Detecting End-Effectors on 2.5D data using Geometric Deformable Models: Application to Human Pose Estimation

Xavier Suau[a], Javier Ruiz-Hidalgo[a], Josep R. Casas[a]

[a] *Universitat Politècnica de Catalunya, 1–3, Jordi Girona, 08034 Barcelona, Spain*

Abstract

End-effectors are usually related to the location of limbs, and their reliable detection enables robust body tracking as well as accurate pose estimation. Recent innovation in depth cameras has re-stated the pose estimation problem. We focus on the information provided by these sensors, for which we borrow the name 2.5D data from the Graphics community. In this paper we propose a human pose estimation algorithm based on topological propagation. Geometric Deformable Models are used to carry out such propagation, implemented according to the Narrow Band Level Set approach. A variant of the latter method is proposed, including a density restriction which helps preserving the topological properties of the object under analysis. Principal end-effectors are extracted from a directed graph weighted with geodesic distances, also providing a skeletal-like structure describing human pose. An evaluation against reference methods is performed with promising results. The proposed solution allows a frame-wise end-effector detection, with no temporal tracking involved, which may be generalized to the tracking of other objects beyond human body.

Keywords:
depth image, range camera, end-effector, human pose estimation, extremities.

1. Introduction

Human pose estimation strategies are being widely applied to countless applications. Knowledge on the position of different body parts, specially that of the head and limbs, is the key aspect of many interactive systems. Human pose estimation techniques are extremely helpful for a wide range of applications, from simple pointing and scrolling on a menu to complex gesture recognition. In order to provide a truly immersive experience, marker-less body pose estimation is a must in this field [1, 2].

Recent innovation in consumer oriented depth cameras has received growing interest in marker-less human pose estimation. Such cameras provide a pixel-wise depth estimation of the recorded scene in a working range, the most common ones operating in a typical range between 0.5 and 5 meters. Therefore, such new sensors provide straight-forward 3D information of the scene. Since the acquired data is restricted to a single viewpoint, we denote it as *2.5D data*.

Geometric deformable models (GDM), proposed independently by Caselles *et al.* [3] and Malladi *et al.* [4], have proved performance and flexiblility at describing topology, as stated by Han *et al.* [5]. GDM have been widely applied in the field of image [6] and volume [7] segmentation and component analysis.

Even if the GDM theory is formulated on the continuum, it may be implemented in a discrete domain. An efficient and simple implementation of the GDM is known as the *Narrow Band Level Set* method (NBLS), introduced by Adalsteinsson and Sethian [8], which restricts computation in thin bands surrounding a zero level. Periodic updates of these bands gradually cover the full area (or volume) of the analyzed data set, preserving its topology. The NBLS method is defined for organized points in an evenly spaced grid (pixels in 2D, voxels in 3D), which limits accuracy due to resampling. Recently, Rosenthal *et al.* [9] have proposed an implementation of the NBLS method for unorganized 3D points, preserving their actual position.

In this paper, we propose an adapted version of the NBLS method in the context of 2.5D data. The objective is to exploit connectivities over the depth surface in order to extract topological features. More precisely, the proposed method locates the end-effectors of any 3D object. Since our work focuses on body pose estimation, the end-effectors are mainly the four extremities of a human body (Figure 1). However, other 3D objects have

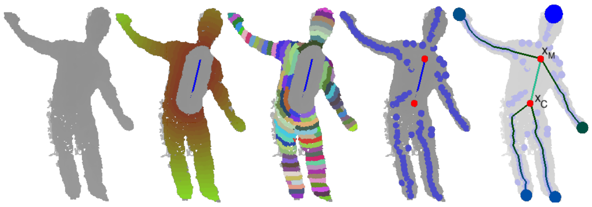

Figure 1: Summary of the steps involved in the proposed algorithm for human pose estimation. From left to right: Foreground mask, R-NBLS propagation, R-NBLS filtering, end-effector graph nodes and end-effectors found with the associated skeleton.

been studied, with special emphasis on the extraction of fingers of a human hand.

Human pose is inferred from the NBLS result and the obtained end-effectors: firstly populating a graph from the previously computed NBLS and, secondly, extracting extremity pose with a shortest path algorithm from the end-effectors.

The proposed method is evaluated against reference methods in Section 6.

2. Related Work

Marker-less human pose estimation has been classically carried out in multi-view environments, involving a considerable amount of regular cameras. Results in this area are impressive since complete 3D movement information is available. Gall *et al.* [10] go beyond pose estimation and cover possible non-rigid deformations of the body, such as moving clothes. Sundaresan and Chellappa [11] predict pose estimation from silhouettes and 2D/3D motion queues. Corazza *et al.* [12] generate a person-wise model which is updated through Iterative Closest Point (ICP) measures on visual-hull data. Pons-Moll *et al.* [13] combine video images with a small number of inertial sensors to improve smoothness and precision of the human body pose estimation problem. Nevertheless, these 3D capture environments are very expensive and cumbersome to setup, since they require precise calibration and, usually, controlled illumination conditions. In addition, the computational cost of 3D methods is prohibitive and real-time is hardly achieved.

