Video-based Motion Capturing for Skeleton-based 3D Models

Liang-Yu Shih, Bing-Yu Chen, and Ja-Ling Wu

National Taiwan University
xdd@cmlab.csie.ntu.edu.tw; robin@ntu.edu.tw; wjl@csie.ntu.edu.tw

Abstract. In this paper, a semi-automatic method to capture motion data from a single-camera video is proposed. The input video is first segmented and analyzed, and a 3D character model with skeleton rigged is used as a reference model. Then, the reference model is modified to fit the subject's contour in the starting frame, and the body's and limbs' contours of the subject are also specified by the user. Our system then extracts the motion from the video by estimating the reference model's poses automatically in each video frame forwardly. Finally, the user can help to refine the result through a friendly user interface.

Key words: video-based motion capture, user-aid, reference model

1 Introduction

With the booming popularity of 3D animations and video games, how to create or obtain the character motion becomes more and more important than before. Motion capture is a good solution for obtaining fantastic motions. Traditional motion capture methods require cooperation from the capturing subject, such as wearing markers, moving in a reduced space, and sometimes even needing to stay on a treadmill, and then the subject's motions are captured through the markers. However, it is impossible to ask for animals' cooperation like these. Therefore, some markerless methods are proposed, named as video-based motion capture, but as addressed in computer vision, automatic reconstruction of subject motion from a single-camera video is still very difficult.

In this paper, we develop a video-based system that extracts 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 used in the system. The concept is to estimate the reference model's pose in each video frame according to the difference between the reference model's and animal's contours, and the error and ambiguity correction is relied on the user's intervention. Beside these, our system uses an automatic method to estimate camera parameters and relationship between the camera and scene, and integrates some interactive techniques in order to provide the user a friendly and efficient interface. Fig. 1 shows the overview of our system, which uses a single-camera video and a reference 3D model as the input and extracts the motion for the model from the video.

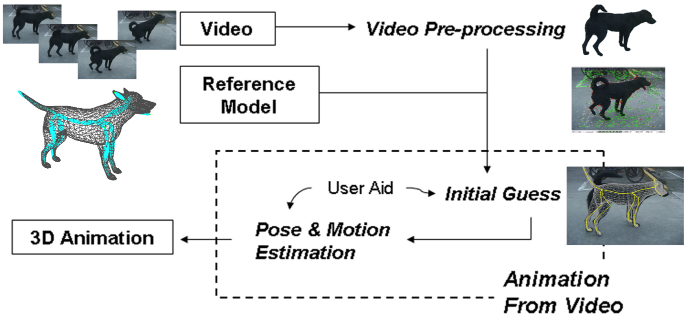


Fig. 1. Overview of our system.

2 Related Work

Video-based motion capture is a popular and difficult topic in computer vision. By relying on prior knowledge about human motion, Howe *et al.* [1] reconstructed the motion of human and resolve the ambiguity. On the other hand, Sidenbladh *et al.* [2] used a probabilistic method tracking 3D articulated human figures. Both of them are widely adopted for automatic character motion reconstruction from a single-camera video. However, Gleicher and Ferrier [3] showed that these 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, capturing motion from multi-view videos [4] [5] performs good results, even for reconstructing the character's mesh details like clothes. However, it requires complicated equipments and environment for making the multi-view videos, which is expensive and time consuming.

To reconstruct animal motion from a video, Wilhelms and Van Gelder [6] presented a method to extract the horse motion from a video by using deformable contour - active snake. The features on the snake contour anchor are used to specify the bones. When the features change in frames, the bones are pulled into the right positions. However, since active contour is very sensitive to noise and parameter tuning is also difficult, it usually needs user's interaction to adjust the contours that are failing to track. Examples-based approaches have recently been recognized as good alternatives 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.* [7] apply Principal Component Analysis (PCA) to automatically select key-images from a live video. Then, the artist is asked to provide 3D pose examples of the key-images and the system interpolates the examples with Radial Basis Function (RBF). Finally, they generate high quality cyclic motions of animals from the video.

3 Video Pre-processing

3.1 Segmentation

In order to obtain character motion from a video, it is needed to cut out the contour of the character in the video. In our system, we provide an intuitive method - GrabCut [8] to help the user to do this. Since the contour of the target animal only lightly changes between two consecutive frames, we use the GrabCut's result of the previous frame as the initial guess of current frame's segmentation. With this modification, the user can cut out the contours easily and efficiently from the video.

