3D Shape Estimation Based on Density Driven Model Fitting

Eugene Borovikov and Larry Davis

Department of Computer Science
University of Maryland
College Park, Maryland 20742, USA

mailto:{yab,lsd}@cs.umd.edu
http://www.umiacs.umd.edu/˜{yab,lsd}

Abstract

We introduce a generic and efficient method for 2D and 3D shape estimation via density fields. Our method models shape as a density map and uses the notion of density to fit a model to a rapidly computed occupancy map of the foreground object. We show how to utilize hierarchical (pyramid-like) object segmentation data to regularize a hierarchical model fitting. With primary focus on estimating 3D shapes of non-rigid articulated objects such as human bodies, we illustrate our approach with examples of efficient model fitting to 3D occupancy maps of human figures. We also discuss a number of extensions of our method to applications involving non-rigid object tracking and movement analysis.

1 Introduction

Describing shapes and motion of non-rigid objects is not a trivial task. While there does not seem to be a general recipe for determining the shape of a generic non-rigid object with subsequent tracking thereof, some approaches can efficiently utilize domain-specific knowledge to build models of objects and track them. The work described in this paper is primarily motivated by the research in 3D human body shape and movement analysis being carried out in a multi-perspective vision laboratory. The research aims at estimating 3D shapes of articulated objects (specifically, human bodies) and describing their motion.

There were numerous approaches to multi-perspective detection and tracking of articulated objects, in general, and human body figures in particular.

The simplest techniques [9, 10] used geometric primitives, e.g. ellipsoids, to approximate the object's constituent parts that are restricted by some high-level constraints on their positions, orientations and connections. The resulting models are fairly elaborate, yet not difficult to fit; but such methods could produce inexact approximations of the target objects primarily because not everything can be well approximated with a quadric.

More sophisticated methods used generalized cylinders [8, 12, 17], which sometimes provide a precision advantage over simpler geometric primitives, but still suffer from one significant drawback – bounded model pieces provided very little clue outside their boundaries as to where to drive the model fitting process, especially when the initial guess is poor. The methods also are sensitive to noise in the range/contour data on which they generally relied.

Skeleton based models are well suited for modeling articulated objects. Traditionally, they rely on medial manifolds (axes and/or surfaces) to reduce the volume of data to be dealt with [1, 2, 6, 11, 13]. While being good at preserving the target object's topology, they often are sensitive to segmentation noise and boundary curvature.

Below we briefly discuss papers that have particular relevance to the current research.

Gavrila and Davis [10] extract the foreground object contours and model a human body as a hierarchical set of super-quadrics. Their technique handles complex occlusions including multiple subjects, but makes many restrictive assumptions on the human body shape surface (using super-quadrics).

Bottino and Laurentini [4] utilize rapid volumetric reconstructions and build a piecewise linear approximation to the human body surface. This technique models the body by approximating the body surface, but does not use the volumetric aspect of the precomputed reconstructions.

Ong and Gong [14] rely on the object's contour and use stick-based models to describe the upper body. Their technique is shown to work well for human upper body in conditions where the foreground object is easy to segment. Golland and Grimson [11] use presegmented body blobs to extract fixed topology skeletons, but the method is not shown to be easily generalizable to 3D articulated objects. Chuang [6] uses a generalized potential field to extract a 3D object skeleton, but does not explicitly show how to apply this technique for human body modeling. Carr et al [5] introduce a precise and efficient method for the target object's surface reconstruction via zero level implicit surfaces of radial basis functions (RBFs), but their method relies on exact range data (e.g. from a laser rangefinder), and therefore is hard to apply to dynamic scenes.

Plänkers and Fua [16, 15] presented a framework for modeling and tracking human bodies using multi-layer models that utilize a potential field from a collection of soft meta-balls worn on a given skeleton. Their method fits an implicit iso-surface of the resulting potential field into the given depth maps and object silhouette contours. This fitting method is capable of realistic human body modeling, but it primarily targets object surface and requires a fairly good initial model guess and many meta-balls for complex surface modeling.

In this paper, we propose a generic and efficient method for 2D and 3D shape estimation via density fields. Our method models shape as a density map and uses the notion of density to fit a model to a rapidly computed occupancy map of the foreground object. We introduce a system for generating and manipulating complex density-based models out of application-specific density building blocks, and show how to use these models for both object recognition and shape visualization.

With primary focus on estimating 3D shapes of non-rigid articulated objects such as human bodies, we illustrate our approach with examples of efficient model fitting to 3D occupancy maps of human figures. We also discuss a number of extensions of our method to applications involving non-rigid object tracking and movement analysis.

2 Method motivation

Putting aside for a moment all the variety of shapes and movements that real-world non-rigid objects present us with, we focus primarily on estimating the 3D shape of such elaborate articulated objects as human bodies. Suppose, we have a human subject posing for us in a multi-perspective video capturing facility and we have a fast way of segmenting the body image e.g. by using the real-time volumetric segmentation algorithm described in [3]. The global task is to

1. locate the body and its constituent parts,
2. estimate their shapes,
3. track individual body parts, and
4. recognize and interpret (series of) gestures.

The scope of this paper aims at the first two subgoals, touches upon the third and leaves the fourth for future research. Going along these lines necessitates having an efficient shape estimation and representation framework suitable for further use in tracking and movement analysis applications. This, in turn, requires an economical, yet elaborate and extensible, shape description apparatus equipped with a procedure for efficient recovery of a 3D object shape from multi-perspective data.