On the other hand, very interesting works have studied how to extract human pose from single color cameras. In this direction, Guan *et al.* [14] obtain a synthetized shaded body. Body pose is estimated by searching into the learned poses, reflectance and scene lighting which most likely produced the observed pose. Yan and Pollefeys [15] recover the articulated structure of a body from single images with no prior information. In their work, trajectories of segmented body parts are mapped on linear subspaces to model the global body movement. Brubaker *et al.* [16] use a simple lower-body model based on physical walking movement called *Antropomorphic Walker*, proposed by Kuo [17]. Hasler *et al.* [18] propose a pose estimation algorithm which performs on mono and multiple uncalibrated cameras. Unfortunately, single color cameras inherently provide poor information, due to information loss originated from perspective projection. Single-camera based methods are usually very specific and hardly generalize to different kinds of movement, scenes and view points.

Human pose estimation from 2.5D data is a current research topic as a result of the mentioned increasing performance of depth sensors. Shotton *et al.* presented in [19] a method to classify body parts using a Random Forests strategy. Body pose is then inferred from the body parts' centroids. Baak *et al.* [20] combine local feature matching with a database lookup, achieving a fast and robust end-effector tracking. Zhu *et al.* [21] propose a tracking algorithm which exploits temporal consistency to estimate the pose of a constrained human model. Knoop *et al.* [22] propose a fitting of the 2.5D data with a 3D model by means of ICP. Grest *et al.* [23] use a non-linear least squares estimation based on silhouette edges, which is able to track limbs in adverse background conditions. While these three methods focus on upper-body pose, Plagemann *et al.* [24] present a fast method which localizes body parts on 2.5D data at about 15 frames per second. Ganapathi *et al.* [25] extend the work in [24] and extract full body pose by filtering the 2.5D data, using body parts' locations.

3. Preliminary concepts on 2.5D data

3.1. Input Point Cloud

2.5D data consists of a set of 3D points which corresponds to a sampling of the scene surface from the camera viewpoint. The object of interest is usually segmented using depth cues. In this work, the set of 3D points corresponding to the object of interest is denoted as the input point cloud D.

3.2. Apparent vs. physical area

Since 2.5D data is obtained from a single viewpoint, the concept of apparent area arises. A pixel on the captured image corresponds to a physical surface $A^p(z)$ which varies quadratically with the distance z to the camera. Indeed, one may find a function Γ^C relating the physical area of a given pixel to its apparent area, as a function of z. We propose to find it empirically (Figure 2) using a physical surface of known area (a DIN A2 sized surface at various distances).

Let S be a surface of unknown physical area A^S and an apparent area of N^S pixels. We assume that S is sufficiently perpendicular to the camera axis (we will see that this hypothesis is verified in this work), then $A^S \approx \Gamma^C(z^S) \cdot N^S$ is approximated as the physical area of a pixel at the average depth level z^S of S, times the number of pixels in S.

Knowing the physical area of a pixel avoids dealing with the apparent area, which is a great advantage of 2.5D sensors.

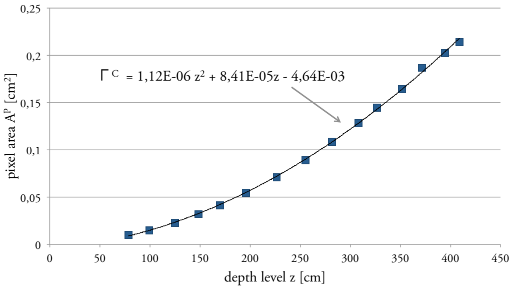

Figure 2: Empirical estimation of the law Γ^C for the Kinect camera, which gives the actual size of a pixel at a given depth level.

3.3. *Connectivity*

The raw data used in this paper consists of a point cloud of unorganized 3D points. In order to find connected regions in the point cloud, a connectivity condition should be defined. We state that a point p is (λ, ρ)-connected if the number of points in a ball of radius ρ centered at p is greater than λ. Thus, a region will be (λ, ρ)-connected if all its points are (λ, ρ)-connected too.

4. Geodesic distance estimation using Geometric Deformable Models and Narrow Band Level Sets

Geometric deformable models are based on the theory of curve evolution and the level set method [26]. The basic idea is to deform an initial curve or contour, which is registered to the data domain, depending on some pre-defined external and internal forces. Internal forces will expand the actual curve over the data keeping it smooth. External forces are computed from the available data and have an effect on the curve evolution. By way of example, imagine a drop of corrosive acid eating an object. The initial curve will be the drop at time zero, internal forces depend on the amount of acid in the drop, its corrosive power, etc. and external forces depend on the resistance of the underlying material. Thus, corrosion will slow down in resistive zones of the material and advance faster in areas more prone to corrosion.

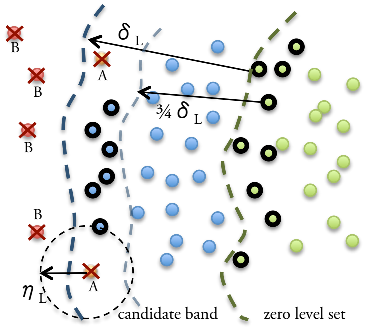

Figure 3: R-NBLS propagation example. The green points are the actual zero level set L_t^0, and those with a thick black boundary form the actual contour $C(s,t)$. The blue points in the middle are the candidate narrow band of with δ_L, with its contour also marked with thick point boundaries. Points labeled A (orange) are rejected because of the density condition in Equation (2). Points labeled B (red) are rejected because of the proximity condition also in Equation (2).

