Kinematic-based Markerless Human Tracking in 3D Probabilistic Occupancy Grids

Rodrigo de Bem, Maurício Goulart, Gisele Simas and Silvia Botelho

***Abstract*—Markerless human motion tracking can be employed in many applications such as automatic surveillance, motion capture, human-machine interface and activity recognition. This problem has been extensively studied in the computer vision research community in the last years. In this context, the present paper presents an approach for 3D markerless human motion tracking based on a skeletal kinematic model of the human body. This method is applied over a 3D probabilistic occupancy grid of the environment, which is constructed by means of a Bayesian fusion of images from multiple synchronized sensoring cameras. Although the use of kinematic models in 3D human tracking is widely employed, its use over 3D probabilistic occupancy grids still was not vastly investigated in the literature. The experiments were performed using a public dataset with video sequences of people in motion. The results show that the method is capable of dealing adequately with the 3D markerless human motion tracking problem.**

***Index Terms*—skeletal kinematic model, 3D probabilistic occupancy grid, 3D markerless human tracking.**

I. Introduction

Visual tracking, which is the recursive detection and location of objects (or more generally, visual patterns) in videos [1], is a classical computer vision problem. In this context, the visual tracking of people has been studied extensively in the literature [2], [3]. Among many applications, one could mention automatic surveillance, motion capture, human-machine interface and activity recognition, as examples of relevant problems where the visual tracking of people is employed.

Tracking articulated targets, such as the human body, using 2D images is a difficult problem to be treated mainly due to: i) the complex nature of 3D movements; ii) the loss of information in images because of 2D space restriction; iii) the color changes caused by luminosity variations; iv) the existence of others objects moving into the scene. Thus, to minimize some of these issues, multiple synchronized cameras can be used to sensor the environment where people are moving. From the set of images captured by the cameras a 3D reconstruction of the environment can be performed.

The reconstruction technique used in the present approach is called 3D probabilistic occupancy grid [4]. This technique was proposed as a way to overcome some problems, for instance, the existence of phantom volumes and holes in the reconstructions. These problems occur when other popular methods are employed, such as the shape-from-silhouette. In the 3D probabilistic occupancy grid approach, the images obtained by the multiple synchronized cameras are fused by means of a Bayesian inference [5]. Doing so, the decision about the occupancy or not of a position (voxel) into the 3D space is taken accounting all the information coming from each camera, allowing a better inference compared with shape-from-silhouette, which takes the decision accounting each 2D image separately.

Although one can find some model-free proposals in the literature [6], many 3D motion tracking methods employ predefined representation models of the targets [7], [8]. Usually, the object appearance model is associated with the object kinematic model that describes the possible movements and valid poses [9]. The main contribution of the present work, which is still not vastly investigated in the literature, is the use of a human body skeletal kinematic model to perform the markerless visual tracking of people in a 3D probabilistic occupancy grid. Experiments were performed with videos sequences from a public dataset [10]. The results show that the method is capable of dealing adequately with the 3D markerless human motion tracking problem.

II. Related Work

Markerless human tracking and motion analysis have been studied for some years and remarkable achievements were already accomplished [2], [3]. Several approaches can be found in the literature, which deal with many different application's scenarios, data acquisition approaches and tracking methodologies. A vast review of works can be find in [12], [13], [14], [15].

In this context, it is hard to establish a sufficiently general taxonomy that could be used to classify and analyze the works in the field. Thus, some relevant attributes must be chosen in a way that some extent of comparison among different approaches is allowed. It can be noticed, for instance, that many methods focus on the recovery of human poses from monocular images or videos. Agarwal and Triggs [16] propose a learning-based method along with a histogram-of-shape-context silhouette shape descriptor which allows the recovery of 3D human body pose from monocular images. Urtasun et al. [17] also present a method for 3D pose recovery from monocular videos, but this method employs a strong motion model in the tracking process and do not use a learning approach.

The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Vanessa Testoni.

Rodrigo de Bem (rodrigobem@furg.br), Gisele Simas (giselesimas@furg.br) and Silvia Botelho (silviacb@furg.br) are professors affiliated to the Computational Sciences Center (http://www.c3.furg.br) of the University of Rio Grande (http://www.furg.br), Rio Grande, Brazil. Maurício Goulart (mauricio.goulart@furg.br) is a master student of the Computational Modeling course at the same institute.

This paper is part of the Special Issue on Vision-based Human Activity Recognition.

Digital Object Identifier 10.14209/jcis.2015.12

Bourdev and Malik [18], as well as Yang and Ramanan [19], focus on the identification of body parts to pursue the pose estimation. The former propose the concept of poselets, which maps the appearance of body parts to their 3D pose; while the latter learn the different body parts based on the HOG descriptor [20]. More recently, deep neural networks have also been applied to the monocular pose estimation problem, as presented by Toshev and Szegedy [21] and Tompson et al. [22].

Another extremely relevant methods are the ones based on depth images. The advent of low cost devices, such as the Microsoft Kinect, called attention of the research community for such kind of input data. Certainly a remarkable work in this subject is the approach presented by Shotton et al. [23] in which deep decision random forests [24] are used to recover the 3D human pose in real-time from a single depth image. In more recent works, Vemulapalli et al. [25], also recover 3D human poses from depth images, in the context of action recognition, by mapping the skeletons models to a Lie group; and Ionescu et al. [26] estimate the 3D human pose classifying body parts using SIFT [27] and random forests.

Multiple views methods are another kind of approach which still achieves the highest accuracy among the markerless methodologies [28]. In this context, Sigal et al. [29], [30] present a probabilistic graphical model to represent and track a human body in a multiple camera environment. Simas et al. [6] propose a method based on nonparametric belief propagation which can be applied to track people and other previous unknown moving objects. In the works of Starck and Hilton [31] and De Aguiar et al. [32], the employed human body representation model is a mesh-based surface. More recently, Elhayek et al. [28] and Belagiannis et al. [33] proposed approaches in multiple views scenarios facing challenging situations, such as self-occlusions, moving cameras, outdoor scenes and cluttered background. In this context, despite the several methodologies found in the literature concerning the markerless human motion tracking, the use of a kinematic model over a 3D probabilistic occupancy still was not well explored.