2.1 A notion of density field

We propose to model real world objects via parametric *density fields*. An object's density field (or density function) is a non-negative scalar function (formally defined in Section 3) showing how much this object differs from its environment at each point of space. The object's density field is positive (or no less than a positive ε, for a threshold-based formulation) where the object is present, and zero (or less than ε) where it is not. Having the object's density field, we can derive and reason about many of the object's geometric and topological properties. For instance, a field's ε-level iso-surface is an estimate of the object's surface, while the field's ridges would correspond to the object's skeleton. Additionally, the density field iso-surfaces can be used to extract the object's connectivity information.

There are, certainly, many ways to define a density field for an object. For example, the left-most binary image of a printed character (fig.1) is that character's density function, which merely indicates presence or absence of ink at every pixel. This binary-valued function with a discrete 2D domain is given explicitly by the binary image itself. It clearly differentiates the object from its environment and captures the objects' shape, thus allowing one to use this map in e.g. correlation based template matching.

This function, however, is too strict (due to its discrete nature): it corresponds to a single image of the character that in reality can be typed or printed in a variety of blur levels and shape variations (e.g. scales,

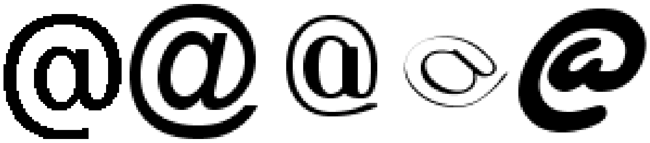

Figure 1. Different styles of the same character

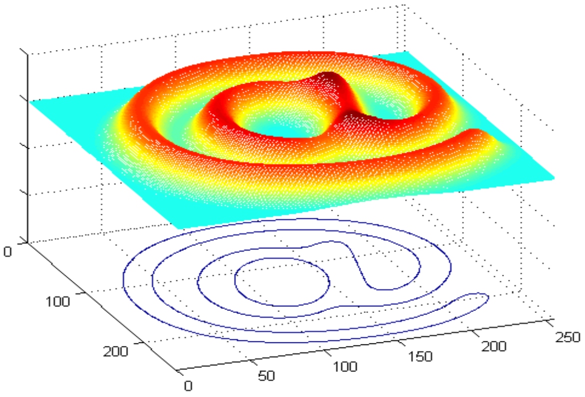

Figure 2. Example of a character density map surface with an iso-line plot

faces, emphases as in fig.1), which this particular density function does not capture.

Such deviations in the same object's imagery effectively introduce a class of similar objects that no single density function can capture. They, however, can be treated by a family of similar density functions, i.e. a *parametric* family of densities with a continuous domain and a set of parameters corresponding to position, orientation, scale, local and global density bandwidth, etc. An example plot of such density for the above example is given in fig.2.Such a density function can be given by a character contour parameterization $\mathbf{x}(t)$, $t \in R$, equipped with a 1-D density distribution for each value of t, i.e. $\phi_t(s)$, where axis s is collinear to the contour's local normal $\mathbf{n}_t$. Both the contour $\mathbf{x}(t)$ and the local density $\phi_t(s)$ can and should depend on some parameters (gathered in a vector $\boldsymbol{\theta}$) that make the resulting two-dimensional model density $f(\mathbf{x};\boldsymbol{\theta}) = \phi_{t(\mathbf{x};\boldsymbol{\theta})}(s(\mathbf{x});\boldsymbol{\theta})$ suitable to fit to any instance of the given character's image.

Note that one can analogously define parametric density models in 3D (or any dimensions, for that matter) and fit them to the given imagery. We shall see some examples of that in the sections to come.

2.2 Two-fold use of the density field

Going the model fitting route, we will see that density fields can be utilized in two ways:

- as target shape representation, and
- as model fitting driving force.

Aside from a compact yet flexible shape representation (seen e.g. in the above example), a correctly chosen parametric density field with infinite spatial support will be capable of driving the model fitting method (described in the next section) to some optimal solution (within an appropriate class).

In the following sections, we give a formal definition for a density function, describe the model fitting method and explain how to apply it to complex articulated objects and further focus on modeling human bodies. We also discuss an approach for defining and fitting hierarchical models to hierarchical volumetric data (such as octrees) and address the use of density-based models in 3D tracking.

3 Density based model fitting

To formalize the density based model fitting method, we need a density function that would model the object's density field, and help drive the model fitting process to a solution.

3.1 Formalism behind the method

We characterize the model of the object by a *density* function

$$f : R^n \times R^m \rightarrow R_+ \cup \{0\}$$

For any model configuration characterized by a parameter vector $\boldsymbol{\theta} \in R^m$, $f(\mathbf{x},\boldsymbol{\theta})$ gives a density value at any $\mathbf{x} \in R^n$. Additionally, for some positive M and D, and for $\forall\boldsymbol{\theta}$, the density function has to satisfy

$$\int_{R^n} f(\mathbf{x},\boldsymbol{\theta})\, dV_{\mathbf{x}} = M \qquad \text{(conservation)}$$

$$\max_{\mathbf{x}} f(\mathbf{x},\boldsymbol{\theta}) = D \qquad \text{(consistency)}$$

The (mass) *conservation* condition naturally follows from the requirement that the object's mass remains constant during the model fitting process. The (density) *consistency* condition ensures that the density peaks remain at a certain level. These conditions determine the solution's scale and ensures its feasibility. Refer to sec. 3.3 for further details.