3.2 Camera Calibration

Pollefeys *et al.* [9] provided a structure and motion analysis method to automatically reconstruct a sparse set of 3D scene points from a video. It also decides the camera parameters which describe the relationship between the camera and scene. In our system, we use this method by using Voodoo Camera Tracker[1] to obtain the camera parameters and estimates the projection matrix of each frame.

4 The Reference Model

To extract the animal motion from the video, a 3D model is used as the reference. Besides the mesh information, the reference model also provides the following information:

ASF File - This file defines a skeleton in terms of the model's shape, hierarchy, and properties of its joints and bones. The file format is 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.

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

1. Limited Rotation - Some bones cannot be rotated such as pelvis and sacral.
2. Symmetric Bones - Animals have symmetric components, 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.

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

5 Motion Extraction from a Video

Our approach automatically estimates the reference model's pose to fit the subject's contour in each frame. Wilhelms and Van Gelder [6] mentioned that when motion is not parallel to the image plane, extracting 3D positions at the 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, the subject's motion is required to move along the subject plane, i.e. the $x - y$ plane of the subject's coordinate. The motion reconstruction process is as follow:

1. As a preliminary step, the user is asked to adjust the reference model to fit the subject in the video, which can be done interactively by choosing a best frame that can illustrate several parts of the subject.
2. The system estimates the pose of the reference model automatically in each frame forwardly.
3. The user can tune the estimated poses in an arbitrary frame, and the system will propagate the correction forwardly and backwardly.

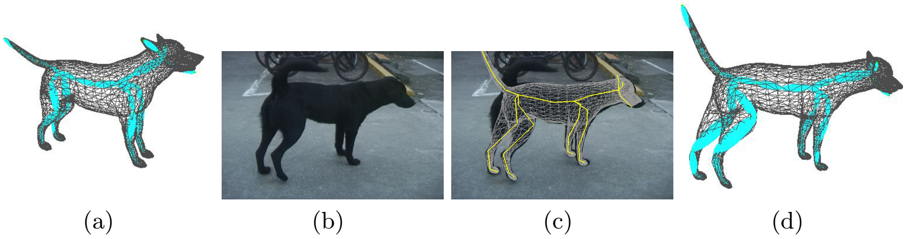


Fig. 2. Initial fitting. (a) The original reference model with skeleton. (b) The reference frame used for initial fitting. (c) The reference model is modified to fit the target animal in the reference frame (b). (d) The reference model after the initial fitting process.

5.1 Initial Fitting

By rotating and scaling the bones under the constraints mentioned in Sec. 4, the reference model is modified for the initial fitting. Because the target animals have different shapes and sizes, in order to reduce the differences between the target animal and the reference model, the user can adjust the reference model by scaling the components proportionately. Fig. 2 (d) indicates that the dog model's hind legs are modified to fit the target animal shown in Fig. 2 (b). A hint for the fitting is to make the model's contour lightly smaller than the subject's contour in the frame. Fig. 2 (c) shows the initial fitting of the reference model in Fig. 2 (a) to the reference frame shown in Fig. 2 (b).

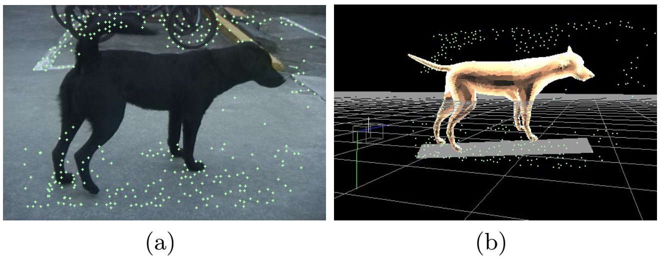


Fig. 3. Scene estimation. (a) 3D scene features project onto the image plane. (b) The reference model in the reconstructed virtual scene.

Scene Estimation Since our input video is a 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. The virtual scene is reconstructed by the result mentioned in Sec. 3.2. The user is asked to modify the subject's coordinate to align the ground and subject's orientation in the real scene. Fig. 3 (b) shows the result of the scene estimation process. The grey plane is subject's $x - z$ plane aligned with the ground in the virtual scene, and the camera is set at the origin in the first frame.