III. Proposed Methodology

A typical visual tracking system is formed by four main components: the observation model, the representation model, the movement model and the tracking algorithm. In the proposed approach the target (a moving person) is observed by means of a 3D probabilistic occupancy grid. The representation model is composed of two parts: a set of Gaussian blobs, modeling the volume occupied by each rigid body part (appearance model); and a kinematic hierarchical model (human body skeletal model), representing the spatial relation between the rigid body parts. In this work no movement model has been used. Finally, the tracking algorithm is based on the Expectation-Maximization (EM) algorithm and on the Cyclic-Coordinate Descent (CCD) inverse kinematic method. The proposed approach is summarized by the diagram shown in the Fig. 1. Each one of the components will be explained in further details in the following sections.

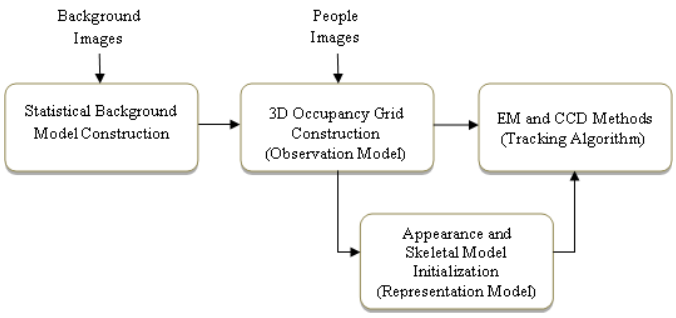


Fig. 1. Diagram showing the main building blocks of the proposed methodology.

A. Observation Model

The observation model defines which kind of sensor information about the targets is extracted from the environment. In the present method, the environment is sensored by multiple synchronized and calibrated cameras. The captured images are used to build a probabilistic volumetric reconstruction of the scene. This reconstruction is composed of voxels, which present a probability to be occupied by the interested objects. Many volumetric reconstruction methods use a simple binary background segmentation of each image and analyze those images individually. Despite being simple and broadly used, these methods can lead to some problems in the determination of the object's volume and position, such as phantom volumes and holes into objects.

To deal with these difficulties Franco and Boyer [4] propose to obtain a fusion of all image information using a 3D probabilistic occupancy grid. This technique tackles some difficulties presented by a number of uncertainties associated to the image capturing stage, like sensor noise, calibration errors and lightning changes.

In this method, every pixel of a camera is treated as a statistical sensor susceptible to uncertainties. The problem is then treated as a Bayesian estimation. The 3D space is discretized into volume elements, called voxels. The Bayesian estimation is used to calculate the probability of each voxel to be occupied by the object of interest.

In the determination of a voxel's occupancy status, the value of the projected pixels are taken into consideration along with a statistical background model for those pixels [35]. The background model is obtained using a video sequence of the background scene without moving objects. The projection between pixels and corresponding voxels is done using the cameras' calibration matrices. A brief review of the method is presented below, while a detailed explanation can be found in [4].

1) Occupancy Grid Theory Review: The voxel occupancy inference is performed using the Bayes' rule. The probability of each voxel of the grid to be occupied is given by

$$p(\mathcal{G}_X|\mathcal{I},\mathcal{B},\tau) = \frac{\prod_{i,p} p(\mathcal{I}_p^i|\mathcal{G}_X,\mathcal{B}_p^i,\tau)}{\sum_{\mathcal{G}_X}\prod_{i,p} p(\mathcal{I}_p^i|\mathcal{G}_X,\mathcal{B}_p^i,\tau)}, \quad (1)$$

where $\mathcal{I}$ is the set of n current images, and $\mathcal{I}_p^i$ is the image data at pixel p in the image of camera i, $i = 1...n$. It is

assumed that the image data corresponding to the set of m observed background images can be summarized into a single statistical model $\mathcal{B}_p^i$ for each pixel p in the image of each camera i, $i = 1...n$. τ symbolizes the prior knowledge about the scene, about the sensor characteristics and the general knowledge about the system. Finally, $\mathcal{G}$ is the space occupancy grid. For each space point X in the grid discretization the voxel occupancy probability is inferred according to the Bayes' rule shown in the Equation (1).

a) Generic sensor model: is the term $p(\mathcal{I}_p^i|\mathcal{G}_X, \mathcal{B}_p^i, \tau)$ used to define the Bayes' rule Equation (1). This term directly relates the pixel's color observation to voxel occupancy, using the **image formation** $p(\mathcal{I}_p^i|\mathcal{F}_p^i, \mathcal{B}_p^i, \tau)$ and **silhouette formation** $p(\mathcal{F}_p^i|\mathcal{G}_X, \tau)$ terms, that will be presented after, and marginalizing over silhouette detection $\mathcal{F}_p^i$ as expressed below:

$$p(\mathcal{I}_p^i|\mathcal{G}_X, \mathcal{B}_p^i, \tau) = \sum_{\mathcal{F}_p^i} p(\mathcal{F}_p^i|\mathcal{G}_X, \tau) p(\mathcal{I}_p^i|\mathcal{F}_p^i, \mathcal{B}_p^i, \tau), \quad (2)$$

where $\mathcal{F}_p^i$ is a binary silhouette detection variable for the pixel p in the image i, $i = 1...n$. $\mathcal{F}_p^i = 1$ if the pixel sensor p in image i reports the presence of an object of interest anywhere along its viewing line, and $\mathcal{F}_p^i = 0$ otherwise.