Now, suppose that the object of interest has been segmented out from the source image and its occupancy map $V \subset R^n$ ($n = 2$ or 3) has been computed. Intuitively, we say that a model fits the object optimally when the object's occupancy map is at the model's densest region. To formulate this in mathematical terms, we define a measure of fitness

$$F(\boldsymbol{\theta}) = \int\limits_V f(\mathbf{x}, \boldsymbol{\theta})\, dV_{\mathbf{x}} \tag{1}$$

and solve the following optimization problem:

$$\begin{aligned} &\underset{\boldsymbol{\theta}}{\text{maximize}} F(\boldsymbol{\theta}) \\ &\text{subject to domain-specific constraints on } \boldsymbol{\theta} \end{aligned} \tag{2}$$

Although there are some (mostly artificial) situations for which it is possible to solve the above problem analytically, most practical problems that we encountered required density functions that made analytic solutions of (2) hardly tractable. Therefore, here we concentrate on numerical optimization and adapt existing minimization solvers (e.g. provided with optimization toolbox in Matlab) to obtain numerical solutions.

3.2 A simple example

Suppose, we are asked to fit a line segment to a three-dimensional box using the inverted distance as a density map. For $V = [0, .5] \times [0, .5] \times [0, 1]$, define

$$f(\mathbf{x}, \boldsymbol{\theta}) = (1 + d(\mathbf{x}, \mathbf{a}, \mathbf{b}))^{-1}$$

where $\boldsymbol{\theta} = (\mathbf{a}, \mathbf{b})$, $\mathbf{a}, \mathbf{b} \in R^3$ are the two end points of a line segment, $d(\mathbf{x}, \mathbf{a}, \mathbf{b}) = dist^4(\mathbf{x}, \mathbf{ab})$, and $dist$ is a distance from point $\mathbf{x} \in R^3$ to the segment $\mathbf{ab}$. Notice that our density is

- consistent, since $\forall \mathbf{a}, \mathbf{b}\ \max_{\mathbf{x}} \frac{1}{1+d(\mathbf{x},\mathbf{a},\mathbf{b})} = 1$, and
- conservative, provided $|\mathbf{a} - \mathbf{b}| = const.$

For this simple problem we can even relax the conservation requirement and solve the following optimization problem:

$$\max_{\mathbf{a},\mathbf{b}} \int\limits_V \frac{1}{1 + d(\mathbf{x}, \mathbf{a}, \mathbf{b})} dV_{\mathbf{x}}, \text{ subject to } \mathbf{a}, \mathbf{b} \in [0, 1]^3$$

A numeric solution to this problem yields

$$\begin{aligned} \mathbf{a} &= \begin{pmatrix} 0.24997 & 0.24997 & 0.01226 \end{pmatrix}^T \\ \mathbf{b} &= \begin{pmatrix} 0.25006 & 0.25006 & 0.99987 \end{pmatrix}^T \end{aligned}$$

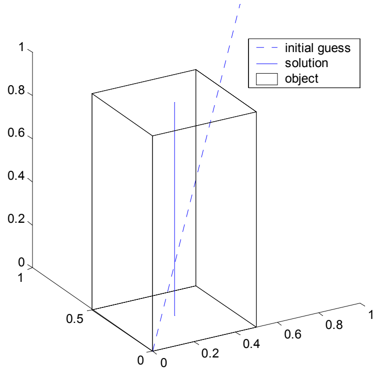

Figure 3. A line segment model fit for a 3D box

which is graphically shown in fig.3.

With a strict conservation constraint (and thus with more information about the target solution), e.g. $|\mathbf{a} - \mathbf{b}| = 1$, the above optimization problem yields a similar numeric solution

$$\begin{aligned} \mathbf{a} &= \begin{pmatrix} 0.25018 & 0.25018 & 0.02931 \end{pmatrix}^T \\ \mathbf{b} &= \begin{pmatrix} 0.24994 & 0.24993 & 1.02939 \end{pmatrix}^T \end{aligned}$$

but would converge faster.

3.3 Conservation and consistency vs scale

The conservation requirement essentially determines the solution's scale (with respect to the spatial variables), which can either be given up-front or estimated as above within certain boundaries. The consistency requirement deals with the field's range scale, and its importance is in avoiding the density field degeneration to something like a δ-distribution.

Since most practical density fields tend to be bounded and finitely integrable, both conditions are often easy to satisfy by re-scaling the field at hand. Indeed, suppose we are given a density g such that $\forall \boldsymbol{\theta} \in R^m$,

$$\int_{R^n} g(\mathbf{x}, \boldsymbol{\theta})\, dV_{\mathbf{x}} = A, \text{ and } \max_{\mathbf{x}} g(\mathbf{x}, \boldsymbol{\theta}) = B \tag{3}$$

Find $\alpha, \beta > 0$ such that for $f(\mathbf{x}, \boldsymbol{\theta}) = \alpha g(\beta\mathbf{x}, \boldsymbol{\theta})$, both conservation and consistency conditions hold:

$$\left\{ \begin{array}{l} \int_{R^n} \alpha g(\beta\mathbf{x}, \boldsymbol{\theta})\, dV_{\mathbf{x}} = M \\ \max_{\mathbf{x}} \alpha g(\beta\mathbf{x}, \boldsymbol{\theta}) = D \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} \alpha = \frac{D}{B} \\ \beta = \sqrt[n]{\frac{AD}{BM}} \end{array} \right.$$

This essentially gives us a *density normalization tool*. For $M = D = 1$ and some density g as in (3) choose $f(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{B} g\left(\sqrt[n]{\frac{A}{B}}\mathbf{x}, \boldsymbol{\theta}\right)$ satisfying the normalized conditions:

$$\int_{R^n} f(\mathbf{x}, \boldsymbol{\theta})\, dV_{\mathbf{x}} = 1 \qquad \text{(normalized conservation)}$$

$$\max_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta}) = 1 \qquad \text{(normalized consistency)}$$

We, therefore, are free to formulate our problems and work in either normalized (dimensionless) or physical (not normalized) or any other scale domain of our choice.

