow the observation of physical characteristics of objects (such as color and texture); d) they have lower costs than other methods.

However, the tracking of objects using only 2D images is a difficult problem to be addressed given the complex nature of 3D motion and loss of information in the images due to the restriction of two-dimensional space. A further complication is added when there are multiple moving objects in the same scene, because they may appear superimposed on the images. Moreover, changes in the observed colors may occur due to lighting variations and sensory uncertainties.

Thus, to minimize such problems, we can use multiple cameras placed around the environment in which the objects of interest moves. Then, from the images captured by different cameras, we can obtain a volumetric reconstruction [5], [3], [4], which consists in the locating and determining of the volume of the objects of interest. The volumetric reconstruction is used to overcome the limitations of the analysis of 3D motion in 2D space, such as ambiguities and self-occlusions.

In this paper, the volumetric reconstruction is obtained using the Probabilistic Occupancy Grid technique, which was recently applied by [6] in the environment monitoring using multiple cameras. This technique allows to obtain a volumetric reconstruction more robust than traditional Shape-from-Silhuette methods, which analyze each image individually. This is due to the fact that the Probabilistic Occupancy Grid method employs Bayesian inference [7] to allow simultaneous evaluation of information from all cameras.

Considering the benefits brought by this 3D reconstruction technique, we decided to study its use in motion tracking. Knowing that the aggregation of more information to the 3D probabilistic grid (besides color) is possible and that this can assist in obtaining further information in the 3D reconstruction [6]; we propose a method to aggregate information from optical flow to the grid in order to improve the perception we have about the motion of objects in the volumetric reconstruction.

Next, a method similar to the Expectation-Maximization (EM) algorithm in order to, using the volumetric reconstruction, identify the 3D positioning of the objects of interest and then track it over time. We employ a simple representation model of this object, which together with the velocity fields proposed, allows the motion tracking method to be applied to objects of different shapes (people or other objects composed of rigid parts).

Therefore, this paper proposes a method for markerless

motion tracking based on multiple cameras, whose main differentials are:

- The volumetric reconstruction used in this paper is obtained by the Probabilistic Occupancy Grid method [6]. This technique was minimally explored in reconstruction of environments monitored by multiple cameras and provides more robust results than traditional methods, especially when the background color of the scene is similar to the color found in the objects of interest. Additionally, information about the movement of objects, obtained through the technique of optical flow, is added to the volumetric reconstruction;
- The motion tracking method proposed is applicable to objects of different shapes (people or other objects composed of rigid parts), this feature was made possible by employing a simple representation model of these objects and by the velocity fields aggregated to the reconstruction.

This paper is organized as follows. Section II gives a summary of the proposed architecture. The subsequent sections III, IV, and V are dedicated to system components. Finally, Section VI shows a set of tests associated with the human motion experiments and Section VII concludes the paper.

II. Architecture of the Proposed Method

In a broader context, we can say that a visual tracking system consists basically of three components: I) *observation model* - the model that defines how the objects of interest are observed; II) *representation model* - way in which the objects are represented in the tracking system; III) *tracking algorithm* - an algorithm that updates the representation model state over time, using the observation model information.

In this paper, the observation model used (Section III) is composed of: optical flow obtained by the Lucas Kanade method [8], and volumetric reconstruction obtained by the Probabilistic Occupancy Grid technique. The representation model (Section IV) uses Gaussian blobs to represent the body parts of the objects of interest. Then, the objects are tracked by a method similar to the Expectation-Maximization (EM) algorithm [5] (Section V).

Figure 1 presents the architecture of the proposed method. The system receives as input a sequence of 2D images captured by multiple synchronized cameras arranged around the environment in which the objects of interest is moving.

1) Observation Model: First, before the system starts to track the movement of objects, a set of background images captured by different cameras is used to construct a model of the background scene. This background model serves as input to the Probabilistic Occupancy Grid method.

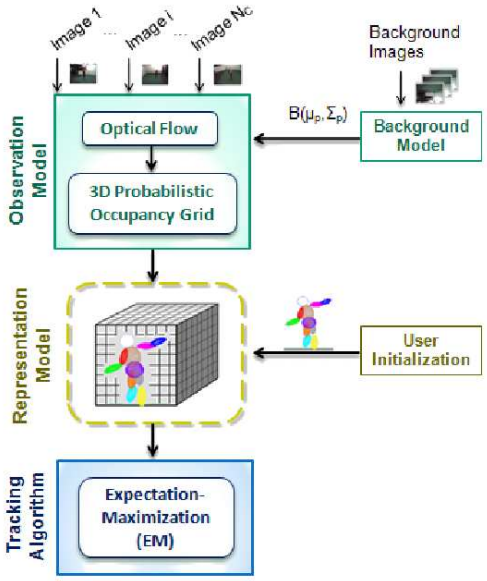

Figure 1. The architecture of the 3D visual tracking approach.

