Capturing Skeleton-based Animation Data from a Video

Liang-Yu Shih, Bing-Yu Chen
National Taiwan University
E-mail: xdd@cmlab.csie.ntu.edu.tw, robin@ntu.edu.tw

ABSTRACT

This paper presents a semi-automatic method to capture animation data from a single camera video. The video is first segmented and analyzed, and a 3D character model with skeleton rigged is input as a reference model. Then, the user modifies the reference model to fit the subject's contour in the starting frame, and specifies the body and limbs contours of the subject character. Our system then reconstructs the motion by estimating the reference model's pose automatically in each frame forward. Finally, the user's intervention helps to refine the result. This paper shows a better motion reconstruction result than totally automatic methods that are widely used in video-based motion capture in computer vision.

Keywords: video-based motion capture, user-aid.

1. INTRODUCTION

With the booming popularity of 3D animation movie and video games, character animation becomes more important than ever. The topic about creating a living motion of human or animal is widely discussed and remains as one of the most difficult problems in computer graphics. Motion capture is a good solution for making fantastic motions. Traditional motion capture methods require cooperation from the subject. The subject has to wear markers, move in a reduced space, and sometimes even need to stay on a treadmill, and then motions are captured through detections of the markers. However, it is impossible to ask for animals' cooperation like these. Therefore, some markerless methods are developed, named as video-based motion capture with automatic reconstruction of motion, but as addressed in computer vision, automatic reconstruction of subject motion from single camera video still has huge difficulties.

This paper develops a video-based system that gets animal motions from an unrestricted monocular video with user's aid. In order to break the limitation of pure automatic method, a reference 3D model and user's intervention are added as input in the system. The concept is to estimate the reference 3D model's pose in each frame of a video, according to the difference between reference model contour and animal's one. Then error and ambiguity correction is relied on user's intervention. Before that, our system uses an automatic method from computer vision to estimate camera parameters and relationship between camera and scene, and integrates some interactive techniques from computer graphics in order to provide users a friendly and efficient interface.

2. RELATED WORK

Video-based motion capture is very popular and difficult topic in computer vision. By relying on prior knowledge about human motion, Howe *et al.* [6] reconstruct the 3D motion of human subject and resolve motion ambiguity. On the other hand, Sidenbladh *et al.* [10] use a probabilistic method tracking 3D articulated human figures. Both of the approaches are widely adopted for automatic character motion reconstructions from a single-camera video. However, Gleicher and Ferrier [5] show that these current computer vision techniques for the automatic video processing fail to provide reliable 3D information, such as stable joint angles over time. They conclude that using these methods is current not feasible.

Recently, capture motion from multi-view video [3] [11] perform good results, even for reconstruction of character's mesh details like clothes. However, it requires complicated equipments and environment for making the multi-view video, that is cost, and time consuming. As a result, this paper aims at capturing animal motions from a free move single-camera video, and the assumptions are more similar to those of researches mentioned earlier.

The first attempts to reconstruct animal motion from video is Wilhelms and Van Gelder's work [13]. They extract the horse motion from a video sequence by using deformable contour - active snake. Features

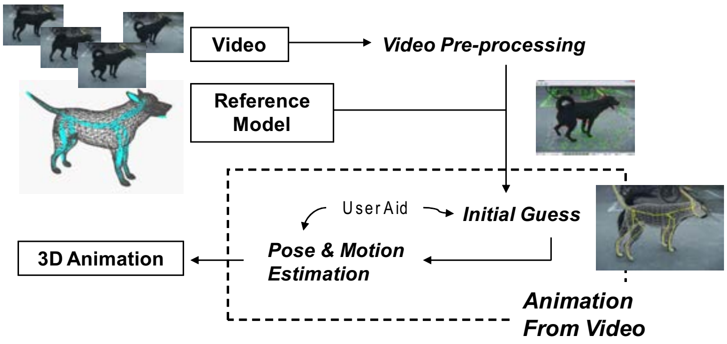


Figure 1: Overview of our system.

on the snake contour anchor specified bones, when the features change in frames, they will pull the bones into the right positions. However, active contour methods are very sensitive to noise and parameter tunings are difficult. Therefore, active snake adopts user interaction, such as adjusting contours that are failing to track. The method heavily relies on users's help, especially when occlusions occur.