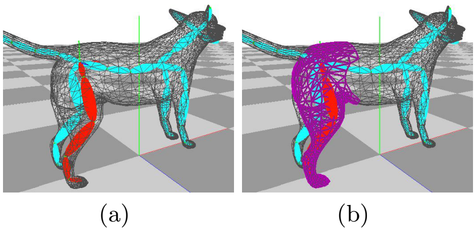


Fig. 4. Component assignment. (a) The user specifies the red bones for right hind leg. (b) The purple triangles are influenced by the selected bones.

5.2 Pose Estimation

General animals have 7 components - head, torso, left and right forelegs, left and right hind-legs, and tail. Hence, the user is asked to specify the bones in each component of the reference model (Fig. 4 (a)). We define our bone as a joint pair $B = (j_1, j_2)$, which j_1 is j_2's parent. After component specification, we need to find the edges $e = (v_1, v_2)$ which mainly influenced by each component's bones. With the weighting information in the WGT File described in Sec. 4, 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 influenced by a joint ($j_1 = j_2$), we find the second large weight with a threshold ($w_i > 0.1$) and put the higher hierarchy in j_1. Then, we can find the involved edges of each component from the map.

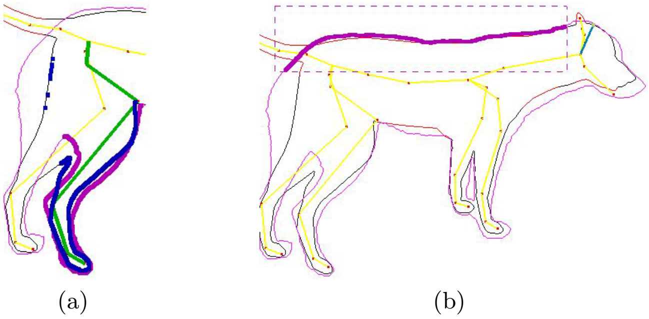


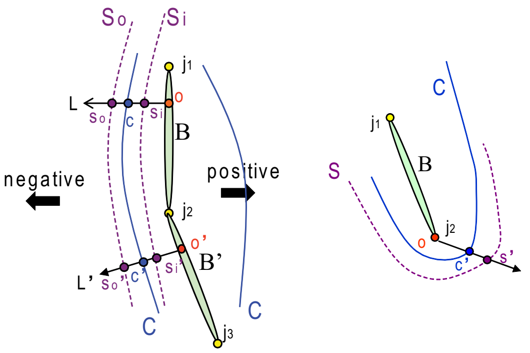
Fig. 5. (a) A partial projected frame, the model's contour overlaps the subject's one. (b) A square selection mechanism.

With the projection matrix obtained in Sec. 3.2 and the component specification, our system projects the involved edges of different components of the reference model onto the image plane, then identifies the contour points of each component. Fig. 5 (a) shows the right hind-leg's bones (green lines) specified by the user and the projected contour (blue strokes) on the image plane. Although we have the subject's contour in each frame, we still need to specify each component's contour as the reference in the pose estimation process. Agrawala *et al.* [10] presented an interactive contour tracking method, but it cannot be used when occlusion occurs. Hence, the user is asked to simply assign the subject's contour of different components in each frame via a square selection mechanism (Fig. 5 (b)).

After the pre-processing is done, the pose estimation is performed automatically by modifying the reference model while referring the information in the previous frame to fit the subject's contour in current frame. The pose estimation contains two processes - limbs rotation and body translation.

Component Rotation Estimation Each bone B in each component has two error items $E = \{e, e_d\}$. To record the difference between the subject's contour of this component and the model's contour which bone 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.

Fig. 6 (a) shows an example about how we calculate the error E. Assume the model's contour C is influenced by the bone B. Then, a point $c \in C$ can be projected onto $\overline{j_1 j_2}$ at point o to form a line L which is pedicular to B, and L



(a) (b)

Fig. 6. Calculation of error data E.

intersects the subject's contour S at point s. Hence, E_p is calculated by all c lie on the right side of B with Eq. (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's contour of B sometimes cannot be projected on line $\overline{j_1 j_2}$, like c' in Fig. 6 (b). Hence, we try to project it onto B's child bone B' first (Fig. 6 (a)). If it still cannot be projected, we choose j_1 or j_2 as o determined by smaller distance to c' (Fig. 6 (b)).