b) Silhouette formation term: models the silhouette detection response of a single pixel sensor (i, p) to the occupancy state of the analysed voxel $\mathcal{G}_X$. The silhouette formation term is defined by two expressions, considering the case when the voxel is occupied ($\mathcal{G}_X = 1$) and when it is not ($\mathcal{G}_X = 0$):

$$\begin{aligned} p(\mathcal{F}_p^i|[\mathcal{G}_X = 1], \tau) &= p(\mathcal{S} = 0|\tau)\mathcal{U}(\mathcal{F}_p^i) \\ &+ p(\mathcal{S} = 1|\tau)\mathcal{P}_d(\mathcal{F}_p^i), \end{aligned} \quad (3)$$

$$\begin{aligned} p(\mathcal{F}_p^i|[\mathcal{G}_X = 0], \tau) &= p(\mathcal{S} = 0|\tau)\mathcal{U}(\mathcal{F}_p^i) \\ &+ p(\mathcal{S} = 1|\tau)[p(\mathcal{R} = 1|\tau)\mathcal{P}_d(\mathcal{F}_p^i) \\ &+ p(\mathcal{R} = 0|\tau)\mathcal{P}_f(\mathcal{F}_p^i)], \end{aligned} \quad (4)$$

where $\mathcal{S}$ and $\mathcal{R}$ are hidden variables. $\mathcal{S}$, the *sampling variable*, is equal to 1 if the voxel X is on the viewing line of pixel (i, p) and equal to 0 if it is not. The *external detection cause* $\mathcal{R}$ equals to 1, accounts for the possibility that some other object lies on the same viewing line as the voxel, and it is equals to 0 if no other object obstructs the viewing line. $\mathcal{U}(\mathcal{F}_p^i)$ is the uniform distribution for the silhouette detection, when the voxel and pixel are not aligned ($\mathcal{S} = 0$), while $\mathcal{P}_d(\mathcal{F}_p^i)$ and $\mathcal{P}_f(\mathcal{F}_p^i)$ are respectively, the **detection** and the **false alarm** probability distributions (when $S = 0$) defined as:

$$\begin{aligned} \mathcal{P}_d([\mathcal{F}_p^i = 1]) = P_D, \quad \mathcal{P}_d([\mathcal{F}_p^i = 0]) = 1 - P_D, \\ P_D \in [0; 1], \end{aligned} \quad (5)$$

$$\begin{aligned} \mathcal{P}_f([\mathcal{F}_p^i = 1]) = P_{FA}, \quad \mathcal{P}_f([\mathcal{F}_p^i = 0]) = 1 - P_{FA}, \\ P_{FA} \in [0; 1]. \end{aligned} \quad (6)$$

Still in the Equations 3 and 4, it is needed to define the **parametric forms** of $p(\mathcal{R}|\tau)$ (*external detection term*) and $p(\mathcal{S}|\tau)$ (*sampling term*). Both terms are considered uniforms. Concerning the *external detection term* it means that the detection is equally likely to be triggered by the voxel occupancy or by other causes anywhere along the viewing line of p. Considering the *sampling term*, the uniform sampling means that all the voxels that fall within a $k \times k$ window around the pixel p have equal weight.

c) Image formation term: seeks to explain the color information of a pixel (i, p), given the knowledge of the background color model at this pixel and whether or not a silhouette detection occurred at this pixel. This term is defined as by two expressions, the first one for the case of silhouette detection at pixel (i, p), and second for the case when no silhouette detection occurred:

$$p(\mathcal{I}_p^i|[\mathcal{F}_p^i = 1], \mathcal{B}_p^i, \tau) = \mathcal{U}(\mathcal{I}_p^i), \quad (7)$$

$$p(\mathcal{I}_p^i|[\mathcal{F}_p^i = 0], [\mathcal{B}_p^i = (\mu_p^i, \sigma_p^i)], \tau) = \mathcal{N}(\mathcal{I}_p^i|\mu_p^i, \sigma_p^i), \quad (8)$$

where $\mathcal{U}_p^i$ is the uniform distribution over the observed colors when the silhouette is detected, since there is no knowledge about the color of the objects of interest. And $\mathcal{N}(\mathcal{I}_p^i|\mu_p^i, \sigma_p^i)$ is a normal distribution in (Y,U,V) space for each pixel, that defines the color background model. The parameters μ_p^i and σ_p^i are estimated from a set of background sample images. Considering the brief presentation until here, the voxel occupancy inference algorithm can be defined as shown in the Algorithm 1.

Algorithm 1 Voxel occupancy inference

for each voxel X in the grid **do**
  for each image i from each one of the cameras **do**
    - calculate the X voxel projection in the image i
    for each pixel p in the $k \times k$ window around the projection of the voxel X **do**
      $p(\mathcal{G}_X|\mathcal{I}, \mathcal{B}, \tau)^* = \frac{\prod_{i,p} p(\mathcal{I}_p^i|\mathcal{G}_X, \mathcal{B}_p^i, \tau)}{\sum_{\mathcal{G}_X} \prod_{i,p} p(\mathcal{I}_p^i|\mathcal{G}_X, \mathcal{B}_p^i, \tau)}$
    end for
  end for
end for
* Sum of log probabilities

B. Representation Model

The representation model defines how the interested objects are "seen" by the motion tracking method. It is composed of two main parts: the appearance model and the skeletal kinematic model.

The appearance model is composed of a set of associated Gaussian blobs [36]. As the human body is an articulated object composed of rigid parts, each blob models a rigid body part of the tracked person. A blob is a Gaussian distribution in tree dimensions often represented by an ellipsoidal shape. Its mean position is given by

$$\mu_X = \begin{pmatrix} \mu_x & \mu_y & \mu_z \end{pmatrix}^T, \quad (9)$$

while the surface is defined by a standard deviation around the mean, defined by

$$\sigma_X = \begin{pmatrix} \hat{\alpha_x^2} & \alpha_{xy} & \alpha_{xz} \\ \alpha_{xy} & \hat{\alpha_y^2} & \alpha_{yz} \\ \alpha_{xz} & \alpha_{yz} & \hat{\alpha_z^2} \end{pmatrix}. \quad (10)$$

These geometric shapes enclose the occupied voxels belonging to objects' rigid parts. The relation between the occupied voxels of the grid and the blobs is given by the Mahalanobis distance: closer voxels have greater probabilities of belonging to a certain blob. The basic components of the representation model are illustrated in Fig. 2.

(a) Ellipsoid (b) Ellipsoid attached to a bone

Fig. 2. Representation model: (a) the appearance model is composed of ellipsoidal geometric shapes (Gaussian blobs), which represent the human body rigid parts; (b) the skeletal kinematic model is composed of a set of joints and limbs with the ellipsoids attached to it.

