Component-wise modeling of articulated objects

Valsamis Ntouskos, Marta Sanzari, Bruno Cafaro, Federico Nardi,
Fabrizio Natola, Fiora Pirri, Manuel Ruiz
ALCOR LAB, Dipertimento di Ingegneria Informatica Automatica e Gestionale
{ntouskos,sanzari,cafaro,fnardi,natola,pirri,ruiz}@diag.uniroma1.it

Abstract

We introduce a novel framework for modeling articulated objects based on the aspects of their components. By decomposing the object into components, we divide the problem in smaller modeling tasks. After obtaining 3D models for each component aspect by employing a shape deformation paradigm, we merge them together, forming the object components. The final model is obtained by assembling the components using an optimization scheme which fits the respective 3D models to the corresponding apparent contours in a reference pose. The results suggest that our approach can produce realistic 3D models of articulated objects in reasonable time.

1. Introduction

The problem of modeling articulated objects, like people, animals and complex human artifacts has a long history in computer vision. Obtaining 3D models of objects from images is essential for many high-level vision tasks. Early approaches suggested a hierarchical composition of the object components, represented as generalized cylinders [4], *geons* [3], or superquadrics [30, 15], just to cite a few well known approaches to the structural descriptions theory. In these early works, components were modeled with parametric 3D shapes of few degrees of freedom, leading to limited resemblance to the actual geometry of the component.

With the popularization of accurate deformable models, introduced also by the computer graphics community (see [5] for a review), more realistic models of the components of an object are obtained. Recent works [32, 37, 38] have successfully shown how some types of animals can be modeled from a single image, relying mainly on the symmetry of the animal's shape. These approaches differ from the ones proposed in computer graphics (e.g. [25, 11, 21]), where input from the 3D artist is essential. The single view modeling methods, however, are not suitable for modeling articulated objects since some of their assumptions become not valid. In particular, the components of the object do not share the same plane of symmetry.

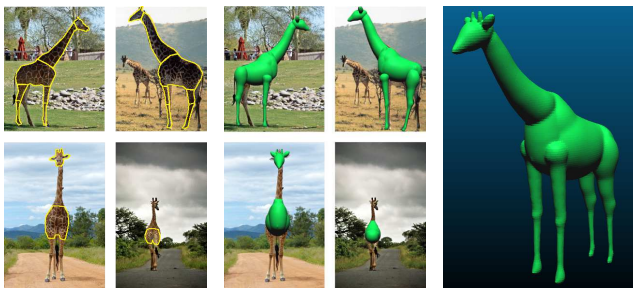

Figure 1: **Left:** Images of an animal downloaded from the web overlaid with segmentation masks, **Center:** modeled components overlaid on the input images, **Right:** final 3D model obtained with the proposed method.

In this work, we provide a solution to the problem of modeling articulated objects by explicitly modeling their components from various aspects. We consider a hierarchical decomposition of the object into components. Depending on the geometric complexity of the component, a different number of views is required for the modeling. For example, an animal's torso typically requires three to four representative views (left, right, front and back). Views of a component lead to the component aspects. An example of the decomposition in components and aspects is presented in Figure 2. From each aspect an approximate model of the imaged component is obtained using the deformation paradigm. Then, these aspect models are merged together to form a component. Components are typical of an object class and, in turn, are assembled considering a reference pose of the object, providing a 3D model of the whole object. Here, we assume that the object components are segmented out in the respective views. It is important to note that the different views need not correspond to the same physical object as far as objects belong to the same specific class. We focus our study on animals as they typically satisfy this property. An example of a 3D model obtained with our approach is shown in Figure 1.

The paper is organized as follows. In the next section we

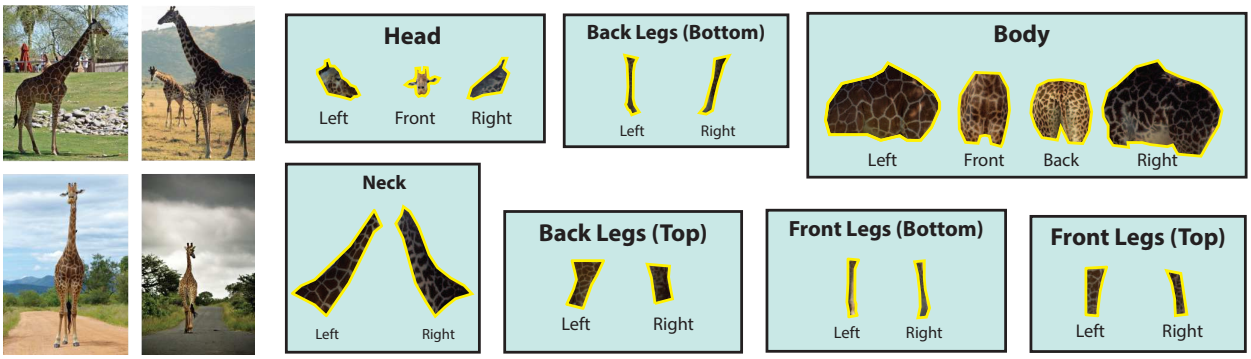

Figure 2: **Left:** Input images of Giraffes providing different aspects of each component, **Right:** representative aspects of each component of the Giraffe model.

review related work. In Section 3 we describe how components are modeled by their aspects. In Section 4 we show how components are assembled to form the final model. In Section 5 we evaluate the proposed method and Section 6 addresses conclusions and future work.

2. Related work

Geometric modeling of objects is becoming popular in computer vision. Following the deformation methods introduced in the pioneering work of Terzopoulos [36], shape generation from images provides good results by exploiting the contour generator. Single view modeling of objects with predefined genus and topology was introduced in [33, 32] using images of the same object family. Additional image cues have been considered in [28, 37] to model object classes from single views, and a similar approach has been taken by [38], exploiting the contour generator. A recent review is found in [29]. Multiple-view reconstruction of different object classes from few images has been successfully obtained using networks of objects with similar viewpoints [8], or for large scale shape reconstruction [39].