Due to the occlusion or user's input, there are few or even no intersect point at the subject's contour of some components. Hence, we treat their 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 $D\left(E_p\right)$ and $D\left(E_n\right)$ calculated by Eq. (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 that in the previous frame.

$$D(E) = (d_1, d_2) = \|E_{frame} - E_{frame-1}\| = w^k(|e - e'|, |e_d - e'_d|),$$

$$\text{where } (e, e_d) \in E_{frame},\ (e', e'_d) \in E_{frame-1}, \text{ and } k = 1 \text{ if } ee' < 0 \text{ else } k = 0. \qquad (2)$$

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 rotation direction by Algorithm 2, and the direction will be opposite when $E = E_n$. 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

Algorithm 1 Component Rotation 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
```

Algorithm 2 Rotation Direction Determination Algorithm

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

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.

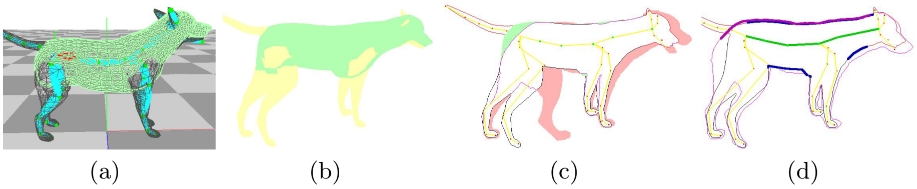


Fig. 7. Translation estimation. (a) The green triangles are specified for the translation estimation. (b) The projection of triangles which are specified for translation estimation. (c) Overlap between the reference model's contour and the subject image (pink region). (d) The model's and subject's contours of the torso component. Notice that there is one side of the subject's contour.

Translation Estimation Only the root joint of the model is translated. We use the region of translation the user specified to estimate the x direction, and estimate the y direction by minimizing $T(E)$ defined in Eq. (3).

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

Fig. 7 (a) shows the triangles specified by the bones to estimate the translation of x, and Fig. 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 Fig. 7 (c) 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 Sec. 3.1 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 and previous frames. 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 3 shows our method of the pose estimation.

Algorithm 3 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
```

5.3 Refinement

The user can specify the amount of frames which the system makes estimation forwardly. There may be some incorrect estimation, and the incorrect result will propagate to the next frame by using our method. Hence, the user can refine the automatic estimated result and propagate the correction backwardly and forwardly. Fig. 8 shows the automatic estimation of the reference model poses in 17 frames, and the bottom-right row shows the correction propagation of dog's left front leg from user's modification in Frame 17.

6 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 30 frames/s and interlace mode. Fig. 9 shows our result of a dog's sitting motion and another motion – a dog's walking motion is shown in Fig. 10. We ignore the tail's motion because it moves too frequently to make estimation.

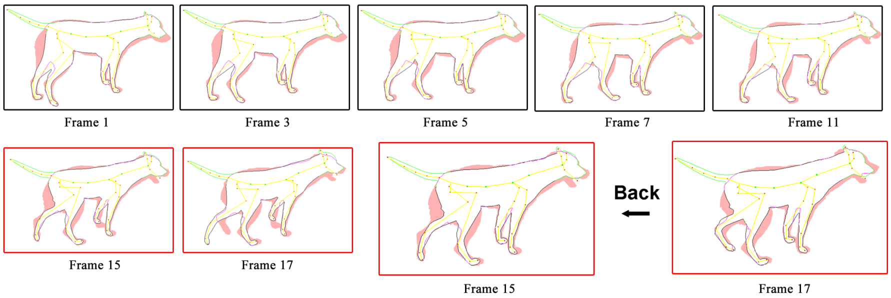


Fig. 8. Refinement. *Top :* The automatically estimated dog's pose in numbers of frame forwardly. *Bottom-Right :* The correction provided by the user is propagated to the backward frame.

7 Conclusions and Future Work

The main advantages of our method are as following:

- **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's 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.