This statistical approach is good to model body parts and fits well with the probabilistic reconstruction method employed, however it carries no information about the relationship between connected body parts, e.g. the position of the hand depends on the position of the arm, which depends on the chest. To tackle these difficulties the skeletal kinematic model is introduced.

The skeletal kinematic model, adapted by the authors from the model presented by Caillette [36], is shown in Fig. 3. It describes the kinematic relationship between rigid body parts. In the human body the moving parts are the limbs which are hold together by joints. A hierarchical skeletal kinematic model is used, in which the bones are attached to joints in a tree structure. These joints are linked one to another from the root of the skeleton (in the pelvis) to the leaves (hands and feet). The kinematic skeleton presents two important functions: it is used to constrain the movement of the blobs to valid poses; and it can be used to interpret the resulting poses of the body.

Into the present approach only the main parts of the human body are represented in the skeletal model, such as torso, arms and legs. Smaller parts, e.g. hands and fingers, are neglected, since the resolution of the 3D reconstruction is not high enough to allow the effective observation of these tiny parts. Another important issue related to the model is the definition of the joints. The interdependence between the adjacent body parts movements creates complex constraints in the rotation of the joints.

Basically, a kinematic model can be employed in direct or inverse kinematic problems. In the first scenario the model state is determined according to a defined set of joints parameters. In the latter situation, the set of joints's angles must be found as a function of a determined model pose. The inverse kinematic problem is usually solved as an optimization problem.

The global position of the model, concerning the root of the kinematic hierarchical tree, is defined by 6 parameters (i.e. 3 for translation and 3 for rotation). The position of each node (joint) is defined according to the position of the parent node. Considering this, the position of a leaf (e.g. foot) is obtained through the application of recursive transformations from the root to the leafs. The Euler angles were used to encode the joints' configurations.

The joints of the model are represented by $\{J1, J2, ..., J21\}$ and the root is located in the pelvis region. All the joints are modeled with 1 degree of freedom (rotation around a fixed axis) because simplicity and performance reasons. More complex joints are indeed ensembles of single joints. To exemplify the functioning of the kinematic model, a transformation between a joint Jt_{i-1} and its child joint Jt_i is defined by a rotation θ_i around the axis w_i in the Jt_{i-1} coordinate system. The rotation is followed by a translation l_i which is correspondent to the length of the bone which link both joints. The global and final position of the joint Jt_i is defined by P_i after the applied rotation and translation. The maximum and minimum possible values for each joint angle θ_i are in the interval $[\theta_i^-, \theta_i^+]$.

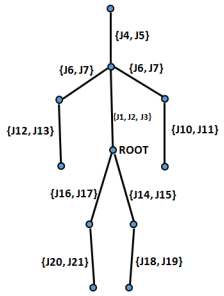


Fig. 3. Skeletal kinematic model, with the root and the other joints $\{J1, J2, ..., J21\}$.

C. Tracking Algorithm

The tracking algorithm matches the representation model to the 3D occupancy grid at each time instant, finding the most likely pose of the tracked person in a given frame. The proposed approach is composed of four steps: i) initialization, ii) updating blobs' parameters using Expectation-Maximization (EM), iii) kinematic model pose estimation by the means of Cyclic-Coordinate Descent (CCD) method, and iv) realignment of the blobs with the kinematic model (bones). Each of the steps is detailed below and shown in diagram of Fig. 4.

i) Initialization: In the first frame of the video sequence the pose of the representation model needs to be initialized. In this work a manual adjustment of the model has been employed. The user must adjust the pose of the skeletal kinematic model, along with the associated Gaussian blobs, to the reconstructed 3D volume in the first frame of the sequence. The defined pose is saved and can be reused in other videos without the necessity of repeating the process. This mechanism, although manual, allows the tracking of people in any initial pose.

ii) Updating Blobs with EM: From a previous frame of a video sequence, the representation model needs to be matched to the current human body pose. To perform this iterative matching through the frames, the EM algorithm is used [37]. In this process the Gaussian blobs' parameters are updated from

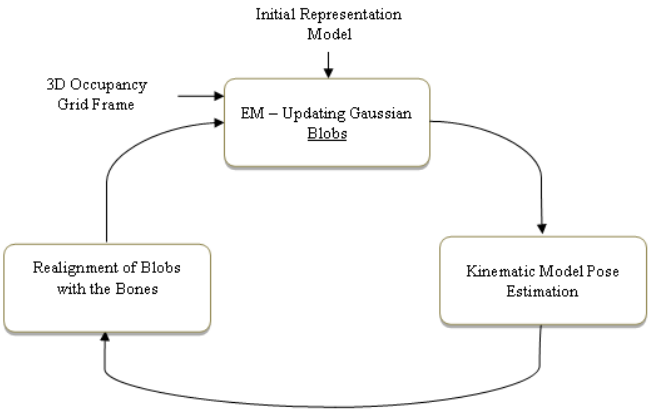


Fig. 4. The tracking process is summarized by the diagram showing all algorithm's steps.

the previous 3D reconstruction frame. In the Expectation step, each voxel of the current frame is associated with the blob with the highest probability of representing that body part, given by the minimal Mahalanobis distance between the voxel and the blob. Then, in Maximization step, the mean vector and covariance matrix of the blobs are estimated according to the associated voxels.

iii) Kinematic Model Pose Estimation: Using the new position of the blobs, given by the EM algorithm in the previous step, the skeletal kinematic model is updated in two phases. First, goal unconstrained positions for the joints are obtained from the associated blobs parameters. Second, which is an inverse kinematic problem, the joints' parameters (joints' angles) must be determined, given the goal positions defined by the blobs in the previous step. A simple approach for the problem is the Cyclic-Coordinate Descent (CCD) algorithm, an iterative local optimization method [36]. Each joint of the skeleton is optimized in an independent way, from leaves to the root. The optimization consists in minimizing the error between the joint positions and their goal positions. Joint angular limitation is applied by clamping the resulting angle.

As shown in Fig. 2, each blob is attached to a bone in the skeletal model. This attachment can be seen as a virtual spring, which pulls the bone towards the blob's position and orientation; the former is defined by the mean, while the latter is given by the biggest axis of the blob. The Algorithm 2 calculates the goal positions for the kinematic model joints. It typically needs only a few iterations to generate a satisfying solution. Further details are provided in [36].