In our case, external forces are computed from the 2.5D data $D = \{\mathbf{x}_i\} \in \mathbb{R}^3$ and are defined to respect and preserve data features like topology or borders. We recall that D is the point cloud corresponding to the object of interest.

Let $\phi(\mathbf{x}, t) : \mathbb{R}^3 \to \mathbb{R}$ be a level set function whose sole purpose is to provide an implicit representation of the evolving curve. Let also $C(s,t) : \mathbb{R}^2 \to \mathbb{R}^3$ be a contour parameterized by s as the zero level set of $\phi(\mathbf{x}, t)$ and $L_t^0 \subset D$ be the subset enclosed by $C(s,t)$. Remark that C is parametrized in a two dimensional space, which is particular to the 2.5D data case. Equation (1) defines ϕ at a given time instant t. In the level set notation, time represents the advance of $C(s,t)$, $t = 0$ being the time-stamp of the initial curve. In order to efficiently perform nearest neighbor queries to evaluate the Euclidean distance $dist_E$ between points, D is organized as a *kd-tree* structure.

$$\phi(\mathbf{x}, t) = \begin{cases} 0 & \forall \mathbf{x} \in L_t^0 \\ \min\{dist_E(\mathbf{x}, C(s,t))\} & \forall \mathbf{x} \notin L_t^0 \end{cases} \qquad (1)$$

The objective is to make the contour C evolve over D preserving the topological properties of the latter. As cited in [5], the Narrow Band Level Set method is a simple solution to implement GDM evolution. An NBLS version for unorganized $\mathbb{R}^3$ points has also been presented in [9]. In the NBLS method, the level set function ϕ is evaluated in a thin layer surrounding the actual zero level set in order to update the zero level set for the next time instant. Such approach limits the number of calculations to these few surrounding points.

In this paper, we propose to add a density condition to the existing proximity condition. The role of this density condition is to filter the data, especially those points near depth edges. Using the above mentioned acid drop example, the density condition may be considered as an external force, since it is implicitly derived from the dataset. Propagation will slow down or stop in zones with low data density, and continue in highly populated zones. Thus, only those end-effectors *densely* connected to the main body will be considered, filtering sparsely represented or very thin ones.

We propose to update the zero level set according to Equation (2). We note that time t may be considered as a discrete time, where $t_k := t + k$ with $k \in \{0, \mathbb{N}\}$.

$$\begin{aligned} L_{t+1} &= \{\mathbf{x}_i\} \quad \text{if} \begin{cases} \phi(\mathbf{x}_i, t) < \delta_L & \text{(proximity)} \\ & \textit{and} \\ \mathbf{x}_i \text{ is } (\eta_L, \delta_L)\text{-connected} & \text{(density)} \end{cases} \\ L^0_{t+1} &= L^0_t \cup L_{t+1} \end{aligned} \tag{2}$$

The candidate narrow band is noted as L_{t+1} and its maximal width is δ_L, determined by the *proximity* condition. The connectivity property of $\mathbf{x}_i$ is used as *density* condition, ensuring that the space surrounding $\mathbf{x}_i$ is dense enough (at least η_L are close enough to $\mathbf{x}_i$), as shown in Figure 3. Therefore, δ_L and η_L are parameters of the proposed NLBS variant, called Restricted-NBLS or R-NBLS.

In order to complete the formulation, we should define how the contour C is updated in the 2.5D context. In practice, the candidate points L_{t+1} which are farther from the previous zero level set are taken as the new contour $C(s, t + 1)$ as shown in Equation (3).

$$C(s, t+1) = \{\mathbf{x}_s\} \in L_{t+1} \quad \textit{with} \quad \phi(\mathbf{x}_s, t) \in \left[\tfrac{3}{4}\delta_L, \delta_L\right] \tag{3}$$

Thus, iterating through Equations (1), (2) and (3) from an initial zero level set $L^0_{t_0=0}$, the sufficiently dense zones of D will be covered.

The geodesic distance between $L^0_{t_0}$ and a given point $\mathbf{x}_k$ which was added to L^0 at the time instant t_k (or iteration k) may be calculated with Equation (4).

$$dist_G(L^0_{t_0}, \mathbf{x}_k) \approx k \cdot \delta_L + \phi(\mathbf{x}_k, t_k) \tag{4}$$

The iterative R-NBLS method stops when the number of points N^C_t of the actual contour $C(s, t)$ is smaller than a given stop threshold. The proposed iterative framework stops when $N^C_t = 0$ (see the second shape in Figure 1 for an example).

4.1. Narrow band filtering by physical area

Zones of the scene being strongly oblique with respect to camera image plane will be sparsely sampled with 3D points, and will not be taken into account when constructing narrow bands. Consequently, the considered narrow band points are relatively parallel to the image plane axes, and the hypotheses in Section 3.2 are valid.