Examples-based approaches have recently been recognized as good alternative to traditional shape modeling and animation methods. The basic idea is to interpolate between a given set of 3D poses or motion examples. Favreau *et al.* [4] apply Principal Component Analysis (PCA) to automatic select key-images from live video sequence. Artists provide 3D pose examples of key-images, then interpolate examples with Radial Basis Function (RBF). They generate high quality cyclic motions of animals from video.

3. OVERVIEW

This paper aims at capturing animal motions through imitations of the subject's motion by the reference model in each frame of the input video sequence. Figure 1 shows the process of our system, starting with a sequence of single-camera video sequence, and a reference 3D model as the input.

Section 4 presents our methods about getting information from the video sequence: it explains the detail operations that we get the target animal's contour and a way for getting the relationship between camera and scene. Section 5 describes some required properties, and controlling manners of the input reference model. Section 6 introduces the motion reconstruction method, which is the most important part in this paper. By using the animal contours and reference model's information, our system automatically estimated the pose of subject animal in each frame. When estimation errors or ambiguity occur, corrections are made with users' aid, and correction will be propagated to other frames as well.

4. VIDEO PRE-PROCESSING

4.1. Segmentation

In order to get subject character's motion from video, users need to cut out the contour of it. Several tools are provided for cutting out the contour [2] [7] [12]. Our system uses an intuitive method - GrabCut [9] to solve this problem. Since the target animal's contour only lightly change between two consecutive frames, we treat the GrabCut's result of previous frame as the initial guess of segmentation. With the modification, users can cut out the contours easily and efficiently from video.

4.2. Camera Calibration

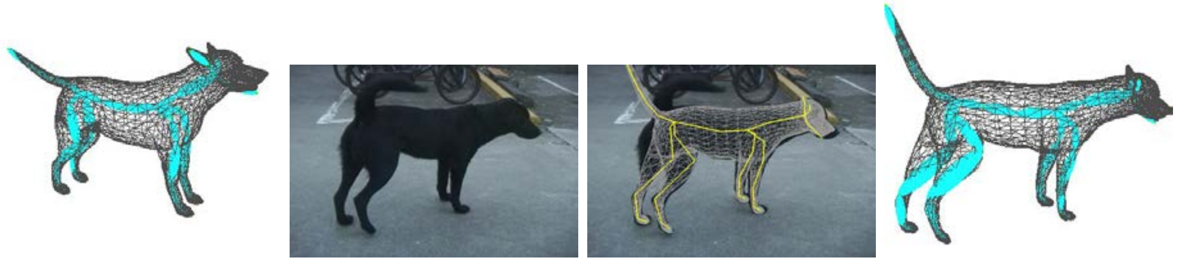
Structure and motion analysis [8] is a computer vision technique that automatically reconstructs a sparse set of 3D scene points from a video sequence. It also decides the camera parameters which describe the relationship between the camera and the scene. Therefore, the system apples the technique by using the Voodoo Camera Tracker[1] to get the camera parameters and estimates the projection matrix in each frame.

5. THE REFERENCE MODEL

5.1. Mesh Information

Users choose a 3D model as reference which helps to get the target animal's motion in the video sequence.

[1] http://www.digilab.uni-hannover.de/



(a) (b) (c) (d)

Figure 2: Initial fitting. *(a) :* The origin reference model with skeleton. *(b) :* Reference frame used in initial fitting. *(c) :* Modify scene and reference model to fit the target animal in reference frame. *(d) :* The reference model after initial fitting process.

Besides 3D mesh information, the reference model also provides other information as below files :

5.1.1. ASF/WGT File

ASF File - This file defines a skeleton in terms of the model's shape, hierarchy, and properties of its joints and bones. The format used by the Acclaim Motion Capture System.

WGT File - This file defines how the skeleton affects the skin. Each vertex is influenced by several joints, and the total influence weights are 1.

5.2. Skeleton Constraint

There are three constraints for the model skeleton although they are not recorded in the ASF File.

1. Limited Rotation - Some bones cannot be rotated such as pelvis and sacral.
2. Simultaneous Rotation - Rotation of one bone will affects other related bones, e.g. ears always rotate with head.
3. Symmetric Bones - Animals have symmetric component, such as right foreleg is symmetric with the left. As a result, scaling one bone will have the same effect on the other symmetric bone.