After the goal joints positions calculation, the inverse kinematic problem must be solved, which means that the global configuration of the skeletal model needs to be determined according to the goal positions and to the models' constraints. In this context, the global position and orientation of the root's kinematic tree is defined by the ordered pair (P_0, R_0); r_i defines the local rotation of θ_i over the local axis w_i, and t_i the translation of length l_i along the first axis of the local coordinate system. The global position P_i of the joint Jt_i is calculated recursively in the kinematic chain $\{Jt_1, ..., Jt_{N_J}\}$ as follows:

Algorithm 2 Calculation the goal joints positions

while the sum of the squared distances between the goals from the last iteration and the current ones is below a predefined threshold **do**

1. compute the goal position for the tip of the current joint using the base of the joint as a fixed rotation point;
2. translate the goal positions of the base and tip of each joint so as to minimize the projection error of the mean of the blobs onto the bone;
3. optimize the goal position of the base of the joint using this time the goal position of the extremity of the joint as a fixed rotation point;
4. the goal position coming from both the current joint and its parent are merged into a single goal position;

end while

$$P_i = P_{i-1} + R_i.t_i \tag{11}$$

where,

$$R_i = R_{i-1}.r_i. \tag{12}$$

Considering these two equations, the direct kinematic formulation is defined by

$$P_i = P_0 + R_0.r_1(t_1 + r_2(t_2 + (... + r_{i-1}(t_{i-1} + r_i t_i)))). \tag{13}$$

This is the formulation for the definition of the entire skeletal kinematic model. The intermediate and recurrent calculations are stored to reuse, avoiding unnecessary computation. Another important optimization adopted was the implementation of the local rotations r_i using quaternions [38], which allow the speed up of rotations calculations, specially when t_i is null.

Considering this formulation of the direct kinematic, the inverse kinematic problem is solved using the Cyclic-Coordinate Descent (CCD) approach. The CCD is a local iterative optimization method. In this method each joint of the kinematic skeletal model is optimized independently, from the leafs up to the root of the hierarchical tree. The CCD method tries to minimize the distance between each joint, and its children joints, and their goal positions calculated in the previous step.

In this context, lets consider a joint Jt_i and its children joints $\{Jt_{i,1}, ..., Jt_{i,n}\}$, with global position given by $\{P_i, P_{i,1}, ..., P_{i,n}\}$ and goal positions $\{G_i, G_{i,1}, ..., G_{i,n}\}$. Considering the simplest optimization problem, which would be the optimization of the joint Jt_i alone or just with one child, the unique degree of freedom of such joint is the rotation θ_i over the axis w_i. Thus, the aim is to calculate the variation $\Delta\theta_i$ which minimize the distance between the joint position $P_{i,1}$ and the goal position $G_{i,1}$.

The position of the joint Jt_i base is P_{i-1}, thus lets consider $\vec{P} = \overrightarrow{P_{i-1}P_{i,1}}$ and $\vec{G} = \overrightarrow{P_{i-1}G_{i,1}}$. In this context, the angular variation $\Delta\theta_i$ which minimizes the distance between $P_{i,1}$ and $G_{i,1}$ also maximizes the scalar product between $\vec{P}$ and $\vec{G}$, having the following expression [39]:

$$\Delta\theta_i = \arctan \frac{w_i.(\vec{P} \wedge \vec{G})}{\vec{G}.\vec{P} - (\vec{G}.w_i).(\vec{P}.w_i)}. \quad (14)$$

The Equation 14 find a solution in the interval $\Delta\theta_i \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. To allow a smooth movement between all the joint of the kinematic tree, an attenuation coefficient η is introduced. This attenuation also limits undesirable oscillations while all the joints are being optimized simultaneously towards contradictory goal positions. In each iteration the angle θ_i is updated according to the following equation:

$$\theta_i = \theta_i + \eta.\Delta\theta_i, \quad (15)$$

where the coefficient η controls the convergence rate. The employed value was determined empirically as $\eta = 0.3$.

From this point, lets consider the existence of more than one child of the joint Jt_i with goal positions to be achieved. The analytical calculation of $\Delta\theta_i$ becomes a complex task. Thus, a heuristic is employed to combine all the individual optimizations of the the joints. Lets denote $\{\Delta\theta_{i,1}, ..., \Delta\theta_{i,k}\}$ as the variations calculated in the Equation 14. The angular variation for the current joint Jt_i is the weighted summation of the individual angles, as follows:

$$\Delta\theta_i = \frac{1}{\sum_{j=1}^{k} \lambda_{i,j}}. \sum_{j=1}^{k} \lambda_{i,j}\Delta\theta_i, \quad (16)$$

where $\{\lambda_{i,1}, ..., \lambda_{i,k}\}$ are the weights which gave more importance to the goals closer to the root of the kinematic tree. If the weights were uniform, the goal positions closer to the root of the kinematic model would just be partially optimized, while the the goal positions near to the kinematic chain's extremities would be favored by their more frequent updates. Considering this, $\lambda_{i,j}$ is equal to the inverse of the number of joints separating Jt_i to the joint associated with the goal $G_{i,j}$.

iv) Realignment of the Blobs with the Bones: The last step is to realign the blobs with the bones of the skeletal model. Since the blobs' shapes are supposed to remain the same through the tracking, the regeneration of the blobs consists in a series of rotations and translations along the skeleton kinematic tree. According to [36], lets assume that the blob B is attached at an offset $\hat{\alpha}$ along a bone of the skeletal model, lets denote by P the global position of this bone obtained after application of the kinematic constraints, and lets denote by R the associated rotation matrix. The corrected mean μ'_X is computed as a simple conversion from local to global coordinates given by

$$\mu'_X = P + R. \begin{pmatrix} \hat{\alpha} \\ 0 \\ 0 \end{pmatrix}, \quad (17)$$

while the corrected covariance matrix σ'_X of blob B is given by

$$\sigma'_X = R. \begin{pmatrix} \hat{\alpha_x^2} & 0 & 0 \\ 0 & \hat{\alpha_y^2} & 0 \\ 0 & 0 & \hat{\alpha_z^2} \end{pmatrix}. \quad (18)$$

IV. Implementation

The proposed methodology was implemented using the C++ programming language and OpenCV and OpenGL libraries. The developed software executes all the stages of the method, from the 3D reconstruction until the markerless human tracking. The software tool includes a visualization functionality, as can be observed in Fig. **??**, and it also allows the entire configuration of method's parameters by the means of the graphical interface shown in Fig. **??**. Besides this, the software is also fundamental in the initialization of the representation model, which is performed as a interactive process. The user should select each joint and visually adjust the approximated angles between articulations. The blobs must also be attached to the bones of the skeleton. The initialization defined by the user can be saved by the software tool and reloaded to be used as the initial configuration of other videos sequences.