Narrow bands cover the visible and connected parts of the scene surface. However, a given band may be composed of points from both arms, since they contain points at similar $dist_G$ (Equation (4)). In order to separate points of a same band that belong to different contexts, narrow bands are filtered depending on their physical area. Indeed, a maximal area A^{max} is set, so that any narrow band b with a physical area A^b larger than A^{max} is divided into a maximum of α regions, as shown in Equation (5).

$$N_{regions} \leqslant \alpha = \lceil \frac{A^b}{A^{max}} \rceil \tag{5}$$

Let $N(b)$ be the number of points in b and $\bar{z}_b$ its mean depth. By applying the approximation of Section 3.2, there exists a maximum number of points $N^{max}(z_b)$ which keeps α constant at every depth level z_b (Equation (6)). Therefore, if $\bar{z}_b$ varies (*i.e.* a person moving towards or away from the camera), narrow bands will still be divided into no more than α regions.

$$N^{max}(\bar{z}_b) = \frac{A^{max}}{\Gamma^C(\bar{z}_b)} \tag{6}$$

Remark that a standalone A^{max} restriction could result in very small residual regions as shown in Figure 4. Therefore, besides the maximal area condition given by A^{max}, some additional restrictions must be verified in the narrow band filtering step. More precisely, the filtered regions must be (η_L, δ_L)-connected regions themselves (which implicitly forces a minimal region size of η_L points). After the filtering step, a set of regions is obtained for every narrow band, all of them being (η_L, δ_L)-connected. Some examples of the filtering step are presented in Figure 5.

5. Detecting end-effectors in a topologically weighted graph

End-effectors are topologically prominent protuberances in the object under study, restricted to the viewpoint of the range camera. A method to detect end-effectors from the result of the R-NBLS method is discussed in this Section (see Figure 1).

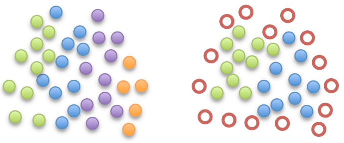

Figure 4: On the left, a narrow band filtering with a standalone A^{max} restriction, which results in a $N^{max} = 10$ maximal size. Note the residual region of 5 points (in orange). On the right, the filtering step with the additional restriction on connectivity with $\eta_L = 5$. The non-filled points have been filtered since they are not (η_L, δ_L)-connected.

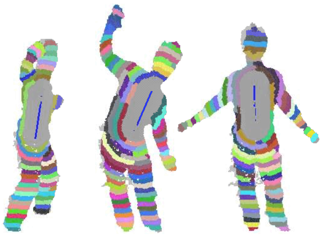

Figure 5: Three examples of narrow band filtering. The obtained regions are randomly painted. In this case, an $A^{max} = 70\,cm^2$ has been applied together with a $(\eta_L = 30, \delta_L = 4\,cm)$-connectivity.

5.1. Graph root

The proposed R-NBLS method requires the specification of an initial zero level set $L^0_{t_0=0}$ as starting point. Such origin region, called graph root, may be a single 3D point or a set of points. The definition of the graph root will strongly depend on the application. In this paper, the cases of a whole human body and a human hand are studied, with their specific graph roots.

The human body case. For this specific case, we propose to use a straight line as graph root (blue line in Figure 1). Such line connects the centroid $\mathbf{x}_C$ of D with $\mathbf{x}_M$, the latter being the midpoint between $\mathbf{x}_C$ and the head position $\mathbf{x}_H$, which is obtained with [27]. Those points placed at a given distance δ_0 of l_C are labeled as initial zero level set $L^0_{t_0}$, from which the R-NBLS propagation can start. Despite head estimation, which exploits some temporal information to increase tracking robustness [27], the rest of the proposed algorithm is frame-wise, without any temporal dependency.

5.2. End-effector graph construction

In general, R-NBLS filtered regions belong to prominent parts of the analyzed object (*i.e.* arms, legs) due to the band splitting considering connectivity (Section 4.1). The objective of this paper being that of finding end-effectors, it seems reasonable to use these context-wise regions.

In a consecutive phase to the narrow band filtering (Section 4.1), a graph is constructed on the filtered regions. The centroid of every region is taken as a graph node, with an associated creation time t_k coming from the R-NBLS propagation step explained in Section 4.

Graph nodes are linked in pairs with graph edges. We propose to only include those edges which link a source node n_i with time t_i and a sink node n_j with time t_j such that $i < j$, resulting in a directed graph (from inner to outer narrow bands). This way, any path constructed on the graph will be consistent with the node creation instants, not linking nodes with previously created ones.