Differently from the 3D reconstruction methods we model an object not as a single rigid structure but as an articulated one. As opposed to SfM and factorization techniques, we model the views by deformation, we merge the obtained aspect models into components, and combine the components by a global optimization scheme, in order to estimate the view direction without requiring user input. The method allows us to join the components in several poses, this is the main novelty of our approach. The relation between the apparent contour and the contour generator, that we exploit here for assembling the components, has been studied since the early days of computer vision. Koenderink in [18] studies various properties of the contour generator based on the results of differential geometry, establishing in [17] a rule relating the curvature of the contour and the curvature of the surface, which is also investigated in [14]. A comprehensive study of the contour generator of evolving implicit surfaces is found in [31]. The problem of fitting 3D objects in their apparent contour has been treated in [9] where optimization is performed to find 3D-2D correspondences, considering a parametric representation of the surface and an estimation of the view direction, initialized by the user. The problem has been also treated in [6] for non-rigid surface sequences.

The final visual quality exploits surface smoothing. Level-set based methods have been widely used for this task (for a survey see [7]), based on an implicit surface representation, and have the advantage of topological flexibility. We follow the approach of [22], enabling Boolean graphics operations, for obtaining a model with no internal faces.

3. Modeling object aspects into components

We consider an articulated object to be formed by *components*, such as head, torso, limbs, where each component can be mapped into a viewer-centered *aspect*. An aspect represents a view of the component from the viewer vantage point [15], as illustrated in Figure 2. The number of components of an articulated object, can be freely determined, the choice being based on common sense. The number of views needed to model a single component depends on the regularity of its shape. Though, we do not rely on shape regularity because the component model is obtained from its aspects by optimization (see Section 3.2). Therefore, if a component is quite irregular, one would want to collect each of its idiosyncratic aspects.

The image selection task, leading to a choice of the components and their aspects, in the spotted views, requires some user input. Such as, for example, the judgment of what is needed to recover a good model. In principle few images are needed, and in our examples we used four images, as shown in Figure 2. This said, the complex problem of automatically determining the number of components and aspects of a natural kind is not faced in this work.

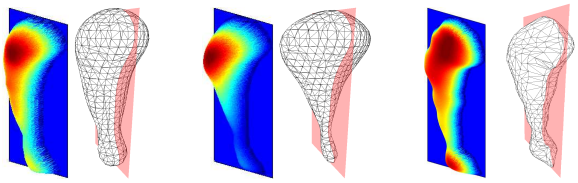

Figure 3: Comparison of the solutions (depth maps) and reconstructed surfaces (meshes) of a cat's leg, **Left**: obtained with (4), **Center**: without the load. (Best seen in colors). **Right**: with load, given noise on the contour segmentation.

3.1. Aspect modeling

Assume to have available a number N_I of images $I_1, \ldots, I_{N_I}$ showing different views of some articulated object category C, which is supposed to have N_c components. Let $\Omega_i \subset \mathbb{R}^2$ be the domain of image I_i, $i = 1, \ldots, N_I$, and assume there is a chart of the segments of all visible components in image I_i, as shown in Figure 2, as for example provided in PASCAL-Part dataset [12], as well as in [19, 20]. Each segment α_{ic} in an image I_i of the object C, defines an aspect of the specific component c. This aspect is mapped into a binary mask after translation and isometric scaling, keeping the proportions of the components w.r.t the original image. Let $T{:}\Omega_i \mapsto \Omega_{Tc}$ be the transformation applied to α_{ic}, then we define the mapping $A_{ic}{:}\Omega_{Tc} \mapsto \{0,1\}$, which returns precisely the binary mask of the transformed segment $\hat{\alpha}_{ic}$. The projection of the binary mask back into $\hat{\alpha}_{ic}$, is $A_{ic}^{-1}{=}\{(u,v){\in}\Omega_{Tc} \ |A_{ic}(u,v){=}1\}$. Let $\partial A_{ic} = \{(u,v) \mid |\frac{dA_{ic}}{du} + \frac{dA_{ic}}{dv}| > 0\}$, ($|\cdot|$ absolute value). We assume that ∂A_{ic} is a closed simple (Jordan) curve dividing the Euclidean plane in interior and exterior regions, where the interior is defined to be $int(A_{ic}) = \{A_{ic} = 1\}$, and it has a prescribed sense of rotation. We define $F(u,v)$ the distance field at point $(u,v) \in int(A_{ic})$, namely:

$$F(u,v) = \min_{\hat{u},\hat{v}} \{\|(u,v) - (\hat{u},\hat{v})\|_2 \ |(\hat{u},\hat{v}) \in \partial A_{ic}\} \quad (1)$$

Let $\mathbf{q} \in int(A_{ic})$ be the center of a circle bitangent to ∂A_{ic}, having radius $r_{\mathbf{q}}$, namely $\mathbf{q}$ is on the medial axis of $int(A_{ic})$, we define:

$$h((u,v),\mathbf{q}){=}\min_{\mathbf{q}} \{\|\mathbf{q} - (u,v)\| | (u,v){\in}int(A_{ic})\}{+}r_{\mathbf{q}} \quad (2)$$

To obtain the 3D model from A_{ic} we minimize the elastic energy deforming the distance between nearby points, which is driven both by internal forces, inducing local stretching and bending, and external forces. A surface $\boldsymbol{\varphi}{\subset}\mathbb{R}^3$, parametrized by the function $g{:}\Omega_{Tc}{\mapsto}\mathbb{R}$, is computed by minimizing the strain energy functional defined by the first and second fundamental forms [35], plus an external force G, or load. Energy strain linearization is attained by considering the first and second derivatives of g [5]. The