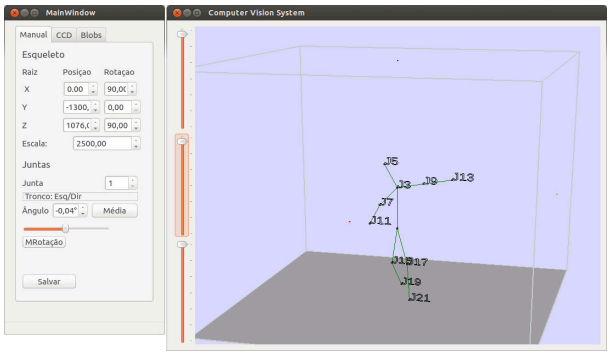


Fig. 5. Data visualization interface.

Fig. 6. Parameters configuration interface.

V. Experiments and Results

Several experiments were performed to assess all the different parts of the proposed methodology. The video sequences of moving people were obtained in the public online dataset 4D Repository [10]. The repository contains a set of live and dynamic events, such as human activities, captured using the multi-camera platform GRImage [11]. The dataset provides, for each sequence: i) the calibration information for the multi-camera set up, ii) images acquired from multiple cameras,

iii) silhouettes extracted from these images by eliminating the background, iv) reconstructed mesh geometry at different time frames. Fig. 7 shows a 3D probabilistic occupancy grid, while a isosurface showing only the voxels with a defined probability of occupancy is shown in Fig. 8.

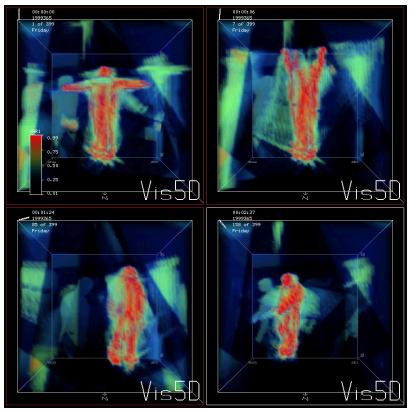


Fig. 7. Four frames of the 3D occupancy grid of the observed environment where a person moves. The red regions represents the higher probabilities of the voxels to be occupied. Visualization performed with the Vis5D software [34].

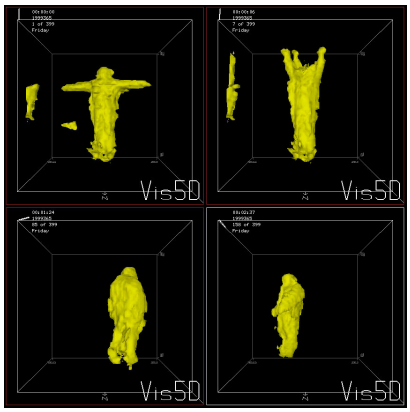
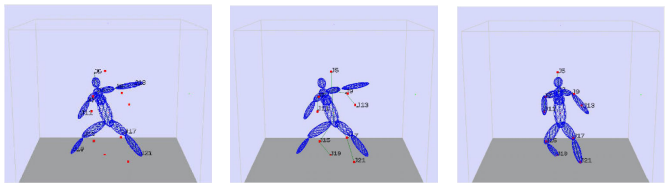


Fig. 8. The isosurface of probability 0.70 in the occupancy grid. Visualization performed with the Vis5D software [34].

Concerning the tracking process, experiments were realized to evaluate the effectiveness of the pipeline shown in Fig. 4. The iterative tracking process is initiated by the EM algorithm which adjust the Gaussian blobs to the new frame voxels. From this step the goal positions for the kinematic model joints are obtained, as shown in Fig. **??**(a). These positions are the input of the kinematic model pose estimation, where the inverse kinematic problem is solved using the CCD method, shown in Fig. **??**(b). Finally the Gaussian blobs are realigned to the skeletal model, which is shown in Fig. **??**(c).



(a) New goal positions denoted by the (red) dots.

(b) Kinematic model pose estimation with the CCD method.

(c) Realignment of the blobs to the kinematic model.

Fig. 9. Kinematic-based tracking process.

The CCD algorithm functioning is illustrated in Fig. 10. From a initial state of the kinematic model and a set of goal positions, the method iteratively minimizes the distance between the joint and the goal positions. It can be noticed that in the final configuration some of the joints are not exactly located over the goal position. It happens because the optimization is performed considering the kinematic constraints over the global model.

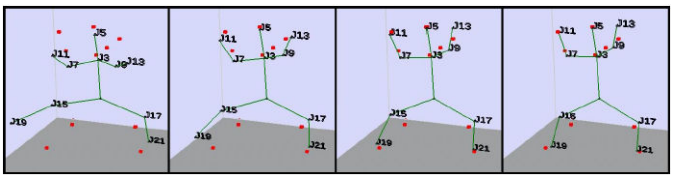


Fig. 10. Four iterations of the Cyclic-Coordinate Descent (CCD) algorithm, from the goal positions (red dots) and the initial model state (left), to the optimized final configuration (right).