A distance weight $w_{i,j}$ is calculated for every edge linking nodes n_i and n_j, with an additional distance penalty depending on the time elapsed between the creation of the nodes. Such penalty limits the construction of graph paths with strong jumps in creation time, this effect happening only in strictly necessary occasions. A 10% gain is added to the penalty $\alpha = 1.1$, so that it penalizes slightly more than an integer number of jumps. The proposed node weights are calculated with Equation (7).

$$w_{i,j} = \underbrace{|n_i - n_j|}_{distance} + \underbrace{(j - i - 1) \cdot \delta_L \cdot \alpha}_{penalty} \tag{7}$$

5.3. End-effector Estimation: Shortest Path from Farthest Level

A Dijkstra shortest path algorithm is run on the graph constructed in Section 5.2. End-effectors are extracted as the shortest paths from the farthest nodes to the graph root. Paths are searched starting at the node with greater t_k and ending at $\mathbf{x}_M$ (arms) or $\mathbf{x}_C$ (legs). If it does not exist any path from that node, successive nodes are taken as path sources by decreasing t_k until all paths have been found. Indeed, two paths ending at $\mathbf{x}_M$ are searched and labeled as arms, and two other paths are searched to end at $\mathbf{x}_C$, which are taken as legs. Since our work focuses on human body, end-effectors are often referred as extremities. Some conditions should be verified in order to accept a path as and end-effector of a human body.

- A path must have at least 3 segments, avoiding too short noisy detections.
- For a given graph, those nodes which belong to an already accepted path become unaccessible for further path estimations.
- Those nodes at a geodesic distance smaller than $30\,cm$ from the central line l_C are not taken into account, since we are looking for human extremities. Such restriction limits the detection of extremities

starting close to the body centroid. We assume this draw-back of the algorithm, as shown in Figure 8.

When both arms and both legs have been found, path search stops. The result computed by the presented algorithm constitutes the end-effector positions along with a skeletal-like structure describing the limbs. It should be noted that some poses may result in undetectable extremities, since their topological prominence is not clear enough. Therefore, only those end-effectors which are sufficiently detached from the body will be detected. Figure 1 presents a summary of the proposed algorithm, containing from left to right: raw depth estimation, R-NBLS propagation with $\eta_L = 80$ and $\delta_L = 4\,cm$, narrow band filtering with $A^{max} = 50\,cm^2$, graph nodes, and the obtained end-effector estimation on the right of the figure. The head has been found as in [27]. Nevertheless, note that it could be detected as an extra path from $\mathbf{x}_M$.

5.4. *Right and Left extremity decision*

Taking advantage of the extremity graph, a decision whether a limb corresponds to the right or left hand is taken. Remark that no temporal cues are involved in the decision.

For hands, the direction of first segment (t_0 to t_1) of every graph path (g_A and g_B) is calculated, obtaining two vectors $\mathbf{f}_A$ and $\mathbf{f}_B$. A simple decision depending on the orientation of $\mathbf{f}_A$ and $\mathbf{f}_B$ is performed, taking as right hand the path with $\mathbf{f}_i$ more oriented to the horizontal axis to the right (positive X coordinates). Remark that using the first graph segment is strongly invariant to the position of the end-effector associated to the graph path. A similar reasoningh is done for feet, using their two graph paths.

Yet being a basic classification approach, experimental results show that the proposed decision framework is effective (Section 6.3).

6. Experimental results

The following results have been obtained with a Kinect sensor which delivers both depth and color images at a frame-rate of about 25 fps with a resolution of 640 × 480 pixels. The color information is discarded and only the depth estimation is exploited in the experiments below. In Section 6.3, the proposed method is compared to two reference methods [19, 25].

6.1. *Effect of the parameterization on the human pose estimation*

Some aspects related to the tuning of the parameters of the proposed algorithm are shown in this section. Only δ_L and A^{max} are important parameters, η_L being a filtering parameter which may be kept invariant for a given sensor. For the Kinect camera, a value of 0.5 points per cm^2 during propagation has proven to be adequate. Therefore, the value of η_L is tightly related to the δ_L parameter ($\eta_L = 0.5 \cdot \pi \cdot {\delta_L}^2$).

The narrow band maximal width, δ_L, determines the resolution of the propagation, and also the areas which will be covered. A too low value of δ_L leads to propagation cuts, the advancing contour being too poor (few or no points) at some topologically narrow zones. On the other hand, a high δ_L value affects precision and extremities are less frequently detected (greater topological prominence is needed). Figure 6.a shows two R-NBLS propagations with $\delta_L = 4\,cm$ and $\delta_L = 10\,cm$. Both arms are close to the body and their topological prominence cannot be detected with a large δ_L.

The maximal area A^{max} controls the population of the end-effector graph. The smaller A^{max}, the more nodes are included in the graph, allowing more freedom for finding plausible paths. However, extremity detection is less stable with very low A^{max} values. On the contrary, if A^{max} is increased, the graph is poorly populated, resulting in a very rigid extremity estimation. In Figure 6.b, three examples are shown to illustrate the effect of various A^{max} values on the extremity detection.

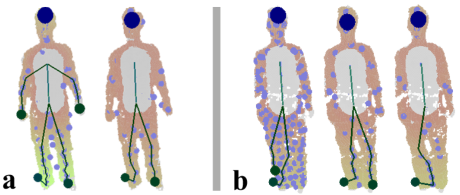

Figure 6: (a) Low values of δ_L provide more precision for the detection of topological prominence, even if the resulting paths are less straight (noticeable at the legs). (b) The maximal area parameter (left to right, $A^{max} = 20\,cm^2, 70\,cm^2$ and $200\,cm^2$) determines the population of the end-effector graph. Values of A^{max} which allow a proper detection of close legs are considered as trade-off values. Indeed, the narrow bands covering the legs will split into two regions, allowing the computation of two paths to $\mathbf{x}_C$. In the example, $20\,cm^2$ is too low and $200\,cm^2$ too high, while $70\,cm^2$ seems to be a convenient trade-off.