Algorithm 1: Aspects modeling

Input: Aspects $A_{ic}, i = 1, \ldots, N_{A_c}, c = 1, \ldots, N_c$, aspects parameters $\mathbf{Q}_\lambda, \mathbf{Q}_\beta$
Output: Aspect models $B_{ic}, i = 1, \ldots, N_{A_c}$, $c = 1, \ldots, N_c$

```
for c = 1 : N_c do
    for i = 1 : N_{A_c} do
        Generate a triangulation for A_ic;
        Choose the set of shape functions (at least
        quadratic) and the quadrature nodes;
        Assemble the stiffness matrix and loads vector
        using the quadrature rule;
        Find the weights of the shape functions
        solving the equation KX = H;
        Find the displacements g_ic using eq. (5);
        Compute mesh B_ic based on the triangulation,
        and closure by reflection, of φ_ic.
```

energy functional is:

$$E(g) = \int_{\Omega_{Tc}} \mathbf{g}_\lambda^\top \mathbf{Q}_\lambda \mathbf{g}_\lambda + \mathbf{g}_\beta^\top \mathbf{Q}_\beta \mathbf{g}_\beta - 2Gg \ \mathrm{d}u\mathrm{d}v \quad (3)$$

Here $\mathbf{g}_\lambda{=}(g_u, g_v)^\top$, $\mathbf{g}_\beta{=}(g_{uu}, g_{vv}, g_{uv})^\top$, $\mathbf{Q}_\lambda$ is a $2{\times}2$ matrix of stretching parameters, $\mathbf{Q}_\beta$ is a diagonal $3{\times}3$ matrix of bending parameters, assumed known, and G is the load:

$$G(u,v){=}\frac{F(u,v)}{h(u,v)} \left(\delta_1(u,v)\gamma_1{+}(1-\delta_1(u,v))\gamma_2\right) \quad (4)$$

Here F and h are defined in eq.(1,2), $\delta_1(u,v)$ is the indicator of ∂A_{ic} convexity at (u,v) and $\gamma_1, \gamma_2 \in \mathbb{R}_+$ are weights. This external force is applied to make the final surface growing steeper both near the boundary and where the initial mask is thinner and convex (see Figure 3). The scheme for finding the solution $g(\cdot)$ of the energy functional (3) is based on the Finite Element method, as described in [10], applied to the associated Euler-Lagrange equation. The approximation of the displacement $g(u,v)$, which minimizes the energy functional (3) is obtained as:

$$g(u,v) = \mathbf{X}^\top \mathbf{\Phi}(u,v), \quad (5)$$

Here $\mathbf{\Phi}$ is the coefficient matrix of the continuous shape functions, $\mathbf{X}$ is the matrix of the unknown weights, obtained by solving the following quadratic minimization problem:

$$\min_{\mathbf{X}} \left\{\mathbf{X}^\top \mathbf{K}\mathbf{X} - \mathbf{H}^\top \mathbf{X}\right\}, \quad (6)$$

with $\mathbf{K}$ the stiffness matrix and $\mathbf{H}$ the vector of the loads. To constrain the solution at the boundary ∂A_{ic}, homogeneous Dirichlet conditions are applied into the PDE problem formulation. A smooth closed surface B_{ic} for each aspect (segment $\hat{\alpha}_{ic}$) of component c of object C, as viewed in image

I_i, is obtained by joining φ_{ic} with its reflection along the z=0 plane, see Figure 3. Algorithm 1 describes the main steps involved.

3.2. Component building

Let $\mathcal{B}_c$ be the set of closed surfaces, obtained as described above, which we denote the aspect models of the component $c = 1, \ldots, N_c$. For each component there are N_{A_c} aspect models, namely, $\mathcal{B}_c = \{B_{1c}, \ldots, B_{sc}\}$, with $s \leq N_{A_c}$. To obtain a consistent model for c, the aspect models in $\mathcal{B}_c$ need to be combined. To achieve this we chose a reference model $B_{rc} \in \mathcal{B}_c$ and estimate the 3D transformation between each aspect model $B_{ic} \in \mathcal{B}_c$ and the reference model B_{rc}, as illustrated in Algorithm 2. Each aspect model B_{ic} is labeled with respect to the image I_i, it was obtained from, and with respect to the component c it is a view point of, hence we use feature points extracted from the image I_i (see Figure 2), to compute the relative transformation $T_{ri}^{(0)}$ between B_{rc} and B_{ic}. A refined solutions is then obtained by 2.5D registration.

Algorithm 2: Aspect registration

Input: Indexes of reference aspect models rc, $\mathcal{B}_c$, $\hat{\alpha}_{ic}$, $i = 1, \ldots, N_{A_c}$, $c = 1, \ldots, N_c$
Output: Transformation T_{ri} between reference B_{rc} and aspect models $B_{ic} \in \mathcal{B}_c$, $i = 1, \ldots, N_{A_c}$

1 **for** $c = 1 : N_c$ **do**
2 **for** $B_{ic} \in \mathcal{B}_c$ **do**
3 Detect a set of feature points F_{ic} in the segment $\hat{\alpha}_{ic}$, (e.g. by keypoints, SURF [2] features or similar);
4 Project F_{ic} on B_{ic} to obtain the 3D feature points X_{ic};
5 Find feature matches $F_{ic} \leftrightarrow F_{rc}$;
6 **if** $\#\text{matches} > 3$ **then**
7 Estimate 3D transformation $T_{ri}^{(0)}$ based on $X_{ic} \leftrightarrow X_{rc}$ up to an affine transformation
8 **else**
9 Ask user for manual initialization
10 Apply $T_{ri}^{(0)}$ on B_{ic};
11 Compute depth image $\bar{d}_{ic}$;
12 Dense 2.5D registration of $\bar{d}_{ic}$ w.r.t. d_{rc}.