6. ANIMATION FROM VIDEO

Our approach automatically estimates the reference model's pose to fit the subject contour in each frame. Wilhelms and Van Gelder's work [13] mentioned that when motion is not parallel to the image plane, extracting three-dimensional positions at joints is an extremely under-constrained problem. In this paper, we allow the subject to be at an angle to the image plane, instead of only limited to a parallel plane. However, motion have to move along with the subject plane, the $x - y$ plane of the subject's coordinate. The motion reconstruction process is as follow:

1. As a preliminary step, the user adjusts the reference model to fit the subject in the video. This is done interactively by choosing a frame that can best illustrate different parts of the subject.
2. The system estimates the pose of the reference model automatically in each frame forward.
3. The user can tune the pose estimation in an arbitrary frame, and then the system will propagate the correction forward or backward.

6.1. Initial Fitting

By rotating and scaling the bones under the constraints mentioned in Section 5.2, the user modifies the reference model for initial fitting. Because the target animals come with different shapes and sizes, in order to reduce the differences between the target animal and the reference model, the uses can adjust the reference model by scaling components proportionately. Figure 2 (d) indicates the dog model's hind legs increase to be more similar to the target animal shown in Figure 2 (b).

A hint for the fitting is to make the contour of model lightly smaller than the subject's contour in the frame. Figure 2 (c) shows the initial fitting of the reference model in Figure 2 (a) to the reference frame shown in Figure 2 (b).

6.1.1. Scene Estimation

Since our input video is free-move single-camera video, we cannot simply put the reference model onto the image plane by using orthogonal projection. Instead, we must reconstruct the 3D virtual scene to simulate the real scene and put the reference model into it. Scene is reconstructed by the result of structure and motion analysis mentioned in Section 4.2. The user is asked to modify the subject's coordinate to align the ground and subject's orientation of real scene. Figure 3 (b) shows the result of the initial fitting process. The grey plane is subject's $x-z$ plane aligned with the ground in virtual scene, and the camera is set at the origin in the first frame.

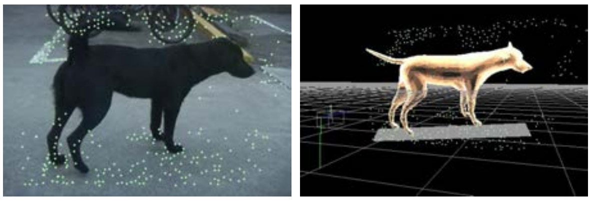


Figure 3: Scene estimation. *(a) :* 3D scene features project onto the image plane. *(b) :* Reference model in reconstructed virtual scene.

6.2. Pose Estimation

General animals have 7 components - head, torso, left and right forelegs, left and right hind-legs, and tail. The user is asked to specify the bones in each component of the reference model (Figure 4 (a)). We define our bone as a joint pair $B = (j_1, j_2)$, which j_1 is j_2's parent.

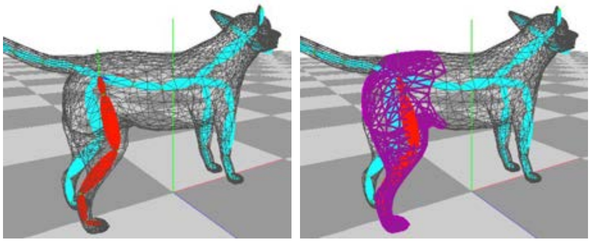


Figure 4: Component assignment. *(a) :* The user specifies the red bones for right hind leg. *(b) :* The purple triangles are influenced by the selected bones.

After component specification, we need to find the edges $e = (v_1, v_2)$ which mainly influenced by each component's bones. With weighting information in the wgt file, we can generate a map of edges and bones. A vertex v is mainly influenced by the joint which has the maximum weighting, so that we can find a joint pair (j_1, j_2) which mainly influence an edge. Since an edge often mainly influence by a joint ($j_1 = j_2$), we find the second large weight with a threshold ($w_i > 0.1$) of joint and put the higher hierarchy in j_1. Then, we can find the involved edges of each component from the map.

With the projection matrix obtained from the VooDoo Camera Tracker in each frame and the user's specification, we can project the involved edges of different components onto the image plane, then identify the contour points of each component. Figure 5 (a) shows the right hind-leg's contour (blue strokes) on the image plane.