For some problems it is convenient to relax either conservation or consistency requirements by requiring only some upper bounds:

$$\int_{R^n} f(\mathbf{x}, \boldsymbol{\theta})\, dV_{\mathbf{x}} \leq M \qquad \text{(relaxed conservation)}$$

$$\max_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta}) \leq D \qquad \text{(relaxed consistency)}$$

This allows for some freedom in optimizing for scale.

3.4 A note on convergence

It should be evident that the method's convergence behavior is strongly affected by the choice of the density function. Different applications may treat the notion of density differently. For our method, we require the density function to be finitely integrable with respect to the spatial variables. It does not have to be differentiable or even continuous with respect to the parameters provided the optimization procedure can handle non-smooth objective functions. One should exercise particular caution when choosing to deal with finite support densities because the method won't have anything to drive it to a solution if the data happens to be outside of the density function's support.

3.5 Relevance to other techniques

Note that some traditional model fitting techniques (e.g. based on super-quadrics or medial sets) can be re-formulated in the given framework with corresponding density functions. Indeed, a surface fit, is nothing more but satisfying a constraint on a density isosurface, while a medial axes/surface fit corresponds to a constraint satisfaction on the density's ridges. Furthermore, the density based modeling provides a more general machinery to describe

- asymmetric body parts (bones are not always medial axes),
- complex topology (super quadrics do not handle holes well),
- natural non rigid deformation e.g. muscle expansion and contraction

and, perhaps, much more, all *within the same framework*.

4 Modeling articulated objects

Articulate objects such as human or animal bodies are complex. They consist of many sub-parts and have a lot of non-rigid tissue. Even most elaborate models of such objects are still limited in their ability to capture all the shape and/or movement variations an object can undergo. All of them target some subset of features modeled to a certain precision. Thus, in any modeling approach among everything else, there are always two things to address:

- resolution: features to model, and
- precision: how exact the model should be.

All that depends on the application at hand, but it is possible to develop a flexible framework allowing arbitrary model resolution and precision.

In a multi-perspective laboratory (e.g. [7]), the smallest reliably segmentable feature would be no smaller than a human fist. This constrains both the model's maximal resolution and its feasible precision, but even models representing the human body up to a hand are fairly complex and can be difficult to fit directly to the data. Naturally, when dealing with articulated objects it is often very desirable to utilize

- a composite density function by combining densities of the components, and
- a hierarchical model going from coarse (few parts) to fine (elaborate sub-parts) model resolution.

4.1 Cumulative densities

Since a target object may consist of several moving parts and sub-parts, we would like its corresponding density field to consist of several fields and dependent sub-fields:

$$f(\mathbf{x}, \boldsymbol{\theta}) = \sum_i f_i(\mathbf{x}, \boldsymbol{\theta}) \qquad (4)$$

where f_i's are the densities of the respective constituent parts. Note that with f_i's being conservative and consistent, the resulting density is conservative but only relaxed-consistent. Indeed, $\forall \boldsymbol{\theta} \in R^m$

$$\int_{R^n} f(\mathbf{x}, \boldsymbol{\theta})\, dV_{\mathbf{x}} = \sum_i \int_{R^n} f_i(\mathbf{x}, \boldsymbol{\theta})\, dV_{\mathbf{x}} = \sum_i M_i = M$$

$$\max_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta}) \leq \sum_i \max_{\mathbf{x}} f_i(\mathbf{x}, \boldsymbol{\theta}) = \sum_i D_i = D$$

The effect of this relaxed consistency is two-fold. On one hand it opens up a window for some scale variations, while on the other hand it validates possible and undesired part interpenetration. To avoid the latter in practical problems, we enforce a *cumulative consistency* constraint for some $\varepsilon > 0$

$$\max_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta}) = \varepsilon + \max_i \max_{\mathbf{x}} f_i(\mathbf{x}, \boldsymbol{\theta}) \tag{5}$$

4.2 Component densities

Individual component densities are the building blocks of the model's cumulative density function. Component densities usually represent individual body parts or the sub-parts thereof. A model designer can either specify (and implement) these functions directly or compose them as products of some well known (library) densities. In the latter case, a component density would be given as

$$g(\mathbf{x}; \boldsymbol{\theta}) = \prod_j g_j(\mathbf{x}_j; \boldsymbol{\theta})$$

where g_j is a library density with $\mathbf{x}_j$ being a d_j-dimensional sub-vector of $\mathbf{x}$, and $\sum_j d_j = n$. In the simplest case $d_j = 1$, and g_j is some $1D$ density. In more complicated cases when $d_j > 1$, the corresponding g_j can be inductively assembled via conditional and lower dimension densities e.g.

$$g_j\left(x_1, \ldots, x_{d_j}\right) = \\ h\left(x_{k+1}, \ldots, x_{d_j} | x_1, \ldots, x_k\right) h\left(x_1, \ldots, x_k\right)$$

4.3 Hierarchical fitting

For elaborate models, the resulting density function is complex and often results in a highly non-linear objective function that has a lot of local extrema, and thus is difficult to optimize. To overcome this difficulty, we propose a hierarchical coarse-to-fine modeling approach.