Then, at each new time instant, the Probabilistic Occupancy Grid method receives an image for each camera and build an occupancy grid. Through this grid, we can verify which volume elements (voxels) are more likely to be occupied by an object of interest. We also added information from optical flow to the grid: for each voxel is assigned a 3D velocity vector.

2) Representation Model: Early in the system, the user initializes the representation model using the ocuppancy grid corresponding to the first time instant: the blobs are positioned correctly according to the body parts which they represent. In the following time instants, this representation model is updated according to the tracking algorithm.

3) Tracking Algorithm: The tracking loop is defined by a method similar to the Expectation-Maximization (EM) algorithm [5], composed of two steps. In the first step (Expectation), each voxel of the grid is associated with a blob of the representation model. Following in the Maximization step, each of the blobs is updated according to the associations made in Expectation step. This updated representation model serves as input to the Expectation step of the next iteration of the tracking loop.

In the following sections, we detail the three components of the proposed system: the observation model, representation model and tracking algorithm.

III. Observation Model

The problem addressed in this section is the probabilistic volumetric reconstruction obtained from multiple cameras

and the optical flow calculated in two ways: I) directly in 3D space, using as input the actual reconstruction volume; and II) first in 2D space, using as input the images from different cameras, and after doing the fusion of this information in 3D space.

A. Volumetric Reconstruction

The reconstructing volumetric of environments monitored by multiple cameras is often done through the use of binary subtraction of background, analyzing each image individually, as it is done in traditional methods of Shape-from-Silhuette. However, this treatment can dramatically change the 3D perception that we would have if we observed all the images together, intuitively, the knowledge of all the images simultaneously conveys more information than the knowledge of just one picture [6].

In this context, Franco [6] proposed to calculate the fusion of the all information of the images in 3D space before to perform evaluations on each image individually. For this, Franco used the Probabilistic Occupancy Grid method. This technique has been widely used in the robotics community to represent the robot navigation environment monitored by depth sensors and by orientation measures [9]. Franco [6] then proposed to extend the concept of Grid Occupancy sensors based on images.

This technique, then, emerges as a way to get a reconstruction more robust, overcoming problems of variations in brightness and color similarity between the background and the objects of interest. Thus, it avoids the occurrence of noise and incomplete reconstructions, which occur more frequently in traditional methods of Shape-from-Silhuette.

In the 3D Probabilistic Occupancy Grid method, the images obtained by multiple synchronized cameras are unified into a occupancy grid. Each pixel of the camera is treated as a sensor susceptible to statistical uncertainties. The problem is then treated as a Bayesian estimation [7]. The 3D space is discretized into volume elements, named voxels, as illustrated in Figure 2, and for each voxel is calculated the probability of being occupied by an object of interest.

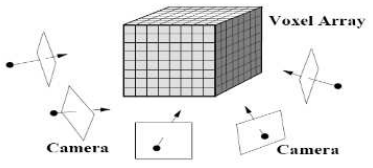

Figure 2. 3D space discretized into voxels.

1) Model background: First, it is necessary to build a model of a background scene free of moving objects. So we bought a set of images of the scene free of objects. From these images, we build a statistical model of background. We can find different models in the literature. In this paper, we employ the model presented by [10]. Typically, each pixel p is modeled by a Gaussian distribution represented by a vector of average color μ_p and a covariance matrix Σ_p. Since $D = 3$ a dimension of the Gaussian, each pixel will have the following probability of belonging to the background:

$$N(\mu, \Sigma) \cong \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\Sigma|}} exp\left\{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T\right\}. \quad (1)$$

2) Bayesian inference: Inference calculations were performed according to the method proposed by [6]. Below, we present the central idea of the method, details can be found in [6], [11].

This method takes as input: i. the background statistical model of the scene, ii. a sequence of images from multiple cameras with the objects of interest, and iii. the calibration matrix of each camera. First, from the Gaussian formula 1, we can calculate the probability of each pixel p of each camera i belongs to the background scene or to the objects of interest: how much greater $N(\mu_p, \Sigma_p)$, greater will be the probability of this pixel belongs to the background.