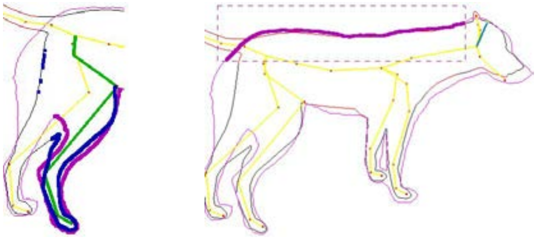


Figure 5: *(a) :* A partial projected frame, the model contour overlaps the subject contour. *(b) :* A square select mechanism.

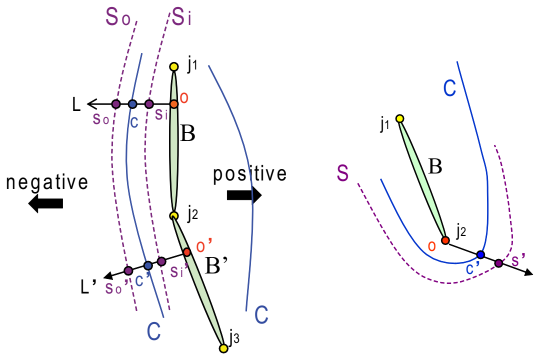
Since we have the subject contour in each frame, we still need to specify each component's contour as the reference in estimated process. Agrawala *et al.* [1] present an interactive contour tracking method, but it cannot work when occlusion occurs. Since we already have the subject's contour totally, the user is asked to simply assign the subject contour of different components in each frame via a square select mechanism (Figure 5 (b)).

The pose estimation contains two parts - limbs rotation and body translation. The system automatically modifies the reference model by referring the information of the previous frame to fit the subject's contour.

6.2.1. Component Rotation Estimation

Each bone B in each component has two error data $E = \{e, e_d\}$. To record the difference between the subject contour of this component and the model contour which B involved, bone B forms a line on the image plane, and we denote the error on the right (positive) side of the line as E_p and the left (negative) side as E_n.

Figure 6 (a) shows an example about how we calculate the error data E. The model contour C is influ-



(a) (b)

Figure 6: Calculation of error data E.

enced by the bone B. A point $c \in C$ can project onto $\overline{j_1 j_2}$ at point o and form a line L which is pedicular to B, and L intersects the subject contour S at point s (Figure 6 (a) shows two different subject contours). E_p is calculated by all c lie on the right side of B with Equation (1) and E_n is calculated in the same way with all c lie on the left side of B.

$$
\begin{aligned}
e &= \tfrac{1}{n} \sum_{1}^{n} \left(\|\overline{so}\| - \|\overline{co}\| \right) \\
e_d &= \tfrac{1}{n} \sum_{1}^{n} \left| \left(\|\overline{so}\| - \|\overline{co}\| \right) \right|
\end{aligned}
\qquad (1)
$$

The model contour points of B sometimes cannot project on line $\overline{j_1 j_2}$, like c' in Figure 1. Hence, we try to project them onto B's child bone B' first (Figure 6 (a)). If they still cannot project, then we choose j_1 or j_2 as o determined by smaller distance to c' (Figure 6 (b)).

Due to occlusion or user's input, there are few or even no intersect point at the subject contour of this component. Hence, we treat these E as invalid. Algorithm 1 shows our method to estimate the component rotation, the system automatically rotates the bones by steps in hierarchy order until no rotation occurs. The goal of our algorithm is to minimize $D(E_p)$ and $D(E_n)$ calculated by Equation (2) of each bone. Our system sets the weighting variable $w = 2$. After estimation, the difference of the contours between the model and the subject is similar to the previous frame.

$$
\begin{aligned}
& D(E) = (d_1, d_2) = \\
& \quad \|E_{frame} - E_{frame-1}\| = w^k \left(|e - e'|, |e_d - e'_d| \right) \\
& \quad \text{, where } (e, e_d) \in E_{frame},\ (e', e'_d) \in E_{frame-1} \\
& \quad \text{, and } k = 1 \text{ if } ee' < 0 \text{ else } k = 0.
\end{aligned}
\qquad (2)
$$