The last step of Algorithm 2 (line 12) is a dense 2.5D registration between the depth image d_{rc} of the reference aspect and the depth image $\bar{d}_{ic}$ corresponding to the transformed i-th aspect of component c. In the following we drop the subscript c as reference is intended to the component c. The registration is obtained via the minimization problem

$$\min_{\xi_i \in \mathfrak{a}(3)} \|d_r - \bar{d}_i(\xi_i)\|_{L_1}, \tag{7}$$

with $\mathfrak{a}(3)$ the Lie algebra of the 3D affine transformation group and ξ_i a twist belonging to this Lie algebra. The objective function involved is non-smooth and non-linear in ξ_i. We consider a local convex approximation of the objective function by iterative linearization with respect to ξ_i and we then apply the Legendre-Fenchel transform, transforming the original minimization problem to a sequence of saddle-point problems. Optimization is performed in a coarse-to-fine framework to avoid local-minima. Let $\mathbf{q}$ be the dual variable, Q the union of pointwise L_1 balls, $\delta\xi_i^{(k)} = \xi_i - \xi_i^{(k)}$, $\mathbf{d}_r$ the vectorized reference depth image, and $\bar{\mathbf{d}}_i(\xi_i^{(k)})$ the vectorized depth image of aspect i transformed according to $T^{(k)} = \exp(\delta\xi_i^{(k)})T^{(k-1)}$. Let $\frac{d\mathbf{p}}{d\xi_i}\big|_{\xi_i^{(k)}}$ be the directional derivative of $\mathbf{p}(\xi_i) = \mathbf{d}_r - \bar{\mathbf{d}}_i(\xi_i)$ with respect to ξ_i evaluated at $\xi_i^{(k)}$. The saddle-point problem at the k-th iteration is

$$\max_{\mathbf{q} \in Q} \min_{\delta\xi_i^{(k)} \in \mathfrak{a}(3)} \mathbf{q}^\top \left(\mathbf{p}(\xi_i^{(k)}) + \frac{d\mathbf{p}}{d\xi_i}\bigg|_{\xi_i^{(k)}} \delta\xi_i^{(k)} \right). \tag{8}$$

A solution is computed by applying primal-dual optimization to estimate the saddle-point at each level.

The optimization significantly improves the registration provided that the initialization $\bar{d}_i$ is situated in the convex basin of the optimal solution. The final solution depends on the choice of the reference aspect and the order in which the remaining aspects are considered, however, given that N_c is a small number, the solutions are virtually equivalent.

Given the transformations, leading to a consistent registration of the aspect models, we merge them into a single component model. To achieve this, we first compute a volumetric representation of each model surface. We use the definition of Inner Product Field (IPF), as described in [22]. The IPFs grants an implicit representation of the aspect models B_{ic} and we can exploit the following result: given $n \geq 2$ implicit surfaces $\phi_1(x), \ldots, \phi_n(x)$, then $\phi_\cup(x) = \min(\phi_1(x), \ldots, \phi_n(x))$ is the union of their interior regions and corresponds to the envelope of the surfaces. As a final step, the component model is slightly smoothed to attenuate possible irregularities and artifacts. The smoothing is applied on the volumetric representation of the aspect model using the Level Set method according to the mean curvature flow [27]

$$\phi_t + V_n \|\nabla\phi\| = 0, \tag{9}$$

where $V_n = -b\kappa$ is the velocity field in the normal direction generated from the surface curvature κ, and $b \in \mathbb{R}$. A mesh is then generated by standard meshing techniques (e.g. [23]).

4. Assembling of the articulated object

Components are assembled in order to obtain a model of the entire object in a reference pose. In particular, we use

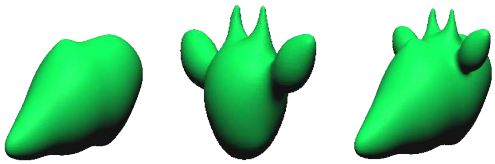

Figure 4: Aspects modeling and component building of the giraffe head. **Left**: side aspect, **Center**: front aspect, **Right**: component model.

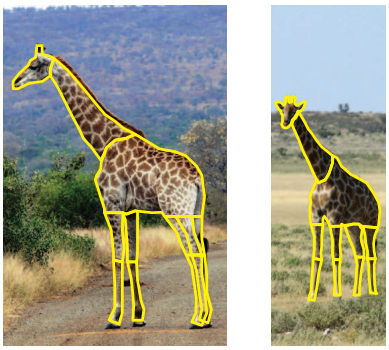

Figure 5: Two views of a giraffe in a reference pose with the overlaid component masks.

the apparent contours of the components in two or more views of the object in a reference pose, as the ones displayed in Figure 5. We assume here that all components are partially visible in the chosen views, that segments are available in each view and obtained by an orthographic projection. The visibility requirement can be relaxed as the number of views increases.

First, we recover the optimal transformation for each component, which makes its projection comply with the apparent contour. We treat this as a 3D-2D registration problem (see [24] for a review). We consider each component as a sufficiently smooth surface S (e.g. of class C^2) and the apparent contour is a planar contour γ. These two entities are related by the contour generator (CG), which is a space curve Γ, defined by the set of visible points on S, where the view direction $\mathbf{v}$ is locally tangent. The projection of Γ according to $\mathbf{v}$ produces γ up to a 2D similarity transformation. To register each 3D component, in its apparent contour, we find a view direction and the corresponding CG, which projects to a contour $\hat{\gamma}$ as similar as possible to γ.