Often, occupancy maps can be efficiently computed in pyramid-like fashion, e.g. as quad- or oct-trees. In such cases, levels in the occupancy map can naturally be matched by corresponding levels in the model. As each subsequent level in the occupancy map hierarchy describes the object more accurately than its predecessor, each subsequent level in the model adds more details to the object's description.

Consider, for instance, human body shape volumetric reconstructions via silhouette visual cone intersections that can be done efficiently [3] in a multi-perspective lab. The program takes a multi-perspective snapshot of the foreground object and computes a set of corresponding silhouettes, which are used for visual cone intersection. A typical hierarchy of resolution-increasing volume reconstructions is shown in fig.4. Each level is represented by an octree.

We observe that the first three (coarsest) levels do not let us recognize anything. Starting at level 4 the foreground object is well localized and its major dimensions are evident. At level 5 one can start distinguishing the object's gross body parts. At level 6, the major body parts and joints are well localized and can be reliably detected. The higher (finer) levels of the hierarchy can bring some additional information for detailed recovery of small body parts such as hands and fingers.

To robustly identify the foreground object's major body parts (e.g. torso, legs, arms), one can utilize levels 4, 5, and 6 by fitting a hierarchical model reflecting the data detail, i.e. model complexity increases as more detailed information becomes available.

At its first level the model might aim only at the torso, so fitting e.g. a single plane patch based density should suffice. Once the torso's position, size and orientation are estimated, the model at its second level inherits the torso and additionally attempts to fit legs and arms using level 5 of the occupancy map. When the gross body parts are estimated, the model on the next level inherits the previous estimation and refines the detail of the foreground object by splitting parts into sub-parts, e.g. an arm into shoulder and forearm. This way, the most elaborate level of the hierarchy will have a good initial guess and thus less chance to get stuck in a local minima.

Peculiarities of initializing the current level from the previous level fit are often application specific and are determined by a particular choice of the constituent densities. As a heuristic, one can look at the difference between the current occupancy map and the occupancy map produced by the previous level fit to determine what wasn't covered by the lower level model; then the sub-parts can be initialized to compensate for that discrepancy.

4.4 Fitting in a sequence

Suppose we have a sequence of occupancy maps constructed using a multi-perspective sequence. The hierarchical fitting procedure can be used to fit a model to the initial frame of a sequence. For the subsequent frames of the same sequence one does not have to run the whole hierarchical procedure again. Instead, one can attempt to track the model over the multi-perspective sequence (via e.g. Kalman filtering) utilizing the top level model(s) from the previous step(s).

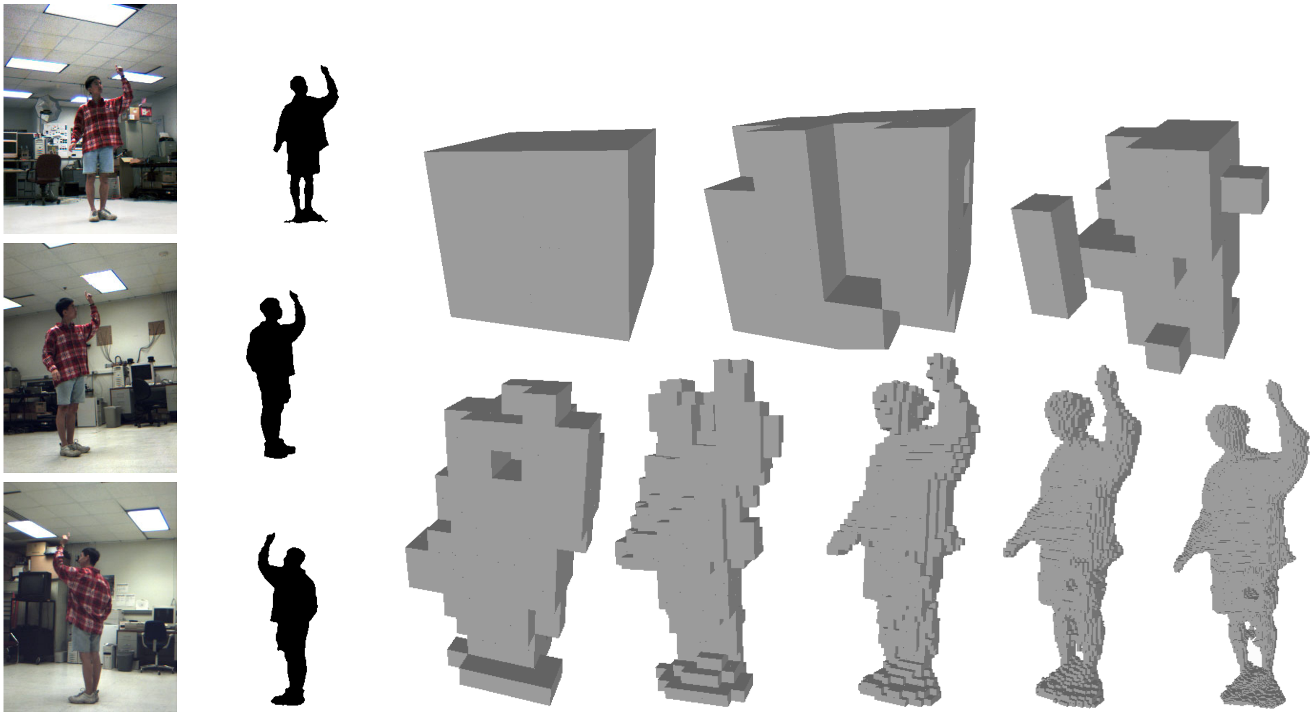

Figure 4. A source multi-perspective snapshot with corresponding silhouettes (left), and a volumetric reconstruction hierarchy (right)

More robust trackers may utilize several levels of the model hierarchy and employ the full hierarchy fitting only for the boot strapping step.