Algorithm 1 Component Rotate Algorithm

```
Require: Bones sort by hierarchy from high to low
repeat
  for $i = 1$ to $n$ do
    repeat
      choose valid $E$ between $E_p$ and $E_n$ as reduce goal
      determine rotate direction to reduce $D(E)$
      rotate $B_i$ with one step parallel to subject's $x - y$ plane
      recompute error data $E_p$ and $E_n$
    until reduction fails or $total\_rotate > max\_rotate$
  end for
until no rotation occurs in all bones
```

To rotate a bone, the system chooses the valid error data between E_p and E_n which has smaller distance $\|e_d - |e|\|$ of E first. Then, the system determines the rotate direction by below rule, and the direction will be opposite when $E = E_n$.

```
if $E = E_p$ then
  if $e > e'$ then {$e'$ is error in previous frame}
    direction = positive
  else
    direction = negative
  end if
end if
```

The bone is rotated by one step at a time, and then the reduction failure and total rotation steps are checked. The rotation will be stopped if the reduction failure occurs or total rotation steps are larger than a threshold. The reduction fails when $D(E)$ is larger than the previous step or both E_p and E_n are invalid.

6.2.2. Translation Estimation

Only the root joint of the model is translated. We use the region of translation the users specified to estimate the x direction, and estimate the y direction by minimizing $T(E)$ calculated by Equation (3).

$$
\begin{aligned}
T(E) &= \sum_{B} \left(\|D(E_p)\| + \|D(E_n)\| \right), B \in Torso \\
\|D(E)\| &= d_1 + d_2
\end{aligned}
\qquad (3)
$$

Figure 7 (a) shows the triangles specified by the bones to estimate the translation of x, and Figure 7 (b) shows the projected region of these triangles. The system translates x to minimize the non-overlay region as shown as the green region of Figure 7 (c)

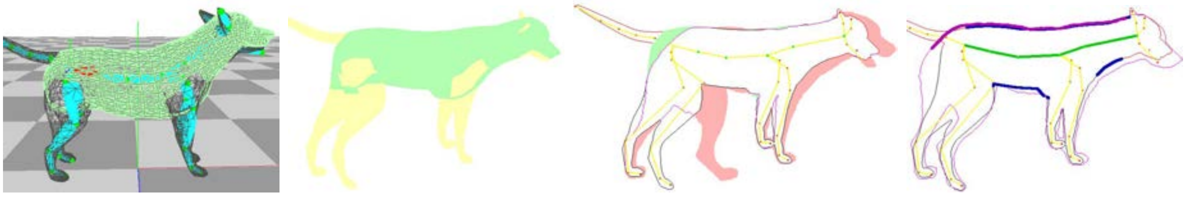


Figure 7: Translate estimation. *(a) :* The green triangles are specified for the translated estimation. *(b) :* The projection of triangles which are specified for translated estimation. *(c) :* Overlap between the reference model contour and the subject image (pink region). *(d) :* The model contour and subject contour of the torso component. Notice that there is one side of the subject contour.

between the subject image and the projected triangles. The subject's tail often makes occlusion with the torso, so we eliminate the tail in segmentation process mentioned in Section 4.1 in order to make a better estimation. In y direction, similar to estimate x, the system translates y to minimize $T(E)$ which is the difference of the torso between the current frame and the previous frame.

In order to prevent the errors occur by large change of translation or rotation, we modify the reference model by a step at a time. Algorithm 2 shows our method of the pose estimation.

Algorithm 2 Pose Estimation Algorithm

```
repeat
  translate $x$ with one *step* to reduce non-overlay region of translation
  repeat
    translate $y$ with one *step* and rotate torso to reduce $T(E)$
    rotate all components excluding torso
  until $T(E)$ cannot be reduced
until non-overlay region of translation cannot be reduced
```

6.3. Refinement

The user can specify the amount of frames which the system makes estimation forward. There may be some incorrect estimation, and the incorrect result will propagate to the next frame by using our method. With this character of our method, the user can refine the automatic estimation result and propagate the correction backward or forward. Figure 8 shows automatic estimation of the reference model poses in 10 frames in top row, and the bottom row shows the correction propagation of dog's right leg from user's modification in frame 9.

7. RESULTS

Our system is implemented in C++ with OpenGL. The video source is captured by using SONY DCR TRV 900 video camera with frame rate 24 frames/s and interlace mode. Figure 9 shows our result of a dog's sitting motion. We ignore the tail' motion because it moves too frequently to make estimation.