Next, the whole grid is covered, and each voxel V is analyzed separately. The calibration matrix M_{Ci} of each camera i is used to map the 3D coordinates of each voxel V for the 2D image plane of this camera. That is, we multiply M_{Ci} (calibration matrix 3×4 of camera i) by $\vec{P_G^h} = \{x, y, z, 1\}^T$ (3D position in homogeneous coordinates of the voxel V) and we obtain $\vec{P_{Ii}^h} = \{x_i, y_i, w_i\}^T$ (2D homogeneous coordinates of the voxel projection in the camera image i):

$$\vec{P_{Ii}^h} = M_{Ci} * \vec{P_G^h}. \quad (2)$$

If the projection $\vec{P_{Ii}^h}$, calculated above, is within the boundaries of the image of this camera, the probability value of the pixel corresponding to the position $\vec{P_{Ii}^h}$ is used to compute the probability of the voxel V belongs to the background scene.

Note that, for reasons of computational efficiency, this mapping of the 3D coordinates of each voxel V for the 2D space of each camera i can be done only once at the beginning of processing, being stored for use in the next instants of time. This mapping information also will be required on obtaining optical flow by the method of fusion of 2D optical flow in 3D space (Section III-B).

At this point, we have for each voxel V a set of probabilities of the pixels related to their projections in different cameras. This information is then unified through Bayesian inference [7], in order to get, at the end of the process, a probability value for each voxel corresponding to the probability of it be occupied. Next, we can then define a threshold probability: all voxels whose probability is above this threshold are considered as belonging to the volume of the objects of interest.

B. Optical Flow

The optical flow is the 2D distribution of the apparent speed of patterns in motion in the image plane [12]. That is, the optical flow field consists of a dense velocity field, where each pixel in the image plane is associated with a single velocity vector [13].

Methods based on optical flow have been frequently used in literature for the motion analysis in both 2D [13] and 3D spaces [14]. However, at this moment, no paper was found using optical flow in conjunction with the 3D Probabilistic Occupancy Grid technique.

We consider the optical flow may be an important source of information about the movement to be analyzed. Therefore, we propose to aggregate information from optical flow to the 3D Occupancy Grid, analyzing two different ways of accomplishing this task: I) directly in 3D space, and II) first obtaining the flow to 2D images from all cameras and then unify this information into a 3D vector for each voxel. These two methods are described below. In both of them we employ a post-processing stage to reduce noise. A comparison between these methods will be presented later in the Section VI.

1) Optical Flow 3D - Extension of the Lucas-Kanade method: The 3D optical flow is an extension of 2D optical flow: instead of dealing with a image sequence (all pixels) and get an answer as 2D velocity field, a sequence of voxel sets are used, and a 3D velocity field is obtained.

Let (x, y, z, t) be the location of a voxel, $I(x, y, z, t)$ its intensity, and δx, δy and δz the voxel moving in a δt, we can write:

$$I(x, y, z, t) = I(x + \delta x, y + \delta y, z + \delta z, t + \delta t). \tag{3}$$

Through the assumption that the voxel intensity is conserved over time, $dI(x, y, z, t)/dt = 0$, and using Taylor series, we obtain:

$$\begin{gathered} I(x + \delta x, y + \delta y, z + \delta z, t + \delta t) = \\ I(x, y, z, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial z}\delta z + HOT, \end{gathered} \tag{4}$$

where HOT represents higher order terms (High Order Terms) ignored. By this equation, we can see that:

$$\begin{gathered} \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial z}\delta z = \frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial z}\frac{\delta z}{\delta t} = 0 \\ \frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial z}V_z = 0, \end{gathered} \tag{5}$$

where V_x, V_y, and V_z represent the velocity at x, y and z of the voxel intensity $I(x, y, z, t)$. Considering $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, $\frac{\partial I}{\partial z}$, and $\frac{\partial I}{\partial t}$ respectively as I_x, I_y, I_z, and I_t we can write a more generally and succinctly:

$$I_x V_x + I_y V_y + I_z V_z = -T_t. \tag{6}$$

Assuming that the velocity of a particular voxel is approximately the same of its close neighbors, we can analyze a window $n = m \times m \times m$ around this voxel, to obtain a set of equations:

$$\begin{bmatrix} I_{x1} & I_{y1} & I_{z1} \\ I_{x2} & I_{y2} & I_{z2} \\ \vdots & \vdots & \vdots \\ I_{xn} & I_{yn} & I_{zn} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \vdots \\ -I_{tn} \end{bmatrix}. \tag{7}$$

Then, using the Least Squares method [15], it is possible to calculate for each voxel of the grid, a 3D vector velocity $\{V_x, V_y, V_z\}$, where $\sum$ represents $\sum_{i=1}^{n}$:

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} \sum I_{xi}^2 & \sum I_{xi}I_{yi} & \sum I_{xi}I_{zi} \\ \sum I_{xi}I_{yi} & \sum I_{yi}^2 & \sum I_{yi}I_{zi} \\ \sum I_{xi}I_{zi} & \sum I_{xi}I_{zi} & \sum I_{zi}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum I_t \\ \sum I_t \\ \sum I_t \end{bmatrix}. \tag{8}$$

2) Fusion of 2D Optical Flow in 3D Space: This method can be divided into two stages: obtaining the optical flow in 2D space and fuse it in the 3D space.