5 Experiments

For our experiments we used both synthetic and real occupancy maps. Synthetic maps were constructed by thresholding model densities while volumetric reconstructions of real human body figures were obtained in the Keck Lab using the real-time volume reconstruction technique.

5.1 Density based model of a human body

To utilize the hierarchical fitting mechanism, we employed a three-level hierarchical model. The first (coarsest) level models only the torso represented via an asymmetric soap-bar-like density function described below. The second level of the model brings head, legs and arms into the picture represented via tube-like densities also described in the next subsection. The third level refines the model further by splitting arms and legs in to their corresponding sub-parts using re-scaled tube densities.

5.1.1 Model building blocks

Simplest of all is the model of the head $h(\mathbf{x}) = \prod_{j=1}^{3} h_j(x_j)$, where h_j is some bell-shaped density (described below). In our experiments we used $e^{-(cx)^2}$.

An asymmetric *density bar* used for the torso is given by $t(\mathbf{x}) = \prod_{j=1}^{3} t_j(x_j)$, where $t_{1,2}$ are *plateau-based* densities and t_3 is an asymmetric *bell-shaped* density.

A *bell-shaped* density is any unimodal rapidly decaying density function b whose graph has the shape of a "bell" usually centered at 0, and not necessarily symmetric, i.e. $\int_{-\infty}^{0} b(x)\,dx \neq \int_{0}^{\infty} b(x)\,dx$, in general. In our experiments, we used

$$t_3(x;s) = \begin{cases} b((2-s)x) & x \leq 0 \\ b(sx) & x > 0 \end{cases}$$

where b is some symmetric bell-shaped density (we used $e^{-\pi x^2}$), and $0 < s < 2$ (we used $s = 0.7$)

A *plateau-based* density is given by

$$p(x;r) = \begin{cases} b(x+r) & x < -r \\ 1 & -r \leq x \leq r \\ b(x-r) & x > r \end{cases}$$

where b is some bell-shaped density with $\max_x b(x) = b(0) = 1$.

Arms, legs and their sub-parts are represented by *density tubes*. A tube density is a density $u(\mathbf{x}) = \prod_{j=1}^{3} u_j(x_j)$, where u_1 is a plateau-based density, and $u_{2,3}$ are (asymmetric) bell-shaped densities.

5.1.2 Cumulative density functions

The density building blocks described above are used to create so called *atomic* densities that usually correspond to the basic (non-subdividable) body parts such as torso. Each level in the model hierarchy is represented by a *cumulative density* function. A cumulative density function is given by

$$f(\mathbf{x}; \boldsymbol{\theta}) = \sum_i f_i(\mathbf{x}; \boldsymbol{\theta})$$

where each f_i is a *member* density. Each member can either be an atomic density or a cumulative density of a complex (multimember) part, in which case it in turn is computed using the above formula with respect to its own sub-members. Since densities are specified each in its own coordinate system, they all are responsible for the appropriate world-to-local coordinate transformations.

5.1.3 Model fitting concerns and details

The final optimization procedure has to observe the feasibility of body parts' positions and orientations via constraints on the model parameter vector $\boldsymbol{\theta}$, which is different for every level of the model hierarchy. The initial level is characterized by just six degrees of freedom for the torso. Level 2 needs to additionally describe rotation angles of arms (6) and legs (4), and level 3 adds four DOF more (by splitting arms and legs) resulting in the total of 20 parameters. We prevent infeasible body configurations by constraining the body part rotation angles, which translates into specifying the appropriate ranges for the parameters in $\boldsymbol{\theta}$.

As we noted before, our cumulative density function needs to satisfy the cumulative consistency condition (5). To enforce it, we punish excessive density by augmenting our cumulative density function $f(\mathbf{x}; \boldsymbol{\theta})$ as $\tilde{f}(\mathbf{x}; \boldsymbol{\theta}) = c(f(\mathbf{x}; \boldsymbol{\theta}))$, where

$$c(y) = \begin{cases} y + 1 & 0 \le y \le 1 \\ \frac{2}{1+4(y-1)^2} & y > 1 \end{cases}$$

As we can see in fig.5, the resulting function behaves like a shifted source density as long as the latter does not exceed 1, otherwise it sharply decreases causing low values in the objective function: $F(\boldsymbol{\theta}) = \int\limits_V \tilde{f}(\mathbf{x}, \boldsymbol{\theta})\, dV_{\mathbf{x}}$.

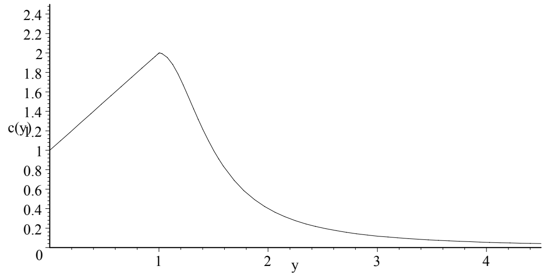

Figure 5. Augmenting the cumulative density

5.2 Results

The density models were fitted in the coarse-to-fine fashion with level 1 of the model being fit to a four-level deep octree. Below we give the results of model fitting. For both synthetic and real-world fits, the initial model was told approximate *up* and *front* directions. The rest of the parameters were estimated automatically. The final density fields were rendered as occupancy maps created by thresholding the density function.