Besides the evaluation and the experiments performed with some of the individual parts of the methodology, an overall assessment was realized. To this purpose a video sequence was used, which consists of a dancer performing dance movements captured by eight synchronized and calibrated RGB cameras positioned around the environment, shown in Fig. 11. In Fig. 12 a volumetric reconstruction of the human body is shown, along with the correspondent representation model. The tracking process was executed over the sequence and Fig. 13 shows three frames of the volumetric reconstruction from the eight cameras' views. Fig. 14 shows the tracked representation model pose for the same frames presented in Fig. 13. As can be observed, the kinematic model imposes restrictions to the body pose. Doing so, the tracking process became more robust, once the model cannot assume invalid poses. The appearance model just along with the tracking algorithm is not capable of achieve such performance, because it does not present the semantic information concerning the human body configuration embedded in the skeletal kinematic model.

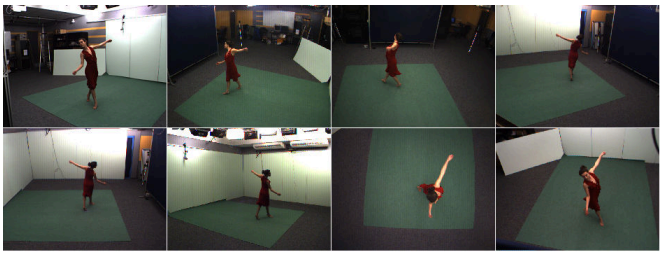


Fig. 11. Sample input images from eight synchronized cameras at a given instant.

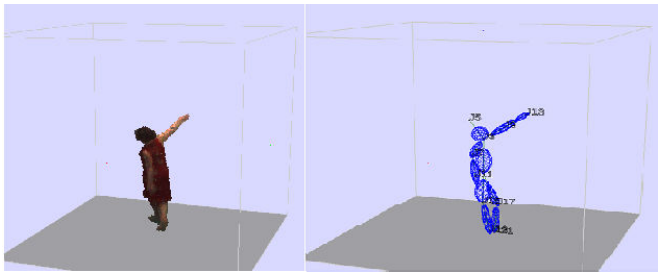


Fig. 12. Sample results: the 3D probabilistic reconstruction of the tracked human body (left), the resulting pose of the human body representation model (skeletal kinematic model and associated Gaussian blobs) (right).

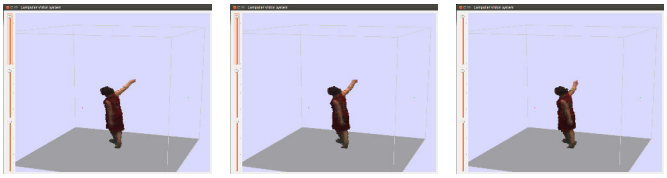


Fig. 13. 3D probabilistic reconstruction.

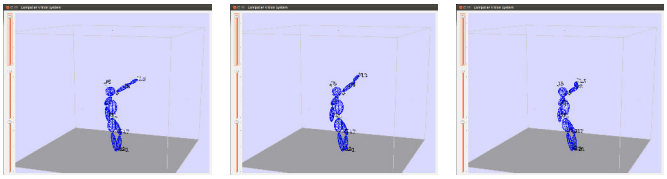


Fig. 14. Dancer representation model pose, composed of the skeletal kinematic model and the associated Gaussian blobs.

VI. Conclusions and Future Work

This paper presented an approach for the visual 3D markerless human body tracking in an environment monitored by multiple cameras. The method employ the volumetric reconstruction of the environment obtained by means of a 3D probabilistic occupancy grid. The markerless 3D tracking of human body is achieved by the use of a skeletal kinematic model associated to Gaussian blobs. Such representation model is iteratively updated by the Expectation-Maximization method, along with the Cyclic-Coordinate Descent algorithm, which solves the inverse kinematic problem. The main contribution of the proposal is to employ a kinematic model over a 3D probabilistic occupancy grid, what is still not extensively investigated in the literature. The tracking algorithm performed successfully as shown by the obtained results. The kinematic model limit the blobs positions to valid human body poses, besides adding semantic meaning to the representation model. As future works, a quantitative evaluation of the results is desirable. More experiments also must be performed to precisely determine strengths and limitations of the methodology.

References

[1] Perez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. In: Proceedings of the IEEE, vol. 92, no. 3, pp. 495–513, doi: 10.1109/JPROC.2003.823147, (2004)

[2] Moeslund, T.B., Hilton, A., Krüger, V., Sigal, L. (Eds.): Visual Analysis of Humans - Looking at People. Springer, doi: 10.1007/978-0-85729-997-0, (2011)

[3] Sigal, L., Black, M.J.: Guest editorial: state of the art in image- and video-based human pose and motion estimation. International Journal of Computer Vision, vol. 87, no. 1, pp. 1–3, doi: 10.1007/s11263-009-0293-2, (2010)

[4] Franco, J.S., Boyer, E.: Fusion of multiview silhouette cues using a space occupancy grid. In: Tenth IEEE International Conference on Computer Vision (ICCV), doi: 10.1109/ICCV.2005.105, (2005)

[5] Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, ISBN: 978-0-387-31073-2, (2006)

[6] Simas, G., de Bem, R., Botelho, S.: A 3d motion tracking method based on nonparametric belief propagation. In: IEEE International Conference on Robotics and Automation (ICRA), doi: 10.1109/ICRA.2013.6630786, (2013)

[7] Mikić, I., Trivedi, M., Hunter, E., Cosman, P.: Human body model acquisition and tracking using voxel data. International Journal of Computer Vision, vol. 53, no. 3, pp. 199–223, doi: 10.1023/A:1023012723347, (2003)

[8] Starck, J., Hilton, A.: Model-based multiple view reconstruction of people. In: Ninth IEEE International Conference on Computer Vision (ICCV), doi: 10.1109/ICCV.2003.1238446, (2003)

[9] Canton-Ferrer, C., Casas, J., Pardas, M.: Voxel based annealed particle filtering for markerless 3d articulated motion capture. In: 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, doi: 10.1109/3DTV.2009.5069645, (2009)

[10] 4D Repository - Inria Perception Group, http://4drepository.inrialpes.fr

[11] GrImage - Grid and Image, http://grimage.inrialpes.fr

[12] Moeslund, T. B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer vision and image understanding, 104(2), 90-126, doi: 10.1016/j.cviu.2006.08.002, (2006)

[13] Poppe, R.: Vision-based human motion analysis: An overview. Computer vision and image understanding, 108(1), 4-18, doi: 10.1016/j.cviu.2006.10.016, (2007)