6.2. *Effect of the parameterization on the detection error and detection rate*

In order to evaluate the proposed method, more than 1000 depth images have been manually marked (hands and feet) as ground-truth. Only those extremities noticeable to the naked eye on the depth images are marked, avoiding guessing limbs' position from the input data.

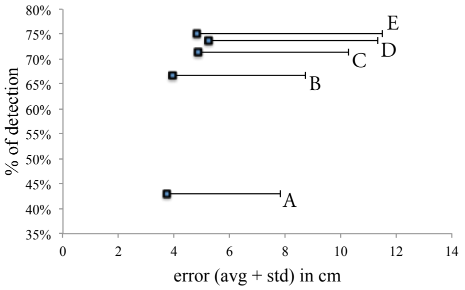

Figure 7: Detection error vs. detection rate (%) for various parameterizations (δ_L, A^{max}). $A = (10\,cm, 120\,cm^2)$, $B = (8\,cm, 70\,cm^2)$, $C = (6\,cm, 60\,cm^2)$, $D = (5\,cm, 50\,cm^2)$ and $E = (4\,cm, 40\,cm^2)$. Parameterization B seems to provide the best trade-off, with an average error of about 3.94 cm and a standard deviation of 4.79 cm for a detection rate of about 70%.

Results after various parameterizations are summarized in Figure 7. The error is measured as the Euclidean distance between the 3D ground-truth points and the estimated ones. In order to include the variance of the estimation, we plot $\bar{\varepsilon}$ and σ_ε on the horizontal axis, where $\bar{\varepsilon}$ is the average error and σ_ε the standard deviation. On the vertical axis, the percentage of detections over the number of ground-truth detections is plotted, taking into account only those detections with an error smaller than 30 cm. Therefore, parameterizations with higher detection rate and lower $(\bar{\varepsilon} + \sigma_\varepsilon)$ will provide the best results.

Experimental results show that parameterization $B = (\delta_L = 8\,cm, A^{max} = 70\,cm^2)$ obtains the best results with the Kinect camera. More precisely, it achieves extremity detection with an error smaller than 30 cm (maximal size of a foot or hand) in about 77% of the over one thousand annotated frames. The average error is $\bar{\varepsilon} = 3.94\,cm$ and its standard deviation $\sigma_\varepsilon = 4.79\,cm$. The percentage of detection increases while the error does so. Such effect shows the trade-off between obtaining many poor detections or less precise detections.

A summary of different situations has been presented in Figure 8 to show how the proposed human pose estimation algorithm performs with various human poses. Both easy situations (cross pose, farther left) and more difficult ones (punching, farther right) are properly solved, providing a 3D estimation of the position of the extremities. When extremities are not topologically prominent (*i.e.* third pose from the left), they are not detected. This is a logical draw-back of the proposed method. Remark that these estimations are obtained without any temporal tracking of the extremities, providing a frame-wise solution to the human pose estimation problem.

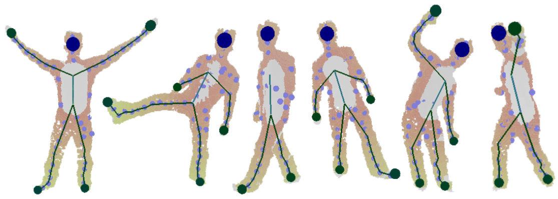

Figure 8: Some examples of the proposed human pose estimation algorithm. Remark that we do not perform any temporal tracking of the extremities, estimating human pose independently at each frame. One may notice how the strategy copes well with some adverse situations (*i.e.* . punching or bending the body). On the other hand, not prominent enough extremities are not detected with the proposed algorithm (*i.e.* third from the left, walking man).

6.3. Evaluation against reference methods

Two reference methods are used to evaluate the proposed method. In [19], Shotton *et al.* propose a body part classification by means of a Random Forest strategy. Ganapathi *et al.* propose in [25] a model-based approach exploiting temporal consistency. They also provide a dataset consisting of 27 sequences of increasing difficulty, recorded with a Time-of-Flight (TOF) camera. Moreover, ground-truth positions obtained with a motion capture device are provided. The experiments presented hereafter are obtained on the dataset in [25].

6.3.1. Classification Precision

The proposed method detects head, hands and feet of a human body. Therefore, we select the subset of markers in the dataset of [25] that represent these body parts. In Figure 9, a summary of the obtained average precision (AP) is provided. The proposed method outperforms [25], only being slightly surpassed in the cases of the head and right foot. The method in [19] obtains slightly better results in average. However, it is a specific classification method, whilst the other two methods are focused on detection, making no classification effort.

Regarding the average detection error, we compute the 3D error between the obtained end-effectors and the selected ground-truth markers. Results are presented in Figure 10, compared to the results obtained by [25]. The proposed method behaves in a similar manner over the whole dataset, obtaining an average error of about 9 cm even in the most challenging sequences (24-27). The method in [25], obtains a slightly better detection error in the first sequences, strongly degrading its results when facing the challenging sequences.