Synthetic fits were invariably nearly perfect. Figure 6 shows source occupancy maps (top row) and their density based model fits (bottom row). As we can see at this scale of rendering, the model volumes are virtually indistinguishable from their source counterparts. For these experiments, the top level fits were done using source octrees six levels deep. The final rendering, however, was done with the octrees of depth eight.

Figure 7 presents density based model fits (bottom) into occupancy maps of a real person making pointing gestures (top). As one can see, the model captures the body shape and posture correctly (for the given resolution) and provides a reliable way to both locate and identify the body parts and describe the person's posture. The model did not aim to capture fine details of the human body, and thus hands and feet were not modeled, but could be by introducing as yet another level in the model hierarchy. Note also that applications targeting higher precision than this will have to refine density building blocks to reflect asymmetries in body parts and provide additional fine constraints on body part movements.

6 Conclusion

In this paper we proposed a generic and efficient method for 2D and 3D shape estimation via density fields. Our method models shape as a density map and uses the notion of density to fit a model to a rapidly

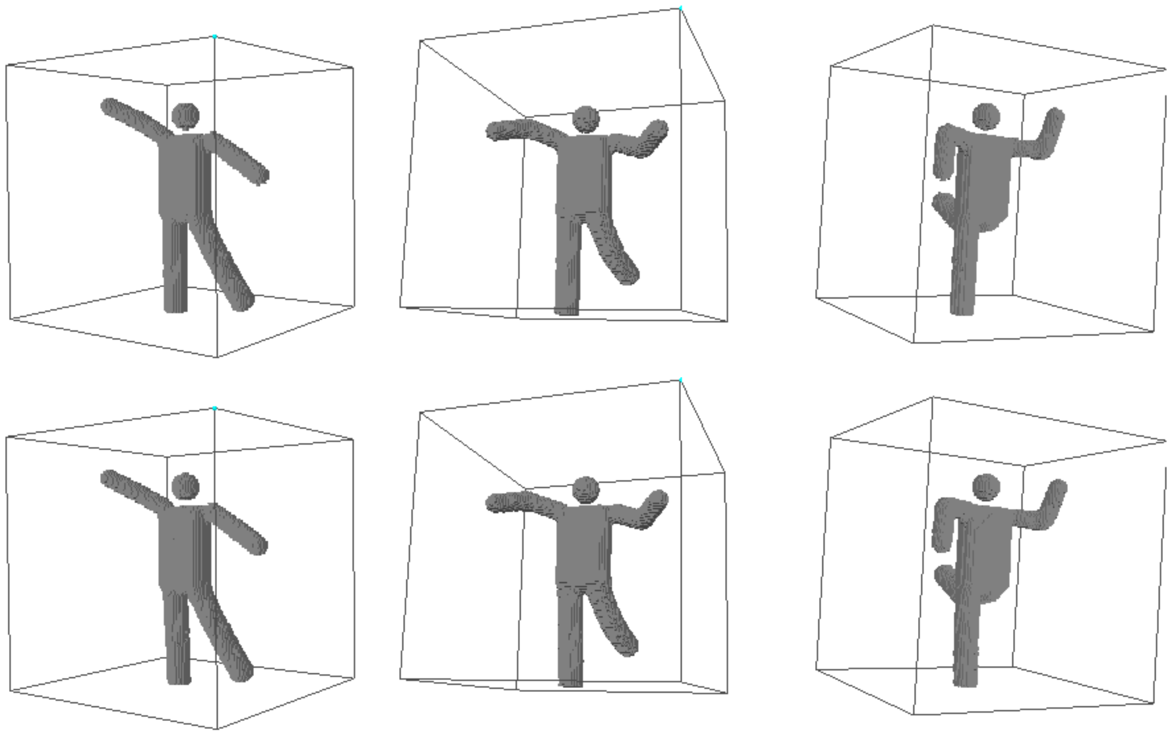

Figure 6. Synthetic images (top) and fitted density models (bottom)