First, the 2D optical flow is obtained in 2D images, through the Lucas-Kanade algorithm [8].

Following, we use the mapping of the 3D coordinate of each voxel V for the 2D space image from each camera i ($\vec{P_{Ii}^h} = M_{Ci} * \vec{P_G^h}$) (see Section III-A). Since each pixel has a flow vector, at the end of the process, each voxel V will be referenced with N_{C_V} 2D flow vectors $\vec{f_{Ii}}$ with $i = 1..N_{C_V}$, where N_{C_V} is the number of cameras in which this voxel can be projected.

Next, we do the opposite, we project the N_{C_V} 2D vector flow, relative to the voxel V, for the 3D space:

$$\vec{f_{Gi}^h} = pseudo\_inverse(M_Pi) * \vec{f_{Ii}^h}, \tag{9}$$

where $\vec{f_{Gi}^h}$ is the flow vector in 3D homogeneous coordinates in space of the grid, obtained from $\vec{f_{Ii}^h}$; $pseudo\_inverse(M_{Ci})$ is the pseudo-inverse of the calibration matrix of camera i (M_{Ci} is not a square matrix); and $\vec{f_{Ii}^h}$ is the 2D optical flow vector of the camera i in homogeneous coordinates.

Then, we calculated an average vector of $\vec{f_{Gi}}$ vectors of 3D flow ($i = 1..N_{C_V}$):

$$\vec{f_R} = \frac{\sum_{i=1}^{N_{C_V}} \vec{f_{Gi}}}{N_{C_V}}. \tag{10}$$

The result $\vec{f_R}$ of this average will be the value of the 3D velocity of the voxel in question. Other methods could be used to calculate the resulting velocity, such as a weighted average with weights according to the influence of each camera. We could also use some metric to evaluate which cameras should have greater importance in the formation of 3D optical flow, but such treatment could produce inadequate results if not encourage similarly at least three cameras linearly independent.

3) Post-processing: Having a flow vector for each voxel, we use post-processing to decrease noise errors. For each voxel, we analyze the flow vectors of a window around the same voxels. The x, y, and z coordinates of the vector flow of voxels contained in this window are sorted separately and the coordinates medians x_M, y_M, and z_M are used as the coordinates of the final flow vector of this voxel.

IV. Model Representation

The tracking of an object involves finding its global position as well as the relative position between each part of the object in each frame of video sequence. In order that our method could be applicable to objects of different shapes, we do not adopt specific models of body of a certain type of object. Moreover, we do not impose kinematic constraints between body parts of an object, because performing this is very difficult when we do not know the type of object to be tracked: the body structure of the objects should be learned during the tracking, and this has a number of complications concerning the compromise between flexibility and robustness of the method (for example, when we should allow a motion that not yet was observed / learned?)

Therefore, we adopt a simple model representation based on Gaussians. Articulated objects, like the human body, are mostly composed of rigid parts, which individually do not show significant changes in their shapes. On this assumption, in this paper, each part of the object is represented by a Gaussian model, named Blob, in which we associate color, position and movement information.

1) Position and Color: With reference to spatial information, a blob is often represented by an ellipsoidal shape, and its surface is defined by the standard deviation around the mean value of position. Similarly, the color information is modeled by a mean and a variance. Therefore, a blob is a Gaussian distribution with six dimensions (the D-dimensional Gaussians are represented by the equation 1 presented in Section III-A).

2) Model Initialization: An initialization procedure is necessary in order to provide an initial estimate of the parameters of the blobs for the tracking algorithm. In this study, we used a manual initialization, in which a user informs for each blob B: a) two extreme points of its main axis of variation, through these points, we can estimate the average value of position and direction of main axis of variation of this blob; b) three values $\sigma_x, \sigma_y, \sigma_z$, that inform the standard deviations on each of the three axes of variation of blob B.

Let R be the matrix of rotation of the blob B in relation to the axes X, Y, Z of the scene, the covariance matrix of position Σ_X of this blob can be calculated by:

$$\Sigma_X = R \cdot \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix} \cdot R^T. \tag{11}$$

To obtain the matrix R, we calculate the angles ω, φ, and κ between the main axis variation of the blob and the axes X, Y, Z of scene (Euler angles), as shown in Figure 3.