Let $\mathcal{Y}(S)$ be a set of points sampled on S. Under the given assumptions, it suffices to identify two points $Y_1, Y_2 \in \mathcal{Y}(S)$ lying on Γ, to compute the view direction. Indeed, observe that Γ depends only on $\mathbf{v}$, and two points with non parallel normals $\mathbf{n}(Y_1)$ and $\mathbf{n}(Y_2)$ define the view direction up to a sign, as $\mathbf{v} = \mathbf{n}(Y_1) \times \mathbf{n}(Y_2)$. Given two points $\mathbf{y}_1, \mathbf{y}_2 \in \gamma$ we seek the corresponding points $Y_1, Y_2 \in \mathcal{Y}(S)$. We identify the best matches by minimizing the energy function

$$E(Y_1, Y_2; \mathbf{y}_1, \mathbf{y}_2) = \sum_{l=\{1,2\}} (E_{cg}(Y_l; \mathbf{y}_l) + E_{curv}(Y_l; \mathbf{y}_l)) + E_{ang}(Y_1, Y_2; \mathbf{y}_1, \mathbf{y}_2) + E_{dist}(Y_1, Y_2; \mathbf{y}_1, \mathbf{y}_2). \quad (10)$$

The term E_{cg} specifies that the points must lie on the CG corresponding to the estimated viewpoint. The last three terms take into account local geometric properties that the contour and CG have to satisfy. All these terms are invariant with respect to 2D similarity transformation, which is a computational bottleneck when considered. We examine now in detail each term.

E_{curv} is based on the relation between the curvature of the surface and the curvature of the apparent contour. First, the sign of the curvature of γ at point $\mathbf{y}$, $\kappa^{\gamma}(\mathbf{y})$ should match the sign of the Gaussian curvature of S at the corresponding point Y [18]. Additionally, $\kappa^{\gamma}(\mathbf{y})$, and the curvature of Γ at the corresponding point $\kappa^{\Gamma}(Y)$, satisfy the relation

$$\kappa^{\Gamma}(Y) = \sin^2\theta\, \kappa^{\gamma}(\mathbf{y}), \quad (11)$$

with θ the angle between $\mathbf{v}$ and the CG at Y [18, 13]. Based on this result, suitable bounds regarding the curvature of γ, Γ and S are provided by the following proposition:

Proposition. *Let S be a smooth surface and $\pi(\cdot)$ the projection operation. The curvature of the contour γ at a non-cusp point $\mathbf{y}$, the curvature of Γ at the corresponding point Y and the principal curvatures of the surface κ_1^S (minimum) and κ_2^S (maximum) at Y satisfy the inequality*

$$\kappa_1^S(Y) \leq \kappa^{\Gamma}(Y) \leq \kappa^{\gamma}(\mathbf{y}) \leq \kappa_2^S(Y), \quad (12)$$
$$\textit{with: } \mathbf{y} \in \gamma,\ Y \in \Gamma, \mathbf{y} = \pi(Y).$$

Proof. Consider a generic point $Y \in \Gamma$. We assume first that Y is not umbilical. The leftmost inequality is trivial as the curvature of Γ at Y, cannot be smaller than the minimum curvature of the surface at Y. The second inequality follows from (11). To show the last inequality we consider the osculating sphere O_Y of the surface at Y which has curvature $\kappa^{O_Y} = \kappa_2^S(Y)$. Regardless of the view direction, γ at $\mathbf{y}$ can at most locally lie on the projected contour of O_Y which is a circle with curvature κ^{O_Y}. Hence, the curvature of γ at $\mathbf{y} = \pi(Y)$ is locally bounded by the curvature κ^{O_Y} which is equal to $\kappa_2^S(Y)$. If the point is umbilical then all equalities trivially hold. □

Corollary. *Considering a point $\mathbf{y} \in \gamma$, a region $R \subseteq S$ is an admissible region of the corresponding point $Y \in \Gamma$ iff $\kappa_1^S(\mathbf{Z}) \leq \kappa^{\gamma}(\mathbf{y}) \leq \kappa_2^S(\mathbf{Z})$, $\forall \mathbf{Z} \in R$ and the sign of $\kappa^{\gamma}(\mathbf{y})$ matches the sign of the Gaussian curvature G^S in R.*

In the following for brevity we omit the explicit relation with the surface/curve points. Based on the previous result

the curvature term can be expressed as

$$E_{curv} = \omega_\kappa \, D_{[\kappa_1^S, \kappa_2^S]}(\kappa^\gamma) + \omega_G \max(-\operatorname{sgn}(G^S \kappa^\gamma), 0), \quad (13)$$

with $D_{\mathcal{J}}(v) = \min_{w \in \mathcal{J}}(\|v - w\|)$ and $\omega_\kappa, \omega_G > 0$ weights relating the terms.

The term E_{ang} expresses the fact that the angle between the normals $\mathbf{n}(Y_1)$, $\mathbf{n}(Y_2)$ matches the corresponding angle on the apparent contour. The same holds for the angle between each of the normals and the connecting segment $(Y_2 - Y_1)$ projected on the plane spanned by the normals. Let c the cost function that penalizes differences between the corresponding angles (e.g. $c(\theta, \phi) = \tan(|\theta - \phi|)$), we define

$$E_{ang} = \omega_n c(\theta_n, \theta_\eta) + \omega_b c(\theta_B, \theta_b), \quad (14)$$

with θ_n, θ_η the angles between the 3D and 2D normals respectively, θ_B, θ_b the angles between the base segment and one of the normals in 3D and 2D respectively, and $\omega_n, \omega_b > 0$ the relative weights. The term E_{dist} is defined as

$$E_{dist} = \omega_d \left(\frac{\|Y_1 - Y_2\|}{d(S)} - \frac{\|\mathbf{y}_1 - \mathbf{y}_2\|}{d(\gamma)} \right)^2, \quad (15)$$

with $d(\cdot)$ the diagonal length of the corresponding entity's bounding box, and $\omega_d > 0$ the relative weight.

Finally, the term E_{cg} is taken equal to the maximum penetration depth of the view ray passing through Y with respect to S and specifies the constraint that Y is on Γ.