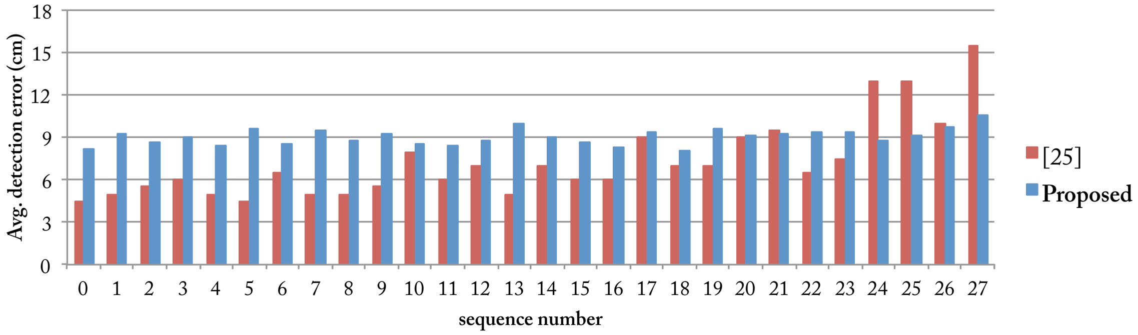

Figure 10: Detection 3D error comparison with [25] over the 27 sequences provided by [25].

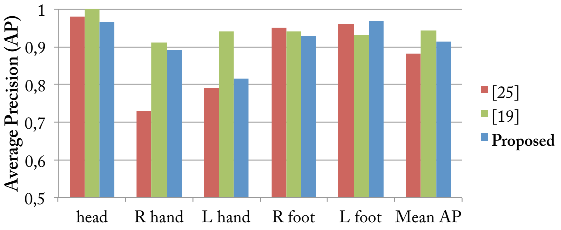

Figure 9: Average precision comparison with [19, 25] over the 27 sequences provided by [25].

As far as processing speed is concerned, the proposed method executes at about 57 fps using the dataset in [25] (176×144 resolution). In this paper, we have used a single core of an Intel Xeon CPU at 3GHz. The work in [25] achieves a frame rate of about $4-10\,fps$ with a specific GPU implementation. The method in [19] claims a $50\,fps$ execution frame-rate on full Kinect images (640×480) using an 8-core desktop CPU, and $200\,fps$ using a dedicated powerful GPU. In [20], a frame-rate of $60\,fps$ is achieved using a commercial CPU and a similar resolution.

6.4. Application to finger detection

The proposed algorithm to detect end-effectors may be applied to other objects besides the already presented human body case. With a suitable zero level set $L^0_{t_0}$ and an adapted parameterization (δ_L, A^{max}), the prominent end-effectors of any object may be found.

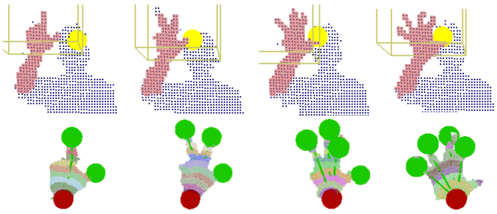

Figure 11: Application to finger detection. Hands have been obtained from the work in [27] (first row).

In [27], a fast and robust algorithm for head and hand detection is proposed. We utilize the obtained hand positions, onto which we apply the proposed R-NBLS end-effector detector, aiming to detect the number of extended fingers.

Some finger detection examples are shown in Figure 11, where one, two, three and four fingers are detected. In this example, the person is located about 2.5 meters far from the range camera. The initial zero level set $L^0_{t_0}$ is set as the centroid of the hand blob (graph root), and the parameterization $(\delta_L, A^{max}) = (2\,cm, 7\,cm^2)$.

7. Conclusion

We propose an end-effector detection algorithm based on topological propagation over 2.5D data. In order to carry out such propagation, Geometric Deformable Models are used. An extension of the Narrow Band Level Set formulation, named R-NBLS, is proposed to implement the GDM, adding a density restriction to better respect the topological properties of the analyzed object. The obtained level set provides a fast method to calculate geodesic distances over the original 2.5D data.

A skeletal-like structure of the object under analysis is estimated independently at each frame, without any temporal tracking of the extremities. In the specific human body case, such skeleton is tightly related to human pose. We provide a simple model to initialize the R-NBLS method in the case of human body.

The proposed method performs about 5× to 50× faster than [25], even in the adverse case of comparing our CPU implementation to the GPU implementation in the reference work. Our proposal is also faster than

[19], even if they use 8 CPU cores instead of the single core used in our proposal. Using a similar resolution, a frame-rate of 60 fps is achieved by [20], taking advantage of using a pre-computed training dataset.

R-NBLS outperforms the method in [25] in classification precision, and achieves similar results in terms of detection error. The method in [19] obtains a slightly better classification precision, taking advantage of a large training dataset and a dedicated classification task.

Other objects have been studied besides human body, with special interest in finger detection given a *hand object*. Fingers are detected at low resolutions, with a person placed about 2.5 meters far from the range camera.