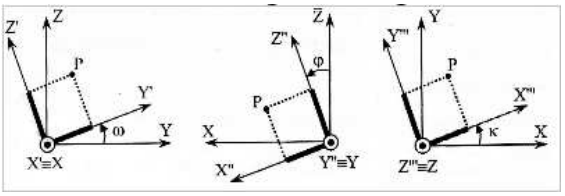

Figure 3. Euler angles - for determination of rotation matrix R.

From these angles and considering that the initial rotation is given by $R_X(\omega)$, a secondary by $R_Y(\varphi)$ and a final given by $R_Z(\kappa)$, the resulting rotation matrix R can be calculated as follows [16], where s represents the sine and c, the cosine:

$$\omega = arctg\left(\frac{z_b - z_a}{y_b - y_a}\right) \varphi = arctg\left(\frac{x_b - x_a}{z_b - z_a}\right)$$
$$\kappa = arctg\left(\frac{y_b - y_a}{x_b - x_a}\right),$$

$$R = \begin{pmatrix} c\varphi c\kappa & s\omega s\varphi c\kappa + c\omega s\kappa & -c\omega s\varphi c\kappa + s\omega s\kappa \\ -c\varphi s\kappa & -s\omega s\varphi s\kappa + c\omega s\kappa & c\omega s\varphi s\kappa + s\omega c\kappa \\ s\varphi & -s\omega c\varphi & c\omega c\varphi \end{pmatrix}. \tag{12}$$

3) Adding motion information to the representation model - Optical Flow: So far, we have associated with each voxel V at time t, a position in 3D space $\vec{X}_{(t)} = \{x, y, z\}^T$ and its optical flow vector $\vec{f}_{(t-1,t)} = \{dx, dy, dz\}^T$ (calculated from the frames relating to the times $t-1$ and t).

Then, using the optical flow vector $\vec{f}_{(t-1,t)}$, which represents the velocity of the voxel at time t, we can estimate the next position $\vec{X}e_{(t+1)} = \{xe, ye, ze\}$, that this voxel should occupy at time $t+1$ (whereas the motion of the voxel remains constant between t to $t+1$):

$$\vec{X}e_{(t+1)} = \vec{X}_{(t)} + \vec{f}_{(t-1,t)}. \tag{13}$$

Moreover, knowing that all voxels of a certain rigid body part of an object of interest (of a specific blob) should be submitted to the same motion, we estimate a transformation matrix H [17] for each blob, which map a linear transformation of rotation and translation of the set of voxel from a blob at time t for the next instant $t+1$.

Let A be the matrix $(4 \times N_V)$ with positions $\vec{X}^h_{(t)}$ in homogeneous coordinates of all N_V voxels of a given blob and let Ae be the matrix $(4 \times N_V)$ with positions $\vec{X}e^h_{(t+1)}$ estimates for the next instant of time, the matrix H (4×4) can be calculated as follows:

$$H \cdot A = Ae \rightarrow H = Ae \cdot pseudo\_inverse(A). \tag{14}$$

V. Motion Tracking

The motion tracking is performed by a similar method to the Expectation-Maximization (EM) algorithm [5]: the traditional EM algorithm was modified to treat motion information and incorporate Euclidean Distance. The method is composed of the following two steps (executed repeatedly):

1) Expectation: In this step, we determine which volume elements (voxels) belong to each blob. For this task, we use the blob parameters estimated in the previous iterations of EM or initialized by the user (in the first iteration of EM).

Thus, for each voxel V, we compute the distance from each blob B, then we assign this voxel to the nearest blob. The distance value used is a balance of three parameters:

$$D(V,B) = k_1 \cdot D_{Euc} + k_2 \cdot D_{Prob} + k_3 \cdot D_{Mov}, \quad (15)$$

where k_1, k_2, and k_3 are constant weighting of the different parameters:

I - Euclidean Distance D_{Euc} is the Euclidean distance between the position of voxel V and the center (mean of position) of the blob B.