We find the global minimum of the energy function with a branch-and-bound search strategy [34, 26]. First, we find the two points on γ which result into the most restricted region on S based on the previous corollary, and use them as initial points for the search. The pair of points which corresponded to the lowest energy value returns the view direction $\mathbf{v}$. The remaining 2D similarity transformation is then recovered by applying a shape matching technique between the resulting contour and the measured one (see [16]). This procedure gives the relative pose of each component with respect to the view. Not depending on all the points of the apparent contour, it is robust with respect to the visible portion of the contour and the shape of the 3D component. The solution can be refined by performing an iterative LSE minimization. We should note that the assembling step is robust with respect to noise as the components are smoothed before it is applied. An example is shown in Figure 3.

By registering each component in the given view we recover their relative position with the only exception of the translation in the viewing direction. We solve this ambiguity by using the other views. In particular since the object is imaged in the same pose from two or more known views, the depth ambiguity is resolved. A single model is computed from the assembled components by following the steps presented at the end of Section 3.2.

5. Evaluation

Modeling time The implementation of the proposed method consists of a mixture of Matlab and CUDA code. In particular, 2.5D registration of the modeled aspects, IPF computation and surface smoothing of the models are implemented in CUDA, while aspect modeling and component assembling are implemented in Matlab. A report of the time required for computing the models shown in this section is presented in Table 1.

Model	AM [sec]	CB [sec]	CA [sec]	Sm [sec]	Total [sec]
Cat	532	1.8	1942	0.09	2521
Dog	514	2.1	1026	0.08	1855
Cow	598	2.2	1311	0.10	1919
Sheep	426	1.9	1417	0.07	1826
Hippo	577	1.8	1514	0.07	2008
Giraffe	479	2.2	1410	0.06	1901
Kangaroo	441	2.0	1396	0.09	1723
Standing Horse	484	1.9	1613	0.05	2017
Landing Horse	505	2.1	1855	0.07	2090
Rearing Horse	494	1.9	1951	0.06	2034

Table 1: Modeling time report (AM-aspect modeling, CB-component building, CA-component assembling, Sm-smoothing).

The experiments were performed on a PC equipped with an Intel i7 3.6GHz CPU, 16GB RAM and an NVIDIA GTX970 graphics card. All models presented in the section have been modeled from four input images. Further results are presented in the accompanying video.

Model comparison We performed an extensive comparison of models obtained with our method using images taken from the web, with respect to models downloaded from the web. All images were taken from Flickr, while most of the downloaded models were obtained from the 3D warehouse of SketchUp, the rest have been taken from other repositories. We evaluated the similarity of our models with respect to the downloaded ones using two different similarity measures, the Hausdorff distance [1] and the normalized symmetric difference. We considered our model as reference and preprocessed the models taken from web to make the results comparable. Preprocessing consisted of the following steps: (a) model clean-up; remove internal faces, recover manifoldness and close holes; (b) manual orientation w.r.t. reference model; (c) automatic non-isotropic scaling for matching the bounding box with the reference model.

The Hausdorff distance was computed directly on the meshes of the models. For the symmetric difference we used the volumetric representation obtained via IPF. The distance is computed as the difference between the number

of voxels in the union and the number of those in the intersection of the two volumes, normalized by the total number of voxels. The results of the comparison are presented in Figure 6, where numbers correspond to the average values of the distances w.r.t. all the downloaded models of each class (3-4 models). These results show that the models computed with our method actually represent the modeled class. Indeed, the average distance with respect to the downloaded models of the same class is consistently smaller in comparison to the distances with respect to the other classes.

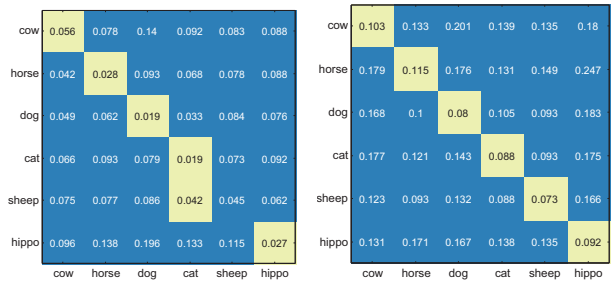

Figure 6: Model comparison (smallest values are highlighted); normalized symmetric differences (**left**) and normalized Hausdorff distances (**right**) between the models.

For a more objective evaluation, we applied the proposed approach to images of 3D models downloaded from the web. In particular, we generated images of the rendered 3D models from four vantage points, on which the segmented aspects were extracted. In this way, the downloaded models acted as ground truth with respect to which our models were compared using the normalized Hausdorff distance. The results of this comparison are presented in Figure 8 and in Table 2, where the mean values are given. We should note here that as this procedure allowed us to easily obtain two images of the object in more "unstable" poses, we were able to model the objects in different poses, as seen for example for the horse (standing, landing and rearing poses).

Cat	Dog	Cow	Sheep	Hippo
0.012	0.012	0.030	0.040	0.013
Giraffe	**Kangaroo**	**Standing Horse**	**Landing Horse**	**Rearing Horse**
0.018	0.023	0.016	0.028	0.020

Table 2: Mean normalized Hausdorff distance between the models reconstructed with our approach and ground truth.

Perceptual study Because of the nature of the problem, similarity distances may not always be representative. To further evaluate the quality of our models we performed a perceptual study with the help of volunteers.

Ten volunteers who did not know the purpose of the study participated in the experiment. Six participants were male and four female, 60% had from 22 to 25 years and

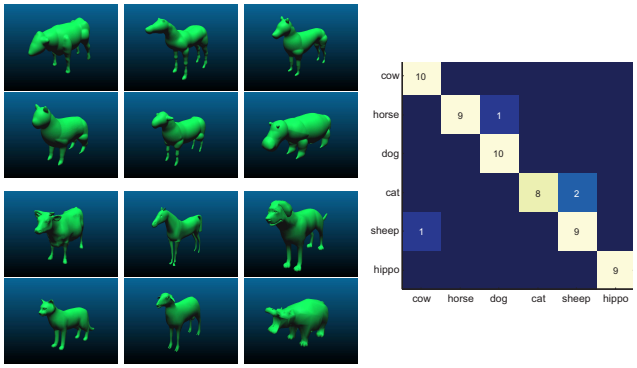

Figure 7: **Left**: Animal models used in the perceptual study. *Top group:* models computed with our method; *Bottom group:* models downloaded from the web. **Right**: Confusion matrix from the perceptual study.