Including topological borders or frontiers is one of the main foreseen points for further work. Such borders could help to improve the propagation in order to better respect the topology. We envisage to use color information and other local descriptors to complement the depth estimation, which will help solving ambiguities, increasing the detection rate.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o}248138.

This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación, under project TEC2010-18094

References

[1] T. B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Computer Vision and Image Understanding 104 (2-3) (2006) 90–126.

[2] R. Poppe, Vision-based human motion analysis: An overview, Computer Vision and Image Understanding 108 (2007) 4–18.

[3] V. Caselles, F. Catté, T. Coll, F. Dibos, A geometric model for active contours in image processing, Numerische Mathematik 66 (1) (1993) 1–31.

[4] R. Malladi, J. A. Sethian, B. C. Vemuri, Shape modeling with front propagation: a level set approach, TPAMI 17 (2) (1995) 158–175.

[5] X. Han, C. Xu, J. Prince, A topology preserving level set method for geometric deformable models, TPAMI 25 (6) (2003) 755–768.

[6] M. Maška, P. Matula, A fast level set-like algorithm with topology preserving constraint, in: Intl. Conf. on Computer Analysis of Images and Patterns, Springer, 2009, pp. 930–938.

[7] R. Whitaker, D. Breen, K. Museth, N. Soni, A framework for level set segmentation of volume datasets, in: International Workshop on Volume Graphics, Vol. D, 2001, pp. 159–68.

[8] D. Adalsteinsson, J. Sethian, A Fast Level Set Method for Propagating Interfaces, Journal of Computational Physics 118 (2) (1995) 269–277.

[9] P. Rosenthal, V. Molchanov, L. Linsen, A Narrow Band Level Set Method for Surface Extraction from Unstructured Point-based Volume Data, in: Intl. Conf. on Computer Graphics Visualization and Computer Vision, 2010, pp. 73–80.

[10] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, H.-P. Seidel, Motion capture using joint skeleton tracking and surface estimation, in: CVPR, IEEE, 2009, pp. 1746–1753.

[11] A. Sundaresan, R. Chellappa, Multicamera tracking of articulated human motion using shape and motion cues., IEEE Transactions on Image Processing 18 (9) (2009) 2114–2126.

[12] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, T. P. Andriacchi, Markerless Motion Capture through Visual Hull, Articulated ICP and Subject Specific Model Generation, International Journal of Computer Vision 87 (1-2) (2009) 156–169.

[13] G. Pons-Moll, A. Baak, T. Helten, M. Muller, H. Seidel, B. Rosenhahn, Multisensor-fusion for 3d full-body human motion capture, Elements (2010) 2–9.

[14] P. Guan, A. Weiss, A. Balan, M. Black, Estimating human shape and pose from a single image, in: ICCV, IEEE, 2009, pp. 1381–1388.

[15] J. Yan, M. Pollefeys, A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video., TPAMI 30 (5) (2008) 865–77.

[16] M. a. Brubaker, D. J. Fleet, A. Hertzmann, Physics-Based Person Tracking Using the Anthropomorphic Walker, International Journal of Computer Vision 87 (1-2) (2009) 140–155.

[17] A. D. Kuo, Energetics of Actively Powered Locomotion Using the Simplest Walking Model, Journal of Biomechanical Engineering 124 (1) (2002) 113.

[18] N. Hasler, H. Ackermann, B. Rosenhahn, T. Thormahlen, H. Seidel, Multilinear pose and body shape estimation of dressed subjects from image sets, in: CVPR, IEEE, 2010, pp. 1823–1830.

[19] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-Time Human Pose Recognition in Parts from Single Depth Images, in: CVPR, 2011, pp. 1297–1304.

[20] A. Baak, M. Meinard, G. Bharaj, H.-p. Seidel, C. Theobalt, M. P. I. Informatik, A Data-Driven Approach for Real-Time Full Body Pose Reconstruction from a Depth Camera, in: ICCV, 2011.

[21] Y. Zhu, B. Dariush, K. Fujimura, Controlled human pose estimation from depth image streams, in: ICCV Workshops, IEEE, 2008, pp. 1–8.

[22] S. Knoop, S. Vacek, R. Dillmann, Sensor fusion for 3D human body tracking with an articulated 3D body model, in: ICRA, IEEE, 2006, pp. 1686–1691.

[23] D. Grest, V. Krüger, R. Koch, Single view motion tracking by depth and silhouette information, Lecture Notes in Computer Science 4522 (2007) 719–729.

[24] C. Plagemann, V. Ganapathi, D. Koller, S. Thrun, Real-time identification and localization of body parts from depth images, in: ICRA, IEEE, 2010, pp. 3108–3113.

[25] V. Ganapathi, C. Plagemann, D. Koller, S. Thrun, Real Time Motion Capture Using a Single Time-Of-Flight Camera, in: CVPR, 2010, pp. 755–762.

[26] J. A. Sethian, Level Set Methods and Fast Marching Methods, Vol. 39 of Cambridge Monograph on Applied and Computational Mathematics, Cambridge University Press, 1999.

[27] X. Suau, J. Ruiz-Hidalgo, J. R. Casas, Real-Time Head and Hand Tracking based on 2.5D data, IEEE Transactions on Multimedia 14 (2012) 575–585.