II - Probabilistic Distance D_{Prob} is a distance parameter that considers the probability of voxel V belongs to the Gaussian blob B. As reported in Section IV, the blob B is represented by a Gaussian with mean vector μ_B and covariance matrix Σ_B. The voxel V, in turn, is associated with a position vector $\vec{X}$ and colors $\left\{\vec{C}^1, \cdots, \vec{C}^{mc}, \cdots, \vec{C}^{N_{C_V}}\right\}$ related to N_{C_V} cameras in which the voxel V can be projected. To calculate the distance D_{Prob}, we use only the color C^{mc} that maximizes the probability of the voxel V belongs to blob B:

$$mc = arg_{i=1\cdots N_C} min(C_V^i - \mu_C)\Sigma_C^{-1}(C_V^i - \mu_C)^T. \quad (16)$$

Then, saying that the voxel V is represented by the vector $\vec{x_v} = \left\{\vec{X_V}, \vec{C_V^{mc}}\right\}$, we have:

$$\begin{aligned} P(V|B) &= \frac{1}{(2\pi)^3\sqrt{|\Sigma_B|}} \cdot e^{-\frac{1}{2}(x_V-\mu_B)\Sigma^{-1}(x_V-\mu_B)^T} \\ P(V|B) &= \frac{1}{(2\pi)^3\sqrt{|\Sigma_B|}} \cdot e^{-\frac{1}{2}D_M(V,B)}, \end{aligned} \quad (17)$$

where $D_M(V,B)$ is the Mahalanobis distance between the voxel V and blob B. Assuming that there is no dependence between the color and position informations, we can simplify $D_M(V,B)$, where μ_X, Σ_X refer to the position and μ_C, Σ_C refer to the color:

$$\begin{aligned} D_M(V,B) = (X_V - \mu_X)\Sigma_X^{-1}(V_X - \mu_X)^T + \\ (C_V^{mc} - \mu_C)\Sigma_C^{-1}(C_V^{mc} - \mu_C)^T. \end{aligned} \quad (18)$$

A standard optimization is to compare the logarithms of probabilities rather than the actual probability. The maximum likelihood is also the maximum log-likelihood function:

$$logP(V|B) = -3log(2\pi) - \frac{1}{2}log|\Sigma_j| - \frac{1}{2}D_M(V,B). \quad (19)$$

Neglecting constant terms and multiplicative factors, we obtain the value that we call D_{Prob}, whose minimization is equivalent to maximizing the original likelihood function:

$$D_P = log|\Sigma_j| + D_M(V,B). \quad (20)$$

Minimize D_{Prob} depends mainly on the Mahalanobis distance $D_M()$. The term $log|\Sigma_j|$ is constant for each blob and uses the encoded variance matrix Σ_B to favor the smaller blobs, providing to these blobs a more likely chance to be chosen by voxels. This term is useful when two blobs are very similar (in position and color), because if only $D_M()$ was used, the blob with the largest variance would always have greater advantage in being chosen.

III - Movement Distance D_{Mov} is a measure of distance between the movements expected for the voxel V and for the blob B, that is, this parameter compares the optical flow vector $\vec{f}_{(t-1,t)}$ of the voxel V with the transformation matrix H of the blob B.

First, let $\vec{X}_{(t)}$ be the position of voxel V at time t, we estimate the next position $\vec{X}e_{(t+1)}$ that voxel V should occupy in $t+1$, by analysis of the optical flow vector $\vec{f}_{(t-1,t)}$ (as described in Section IV): $\vec{X}e_{(t+1)} = \vec{X}_{(t)} + \vec{f}_{(t-1,t)}$.

Next, we use the transformation matrix H of blob B to estimate the next position $\vec{X}e_{B(t+1)}$ that the voxel V should occupy in $t+1$, if it belongs to blob B:

$$\vec{X}e^h_{B(t+1)} = H \cdot \vec{X}^h_{(t)}, \quad (21)$$

where $\vec{X}e^h_{B(t+1)}$ and $\vec{X}^h_{(t)}$ are respectively the positions $\vec{X}e_{B(t+1)}$ and $\vec{X}_{(t)}$ in homogeneous coordinates.

Then we can calculate the distance parameter D_{Mov} between the movements expected for the blob B and for the voxel V given by the Euclidean distance between $\vec{X}e_{B(t+1)}$ e $\vec{X}e_{(t+1)}$.

2) Maximization: In this step, we calculate new values for the mean vector μ_B and for the covariance matrix Σ_B of each blob B, using the information from all voxels assigned to the blob B (during the Expectation step). Next, the updated parameters are then used as initial estimate for the Expectation step of the next instant of time of the image sequence.

VI. Results

A software program was implemented to allow the use of the presented methods in the analysis of video sequences acquired in the public repository [18]. An example of the images of the sequence analyzed is shown in Figure 4.

Tests were realized to allow the comparison between the volumetric reconstructions obtained by the traditional method of Shape-from-Silhouette and by the Probabilistic Occupancy Grid method. Examples of these results are shown in Figure 5.

Figure 4. Sample images from two different cameras.