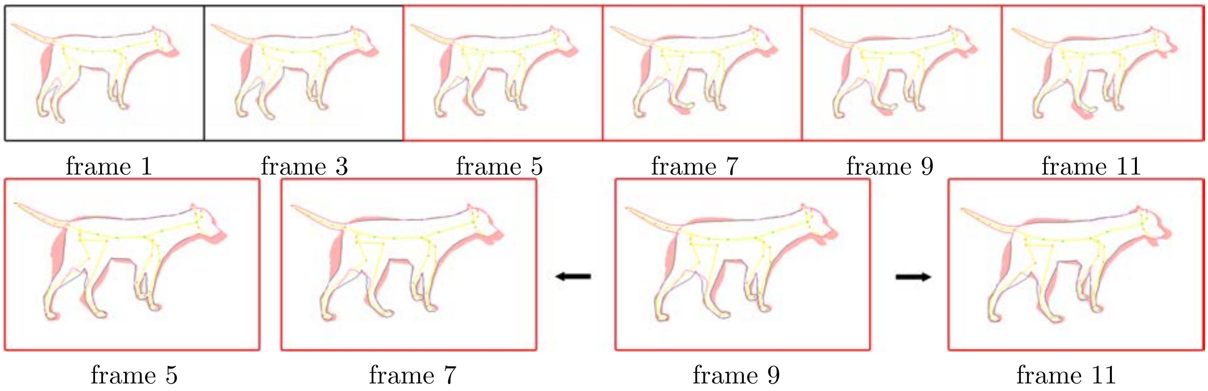
8. CONCLUSION

The main advantages of our method are as follow:

- **UI**: Our system provides an intuitive and friendly user interface for users to make specifications and modifications. By using the reference model with rigged skeleton, the user can easily make the adjustment and instantly preview the change of model contour in the initial fitting and refinement processes.
- **Animation**: With our system, it is easy to get a lively unrestricted in-plane motion even for the users who are not professional artists or do not have enough knowledge of the subject character.

There are two limitations of our method. Due to lack of depth information, our system cannot make estimation of out-subject-plane motion. Although we remain the manual modification of out-subject-plane motion for users, it is still difficult to make accurate estimation. Another limitation is that the unavoidable differences between the reference model and the subject character make our system not robust enough for all scenarios.

For the future work, we suggest to add prior knowledge and example animation data of subject characters in order to reduce the ambiguities and user's interventions.



frame 1 frame 3 frame 5 frame 7 frame 9 frame 11

frame 5 frame 7 frame 9 frame 11

Figure 8: Refinement. *Top :* Estimate the model's pose automatically in numbers of frame forward. *Bottom :* Corrections propagate to backward and forward frames.

ACKNOWLEDGEMENT

This work was partially supported by the National Science Council of Taiwan under NSC95-2221-E-002-273.

REFERENCES

[1] A. Agarwala, A. Hertzmann, D. H. Salesin, and S. M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics*, 23(3):584–591, 2004. (SIGGRAPH 2004 Conference Proceedings).

[2] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proceedings of 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2001.

[3] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Transactions on Graphics*, 27(3), 2008. (SIGGRAPH 2008 Conference Proceedings).

[4] L. Favreau, L. Reveret, C. Depraz, and M.-P. Cani. Animal gaits from video. In *Proceedings of 2004 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pages 277–286, 2004.

[5] M. Gleicher and N. Ferrier. Evaluating video-based motion capture. In *Proceedings of 2002 Computer Animation*, pages 75–80, 2002.

[6] N. R. Howe, M. E. Leventon, and W. T. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *Proceedings of 1999 Neural Information Processing Systems*, pages 820–826, 1999.

[7] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Transactions on Graphics*, 23(3):303–308, 2004. (SIGGRAPH 2004 Conference Proceedings).