[14] Sigal, L., Balan, A. O., Black, M. J.: Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. International journal of computer vision, 87(1-2), 4-27, doi: 10.1007/s11263-009-0273-6, (2010)

[15] Moeslund, T. B., Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding, 81(3), 231-268, doi: doi:10.1006/cviu.2000.0897, (2001)

[16] Agarwal, A., Triggs, B.: Recovering 3D human pose from monocular images. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(1), 44-58, doi: 10.1109/TPAMI.2006.21, (2006)

[17] Urtasun, R., Fleet, D. J., Fua, P.: Monocular 3D tracking of the golf swing. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 2, pp. 932-938), doi: 10.1109/CVPR.2005.229, (2005, June)

[18] Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In Computer Vision, 2009 IEEE 12th International Conference on (pp. 1365-1372), doi: 10.1109/ICCV.2009.5459303, (2009)

[19] Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on (pp. 1385-1392), doi: 10.1109/CVPR.2011.5995741, (2011)

[20] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, CVPR, doi: 10.1109/CVPR.2005.177, (2005)
[21] Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. arXiv preprint arXiv:1312.4659, (2013)
[22] Tompson, J. J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In Advances in Neural Information Processing Systems (pp. 1799-1807), arXiv:1406.2984, (2014)
[23] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM, 56(1), 116-124, doi: 10.1145/2398356.2398381, (2013)
[24] Breiman, L.: Random forests. Machine learning, 45(1), 5-32, doi: 10.1023/A:1010933404324, (2001)
[25] Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on (pp. 588-595), doi: 10.1109/CVPR.2014.82, (2014)
[26] Ionescu, C., Carreira, J., Sminchisescu, C.: Iterated Second-Order Label Sensitive Pooling for 3D Human Pose Estimation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on (pp. 1661-1668), doi: 10.1109/CVPR.2014.215, (2014)
[27] Lowe, D. G.: Object recognition from local scale-invariant features. In Computer vision, 1999. In International Conference on Computer Vison (Vol. 2, pp. 1150-1157), doi: 10.1109/ICCV.1999.790410, (1999)
[28] Elhayek, A., Stoll, C., Kim, K. I., Theobalt, C.: Outdoor Human Motion Capture by Simultaneous Optimization of Pose and Camera Parameters. Computer Graphics Forum (CGF), doi: 10.1111/cgf.12519, (2014)
[29] Sigal, L., Bhatia, S., Roth, S., Black, M. J., Isard, M.: Tracking loose-limbed people. In Computer Vision and Pattern Recognition, CVPR, doi: 10.1109/CVPR.2004.1315063, (2004)
[30] Sigal, L., Isard, M., Haussecker, H., Black, M. J.: Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. International journal of computer vision, 98(1), 15-48, doi: 10.1007/s11263-011-0493-4, (2012)
[31] Starck, J., Hilton, A.: Surface capture for performance-based animation. Computer Graphics and Applications, IEEE, 27(3), 21-31, doi: 10.1109/MCG.2007.68, (2007)
[32] De Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H. P., Thrun, S.: Performance capture from sparse multi-view video. ACM Transactions on Graphics (TOG), 27(3), 98, doi: 10.1145/1399504.1360697, (2008)
[33] Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., Ilic, S.: 3D Pictorial Structures for Multiple Human Pose Estimation, To appear in Computer Vision and Pattern Recognition, CVPR, doi: 10.1109/CVPR.2014.216, (2014)
[34] Vis5D , http://www.ssec.wisc.edu/ billh/vis5d.html
[35] Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 780–785, doi: 10.1109/AFGR.1996.557243, (1997)
[36] Caillette, F.: Real-Time Markerless 3-D Human Body Tracking. Ph.D. dissertation, University of Manchester, (2005)
[37] Dempster, A.P., Laird, M.N., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B, pp. 1–38, doi:10.2307/2984875, (1977)
[38] Herda, L., Urtasun, R., Fua, P.: Hierarchical Implicit Surface Joint Limits to Constrain Video-Based Motion Capture. In: European Conference on Computer Vision, doi: 10.1007/978-3-540-24671-8_32, (2004)
[39] Welman, C.: Inverse kinematics and geometric constraints for articulated figure manipulation. Master's thesis, Simon Fraser University, (1993)

M.Sc. Rodrigo de Bem obtained his master's degree in Electrical Engineering at University of São Paulo (USP) and received his degree in Computer Engineering at Federal University of Rio Grande (FURG). Currently Rodrigo is Assistant Professor at Federal University of Rio Grande (FURG). He has extensive experience with Computer Vision, Analysis of Algorithms, Machine Learning and Information Technology.

Eng. Maurício Goulart is graduated in Computer Engineering at the Federal University of Rio Grande (FURG) in 2013. Maurício is highly experienced in computer vision and algorithms.

M.Sc. Gisele Moraes Simas is graduated in Computer Engineering awarded by the Federal University of Rio Grande - FURG (2008) and she is master's at Computational Modeling from Federal University of Rio Grande - FURG (2012). Currently, she is a professor in the Computer Science Center of the Federal University of Rio Grande and has worked mainly in the following areas: Robotics, Artificial Intelligence, Probabilistic Algorithms and Computer Vision.

Dr. Silvia S. C. Botelho graduated in Electrical Engineering at the Universidade Federal do Rio Grande do Sul (UFRGS), where he also obtained a M.Sc. in Computer Science. Silvia holds a Ph.D. in Robotic from Laboratoire dAnalyse et Archtecture de Systèmes, Toulouse, France. She is currently an Associate Professor at Universidade Federal do Rio Grande - FURG, where she teaches courses related to automation and computer science and supervises Undergraduate and Graduate students. She is the Vice-Director of the Centro de Ciências Computacionais (C3-FURG), having coordinated the Graduate Program in Computer Modeling from 2007 to 2009. She is the author of more than 60 publications in scientific journals and books. Her scientific and academic experience in Automation and Computer Science emphasize on the sensor grids, intelligent systems and robotic. Silvia is the coordinator of several scientific and technological projects involving enterprises and government. She is a member of the Robotic Committee of Brazilian Computer Society.