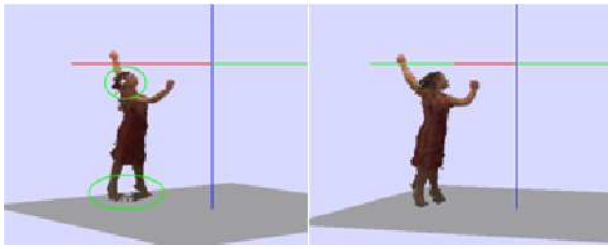

Figure 5. Volumetric reconstruction. L: traditional method Shape-from-Silhouette (failures into green circles). R: Probabilistic Occupancy Grid method.

Figure 6 shows a comparison between the two methods used for obtain 3D optical flow: the Lucas-Kanade method extended to the 3D space, and the proposed method for merging the 2D optical flow in 3D space.

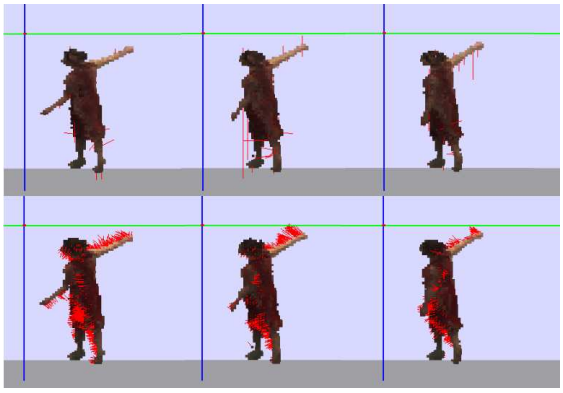

Figure 6. Comparison of the methods used to determine the 3D velocities. Top: Lucas-Kanade algorithm extended to 3D space. Bottom: Proposed method to perform the fusion of the 2D optical flow vectors in 3D space.

Through this figure, we can see that the Lucas-Kanade 3D algorithm has very low precision compared to the solution proposed merging the 2D optical flow in 3D space. This is due to the fact that the 3D optical flow algorithms are very susceptible to noise and require images with a high level of detailing. On the other hand, in the method that performs the fusion of 2D optical flow vectors, the resulting 3D velocities are obtained from images whose resolution is far greater than the grid of voxels. Another important factor is that, for each voxel, the results of 2D optical flow of N_C available cameras are used, so any noise that may result from the 2D optical flow in some camera is attenuated.

Figure 7 shows the initialized colors for the different body parts of the tracked person. Each voxel set of a single color belongs to a single blob of the representation model.

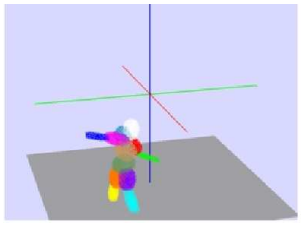

Figure 7. Initialization - The different colors indicate different body parts.

Figure 8 - Column 1 shows the reconstruction and optical flow at different times.

Figure 8 - Column 2 shows the tracking results obtained without the use of optical flow aggregated to the reconstruction, that is, without the use of motion distance parameter D_{Mov} in the Expectation-Maximization method. In this figure, we note that the colors of body parts are different of the identified colors in the initialization, the blobs have switched places, not following the movement of the person (check the blobs on the head and trunk). We can also see, in these frames, that the orange and purple blobs switched places (legs), this occurs more often in body parts with similar color and very close to each other.

The tracking results using optical flow information aggregated to the reconstruction are shown in Figure 8 - Column 3. As can be seen, in the method using optical flow, the blobs followed the movement of their respective body parts. We note that this task was possible even without the determination of joint positions or any definitions of relative positions between the blobs.

VII. Conclusions

This paper presented an approach to markerless motion tracking based on multiple cameras that can be applied to objects of different shapes. The proposed method uses, as observation, optical flow in conjunction with a volumetric reconstruction obtained by the 3D Probabilistic Occupancy Grid method. This technique was recently introduced for use in environments monitored by multiple cameras and still there are few studies that employ it in motion tracking.

A volumetric reconstruction performed with the Probabilistic Occupancy Grid method presented considerable advantages over the traditional methods of Shape-from-Silhuette, as showed in the results section. This fact is due to the advantage of probabilistic reconstruction method to consider information from all images together through fusion obtained by Bayesian inference: the Probabilistic Grid does not perform premature evaluations of the analysis of each camera separately, as traditional methods.