40% from 25 to 29 years. Finally, three subjects reported corrected-to-normal vision and the rest normal vision.

The models presented in Figure 7 (left) were used for conducting the study. Participants were invited to ask questions before the experiment. After providing the necessary information and consent the task was presented to the participants:

"*Various 3D models will be shown on the screen during the experiment. For each model, you need to identify the corresponding animal and give a mark for its quality. You can interact with the model for as long as you prefer before answering.*"

The models were presented on the screen with a uniform green shaded material on blue background, as shown in Figure 7. The participants marked the answers on a special form, where the animal class could be specified freely and a scale of discrete values from 0 to 5 was used for evaluating the quality of the model. The models were presented in a random order to avoid bias caused by repeated ordering.

We consider the null-hypothesis H_0 that participants randomly selected the animal class, while the alternative hypothesis H_1 is that users correctly recognized the animal. Cross-tabulation was performed on the answers provided by the participants regarding the class of animal represented by our models and the resulting confusion matrix is shown in Figure 7 (right). One can observe that the participants almost always identified successfully the animal class. In fact, the null hypothesis is rejected as the chi-square value is $\chi^2 = 247$, corresponding to a practically vanishing p-value. It is important to note that the participants did not know in advance the classes of animals involved. This justifies also the last row of the confusion matrix, as one participant recognized the hippo as a pig.

The distribution of votes given by the participants for the model quality is presented in Figure 9. The models

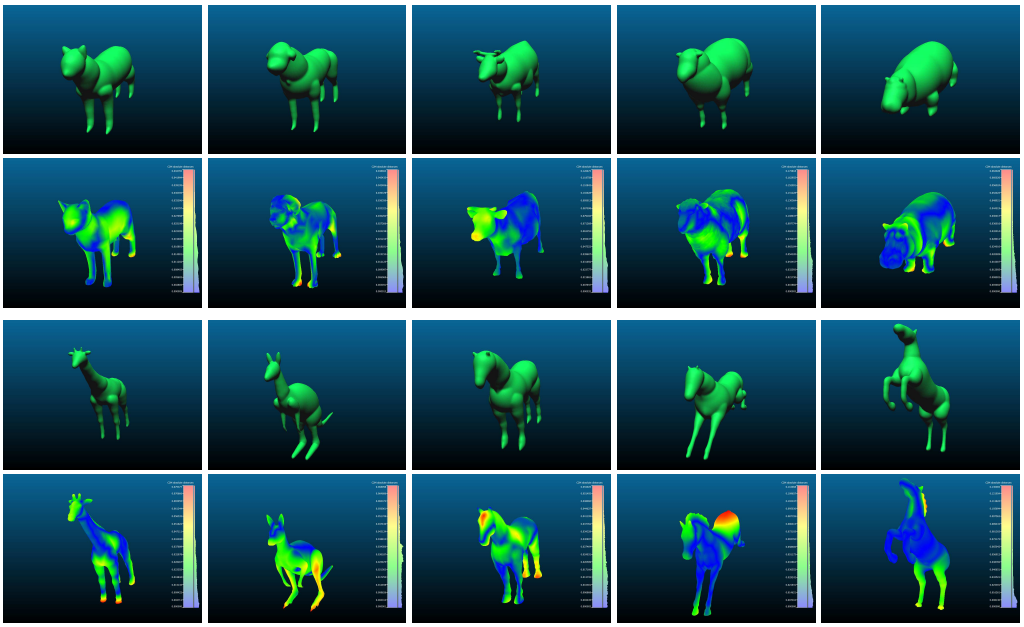

Figure 8: Comparison between animals modeled with our approach (odd rows) from images of models downloaded from the web (even rows) which were used as ground truth. The images of the bottom group show the distribution of the normalized Hausdorff distance on the ground truth model. (Best seen in color and on-screen)

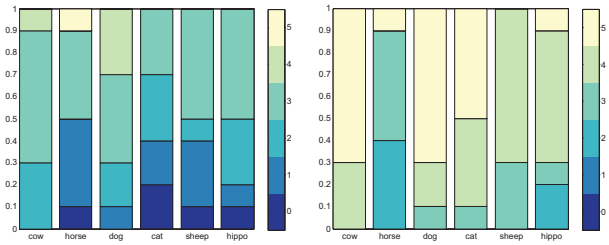

Figure 9: Vote distribution for the models produced with our approach (**left**) and models taken from the web (**right**).

Cow	Horse	Dog	Cat	Sheep	Hippo
70%	50%	70%	30%	50%	50%
100%	60%	100%	100%	100%	80%

Table 3: Per-class percentage of votes above 3 (good) given to the models reconstructed by our method (first row), and the models downloaded from the web (second row).

downloaded from the web received higher votes in average, with a difference of 1.9 scale units with respect to the average vote that our models received. This is understandable considering that our models correspond to more abstract class models, lacking particular details like eyes, nose and tail. Nevertheless, the percentage of the participants, which gave a vote above 3 (good) for the quality of our models (Table 3), indicates that the models are of satisfying quality.

6. Conclusions and future work

We propose a method for computing 3D models of articulated objects, by decomposing them into components. Realistic models of the object components are built by merging together 3D models obtained from different aspects, considering a kind of aspect graph [15], which indicates the essential aspects. Aspects are extracted from images downloaded from the web. The entire object is obtained by reassembling the components using two or more images of the object in a reference pose. Our experiments suggest that our method is able to provide realistic models of the objects, both in terms of a perceptual analysis, and by a quantitative analysis of their similarity with respect to human created 3D models.