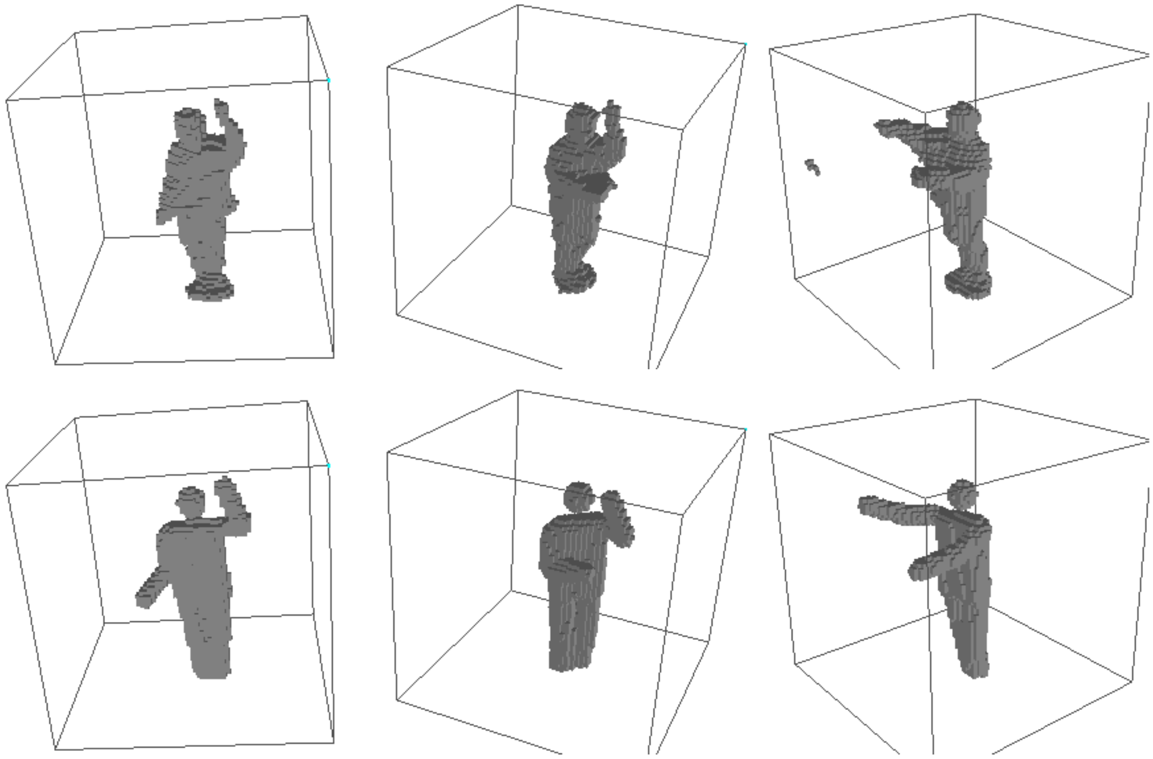

Figure 7. Fitting density models (bottom) to real images (top)

computed occupancy map of the foreground object. We introduce a system for generating and manipulating complex density-based models out of application-specific density building blocks, and show how to use these models for both object recognition and shape visualization.

The major contributions of our method are in

- providing a generic density-based shape modeling framework based on mass conservation,
- utilizing hierarchical model fitting to hierarchical segmentation data (quad- or oct-trees),
- allowing shape modeling to any precision via custom density building blocks.

Our approach differs from similar ones by

- producing models with relatively low number of degrees of freedom,
- efficiently working with volumetric or areal segmentation data,
- easily generalizing to any dimensions.

We consider this method to be important to the following areas:

- multi-perspective recognition and tracking of non-rigid articulated objects,
- physically realistic modeling and visualization,
- flexible and efficient shape representation.

Our future course of research will aim at extending the method for precise 3D model tracking, movement and gesture analysis. Besides that, the method has a potential to be adapted for such areas as robust segmentation in 3D non-rigid imaging by departing from binary occupancy maps to the domain of gray-level 3D images (and sequences thereof). It also can be considered as a tool for 3D (or N-D) cluster analysis and data compression via e.g. extending this method to handle non-parametric distributions.

Acknowledgement 1 *We express our thanks to NSF for sponsoring this research via NSF-EIA 01523672.*

Acknowledgement 2 *We would like to thank Nail Gumerov (UMCP) and Ramani Duraiswami (UMD) for providing useful feedback on our work.*

References

[1] J. August, A. Tannenbaum, and S. Zucker. On the evolution of the skeleton. In *ICCV99*, pages 315–322, 1999.

[2] J. Bloomenthal and C. Lim. Skeletal methods of shape manipulation, 1999.

[3] E. Borovikov and L. Davis. A distributed system for real-time volume reconstruction. In *Computer Architectures for Machine Perseption*. IEEE Computer Society, Sept. 2000.

[4] A. Bottino and A. Laurentini. A silhouette-based technique for the reconstruction of human movement. *Computer Vision and Image Understanding*, 83, 2001.

[5] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH Proceedings*, 2001.

[6] J.-H. Chuang, C.-H. Tsai, and M.-C. Ko. Skeletonization of three-dimensional object using generalized potential field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[7] L. Davis, E. Borovikov, R. Cutler, D. Harwood, and T. Horprasert. Multi-perspective analysis of human action. In *Proceedings of Third International Workshop on Cooperative Distributed Vision*, November 19-20, 1999.

[8] D. Dion, Jr., D. Laurendeau, and R. Bergevin. Generalized cylinder extraction in range images. In *3DIM97*, pages 6 – Geometric Processing, 1997.

[9] F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: Geometry from the bottom up. In *3DWS89*, pages 170–176, 1989.

[10] D. M. Gavrila and L. S. Davis. 3-d model-based tracking of humans in action: A multi-view approach. Proc. the Computer Vision and Pattern Recognition, 1996.

[11] P. Golland and W. E. L. Grimson. Fixed topology skeletons. Proc. the Computer Vision and Pattern Recognition, 2000.

[12] P. Havaldar and G. Medioni. Full volumetric descriptions from three intensity images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):540–545, May 1998.

[13] J. Lee and T. L. Kunii. Model-based analysis of hand posture. *IEEE Computer Graphics and Applications*, 15(5):77–86, 1995.

[14] E.-J. Ong and S. Gong. Tracking hybrid 2d-3d human models from multiple views. Proceedings of the IEEE International Workshop on Modelling People, 1998.

[15] R. Plänkers and P. Fua. Articulated Soft Objects for Video-based Body Modeling. In *Proc. IEEE Int'l Conf. on Computer Vision*, Vancouver, Canada, July 2001.

[16] R. Plänkers and P. Fua. Tracking and Modeling People in Video Sequences. *Computer Vision and Image Understanding*, 81(3), March 2001.

[17] M. Zerroug and R. Nevatia. Part-based 3d descriptions of complex objects from a single image. *PAMI*, 21(9):835–848, September 1999.