A differential of the reconstruction method employed in this paper is the ability to overcome problems of noise and

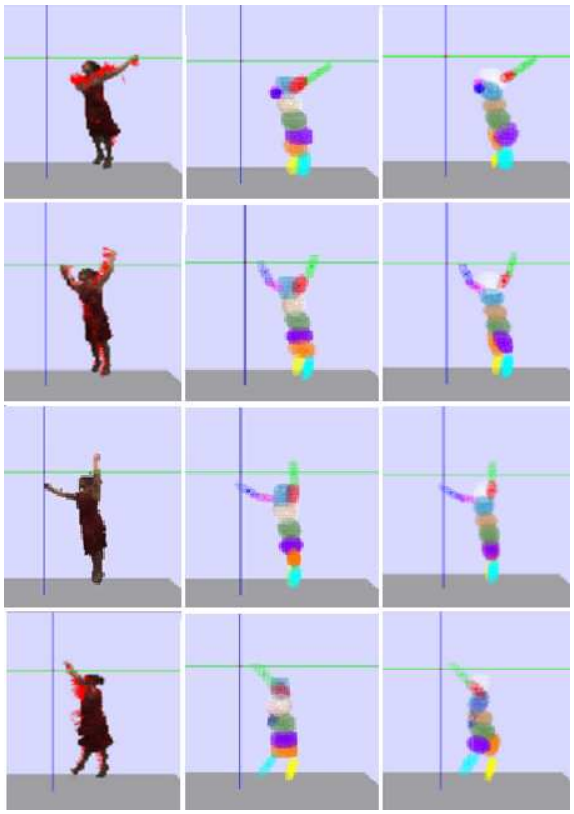

Figure 8. Tracking results. Column 1: reconstruction and optical flow; Column 2: Tracking without optical flow; Column 3: Tracking with optical flow.

incomplete reconstructions (common in the traditional methods). In the showed results, the traditional reconstruction methods failed in the analysis of body parts of objects of interest whose color resembled the color of the background. In addition, traditional methods were more susceptible to variations in light intensity (shadows).

The optical flow has proved to be an important source of information about the movement, allowing the correct execution of the motion tracking, even without consideration of joint positions or any definitions of relative positions between the blobs and, therefore, the method can be applied to objects of different shapes.

Finally, we can conclude that this study showed promising results, establishing a basis for the investigation of more robust methods. As future work, we intend evaluate more accurate tracking algorithms.

ACKNOWLEDGMENT

We thank the National Council for the Improvement of Higher Education (CAPES) and Brazilian Council for Scientific and Technological Development (CNPq).

REFERENCES

[1] H. Zhoua and H. Hu, "Human motion tracking for rehabilitation - a survey," *Biomedical Signal Processing and Control*, 2008.

[2] R. W. Poppe, "Discriminative vision-based recovery and recognition of human motion," PhD Thesis, University of Twente, 2009.

[3] P. Huang, A. Hilton, and J. Starck, "Human motion synthesis from 3d video," *CVPR*, 2009.

[4] C. Canton-Ferrer, J. Casas, and M. Pards, "Voxel-based annealed particle filtering for markerless 3d articulated motion capture," *3DTV Conference*, 2009.

[5] F. Caillette, "Real-time markerless 3d human body tracking," PhD Thesis, University of Manchester, 2006.

[6] J. S. Franco and E. Boyer, "Fusion of multi-view silhouette cues using a space occupancy grid," *ICCV 05*, 2005.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of Imaging understanding workshop*, 1981.

[9] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer*, 1989.

[10] C. Wren, A. Azarbayeajani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[11] G. M. Simas, G. P. Fickel, L. Novelo, R. A. Bem, and S. S. C. Botelho, "Utilizando visão computacional para reconstrução probabilística 3d e rastreamento de movimento," *III MCSul*, 2009.

[12] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[13] R. L. Barbosa, R. B. A. Gallis, J. F. C. Silva, and M. M. Júnior, "A computação do fluxo óptico em imagens obtidas por um sistema móvel de mapeamento terrestre," *Revista Brasileira de Cartografia*, 2005.

[14] J. L. Barron and N. A. Thacker, "Tutorial: Computing 2d and 3d optical flow," in *Tina Memo Internal*, University of Manchester, 2005.

[15] A. Bjorck, "Numerical methods for least squares problems," *SIAM*, 1996.

[16] M. Galo and C. L. Tozzi, "A representação de matizes de rotação e o uso de quaternions em ciências geodésicas," *Série em Ciências Geodésicas*, 2001.

[17] O. Chum, T. Pajdla, and P. Sturm, "The geometric error for homographies," *Computer Vision and Image Understanding*, 2005.

[18] Perception-Group, "Perception's website," http://perception.inrialpes.fr, 2008.