An important extension of this work is the possibility to model the object in different configurations by using a single image. This can be made possible by learning spatial relations between the components (joints, joint range etc.) and possibly also a distribution of the object poses, which would allow to compute realistic models even when some of the components are occluded. Finally, another useful extension would be the automatic selection of the most representative aspects for each component from a set of images.

Acknowledgments

Supported by the EU FP7 TRADR (609763) and the EU H2020 SecondHands (643950) projects. The authors thank the anonymous reviewers for their constructive comments.

References

[1] N. Aspert, D. Santa Cruz, and T. Ebrahimi. Mesh: measuring errors between surfaces using the hausdorff distance. In *ICME*, pages 705–708, 2002. 6

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417. Springer, 2006. 4

[3] I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychol. Rev.*, 94(2):115–147, 1987. 1

[4] T. Binford. Visual perception by computer. In *MSC*, volume 261, page 262, 1971. 1

[5] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *TVCG*, 14(1):213–230, 2008. 1, 3

[6] C. Budd, P. Huang, M. Klaudiny, and A. Hilton. Global non-rigid alignment of surface sequences. *IJCV*, 102(1-3):256–270, 2013. 2

[7] F. Calakli and G. Taubin. Ssd: Smooth signed distance surface reconstruction. *Pacific Graphics*, 2011. 2

[8] J. Carreira, A. Kar, S. Tulsiani, and J. Malik. Virtual view networks for object reconstruction. In *CVPR*, pages 2937–2946, 2015. 2

[9] T. Cashman and A. Fitzgibbon. What shape are dolphins? Building 3D morphable models from 2D images. *TPAMI*, 35(1):232–44, 2013. 2

[10] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. In *ACM SIGGRAPH*, volume 25, pages 257–266. ACM, 1991. 3

[11] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or. 3-sweep: Extracting editable objects from a single photo. *ACM TOG*, 32(6):195, 2013. 1

[12] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, 2014. 3

[13] R. Cipolla. The Visual Motion of Curves and Surfaces. *Phil. Trans. Royal Soc. London A*, 356:1103–1121, 1998. 5

[14] R. Cipolla and P. Giblin. *Visual Motion of Curves and Surfaces*. Cambridge, 2000. 2

[15] S. Dickinson, A. Pentland, and A. Rosenfeld. Qualitative 3-d shape reconstruction using distributed aspect graph matching. In *ICCV*, pages 257–262, 1990. 1, 2, 8

[16] I. L. Dryden and K. Mardia. *Statistical shape analysis*, volume 4. John Willey & Sons, 1998. 6

[17] J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13(3):321–330, 1984. 2

[18] J. Koenderink. *Solid Shape*. MIT, 1990. 2, 5

[19] I. Kokkinos and A. Yuille. Hop: Hierarchical object parsing. In *CVPR*, pages 802–809, 2009. 3

[20] I. Kokkinos and A. Yuille. Inference and learning with hierarchical shape models. *IJCV*, 93(2):201–225, 2011. 3

[21] Z. Levi and C. Gotsman. ArtiSketch: a system for articulated sketch modeling. *Comput. Graph. Forum*, 32(2):235–244, 2013. 1

[22] J. Liang, F. Park, and H. Zhao. Robust and efficient implicit surface reconstruction for point clouds based on convexified image segmentation. *JSC*, 54(2-3), 2013. 2, 4

[23] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM SIGGRAPH*, volume 21, pages 163–169, 1987. 4

[24] P. Markelj, D. Tomaževič, B. Likar, and F. Pernuš. A review of 3d/2d registration methods for image-guided interventions. *Medical Image Analysis*, 16(3):642–661, 2012. 5

[25] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. FiberMesh. *ACM TOG*, 26(3):41, 2007. 1

[26] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *TPAMI*, 31(5):783–794, 2009. 6

[27] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003. 4

[28] M. Oswald, T. Eno, and D. Cremers. Fast and globally optimal single view reconstruction of curved objects. In *CVPR*, pages 534–541, 2012. 2

[29] M. Oswald, E. Töppe, C. Nieuwenhuis, and D. Cremers. A review of geometry recovery from a single image focusing on curved object reconstruction. In *Innovations for shape analysis, models and algorithms*, pages 343–378. Springer, 2013. 2

[30] A. Pentland. Perceptual organization and the representation of natural form. *Artif. Intell.*, 28(3):293–331, 1986. 1

[31] S. Plantinga and G. Vegter. Computing contour generators of evolving implicit surfaces. *ACM TOG*, 25(4):1243–1280, 2006. 2

[32] M. Prasad, A. Fitzgibbon, A. Zisserman, and L. Van Gool. Finding nemo: deformable object class modelling using curve matching. In *CVPR*, pages 1720–1727, 2010. 1, 2

[33] M. Prasad, A. Zisserman, and A. Fitzgibbon. Single view reconstruction of curved surfaces. In *CVPR*, pages 1345–1354, 2006. 2

[34] I. Quesada and I. E. Grossmann. An lp/nlp based branch and bound algorithm for convex minlp optimization problems. *Computers & chemical engineering*, 16(10):937–947, 1992. 6

[35] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *ACM SIGGRAPH*, pages 205–214, 1987. 3

[36] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: recovering 3d shape and nonrigid motion. *Artif. Intell.*, 36(1):91–123, 1988. 2

[37] E. Töppe, C. Nieuwenhuis, and D. Cremers. Relative volume constraints for single view 3D reconstruction. In *CVPR*, pages 177–184, 2013. 1, 2

[38] S. Vicente and L. Agapito. Balloon shapes: reconstructing and deforming objects with volume from images. In *3DV*, pages 223–230, 2013. 1, 2

[39] S. Vicente, J. Carreira, L. Agapito, and J. Batista. Reconstructing pascal voc. In *CVPR*, pages 41–48, 2014. 2