[8] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

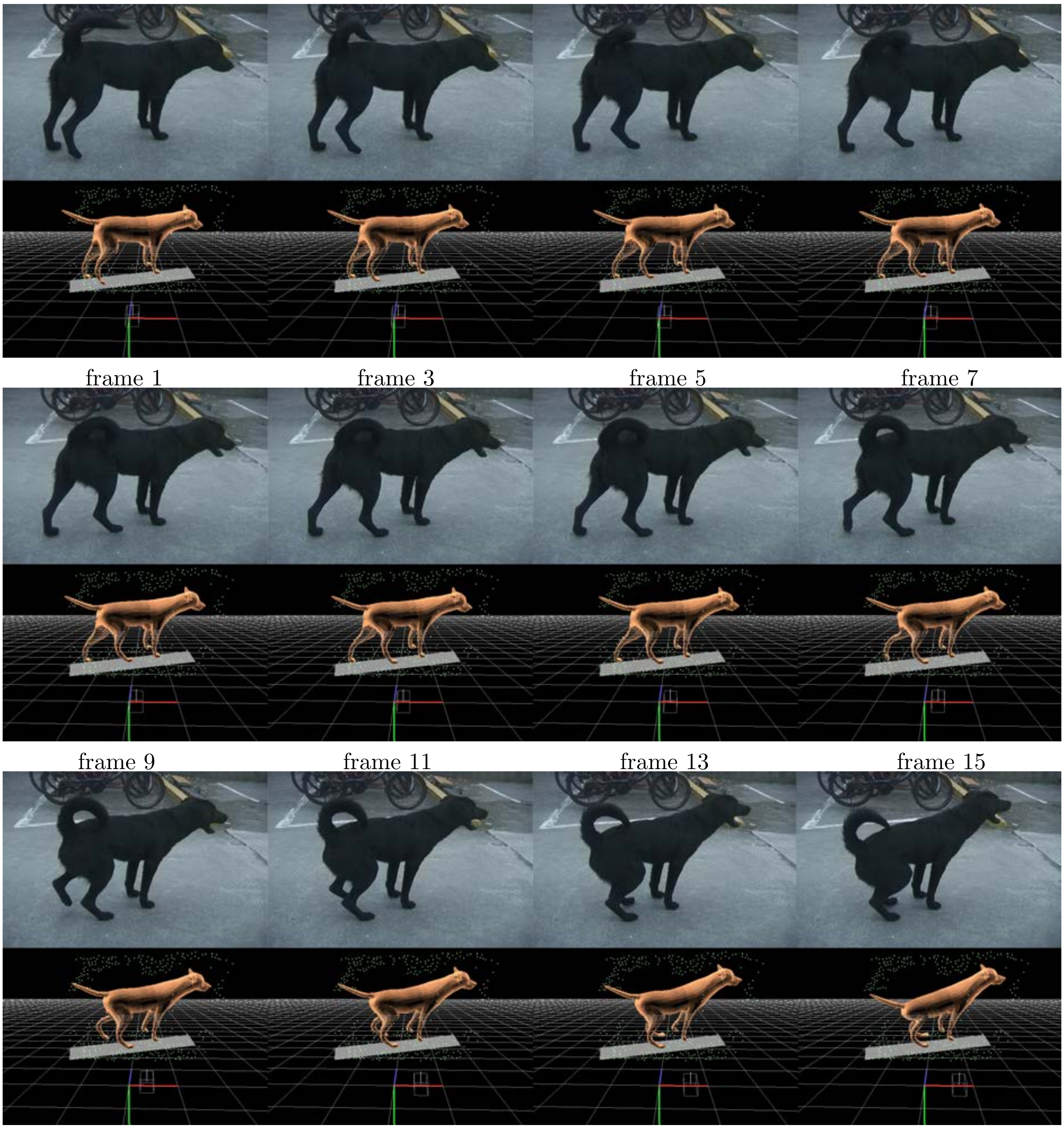
[9] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004. (SIGGRAPH 2004 Conference Proceedings).

[10] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Proceedings of 2000 European Conference on Computer Vision*, volume 2, pages 702–718, 2000.

[11] D. Vlasic, I. Baran, W. Matusik, and J. Popovič. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics*, 27(3), 2008. (SIGGRAPH 2008 Conference Proceedings).

[12] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: an interactive tool for realtime high quality matting. *ACM Transactions on Graphics*, 26(3):9, 2007. (SIGGRAPH 2007 Conference Proceedings).

[13] J. Wilhelms and A. V. Gelder. Combining vision and computer graphics for video motion capture. *The Visual Computer*, 19(6):360–376, 2003.



frame 1 frame 3 frame 5 frame 7

frame 9 frame 11 frame 13 frame 15

frame 17 frame 19 frame 21 frame 23

Figure 9: Result.