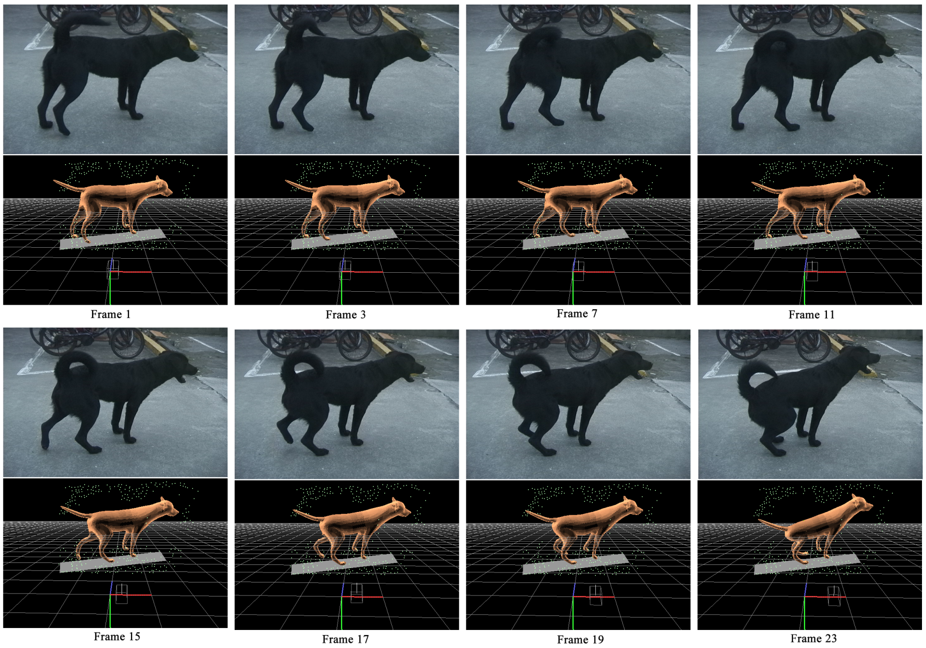
There are two limitations of our method. Due to lack of depth information, our system cannot make estimation of out-subject-plane motion. Although we remain the manual modification of out-subject-plane motion for users, it is still difficult to make accurate estimation. Another limitation is that the unavoidable differences between the reference model and the subject character make our system not robust enough for all scenarios. For the future work, we would like to take into account prior knowledge and example motion data of the subject character in order to reduce the ambiguities and user's interventions.

8 Acknowledgments

This work was partially supported by the National Science Council of Taiwan under NSC97-2221-E-002-224.

References

1. Howe, N.R., Leventon, M.E., Freeman, W.T.: Bayesian reconstruction of 3d human motion from single-camera video. In: Proceedings of 1999 Neural Information Processing Systems. (1999) 820–826



Frame 1

Frame 3

Frame 7

Frame 11

Frame 15

Frame 17

Frame 19

Frame 23

Fig. 9. Result : sit.

2. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3d human figures using 2d image motion. In: Proceedings of 2000 European Conference on Computer Vision. Volume 2. (2000) 702–718
3. Gleicher, M., Ferrier, N.: Evaluating video-based motion capture. In: Proceedings of 2002 Computer Animation. (2002) 75–80
4. de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. ACM Transactions on Graphics **27**(3) (2008) (SIGGRAPH 2008 Conference Proceedings).
5. Vlasic, D., Baran, I., Matusik, W., Popoviċ, J.: Articulated mesh animation from multi-view silhouettes. ACM Transactions on Graphics **27**(3) (2008) (SIGGRAPH 2008 Conference Proceedings).
6. Wilhelms, J., Gelder, A.V.: Combining vision and computer graphics for video motion capture. The Visual Computer **19**(6) (2003) 360–376
7. Favreau, L., Reveret, L., Depraz, C., Cani, M.P.: Animal gaits from video. In: Proceedings of 2004 ACM SIGGRAPH/Eurographics Symposium on Computer animation. (2004) 277–286
8. Rother, C., Kolmogorov, V., Blake, A.: ”grabcut”: interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics **23**(3) (2004) 309–314 (SIGGRAPH 2004 Conference Proceedings).
9. Pollefeys, M., Gool, L.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. International Journal of Computer Vision **59**(3) (2004) 207–232

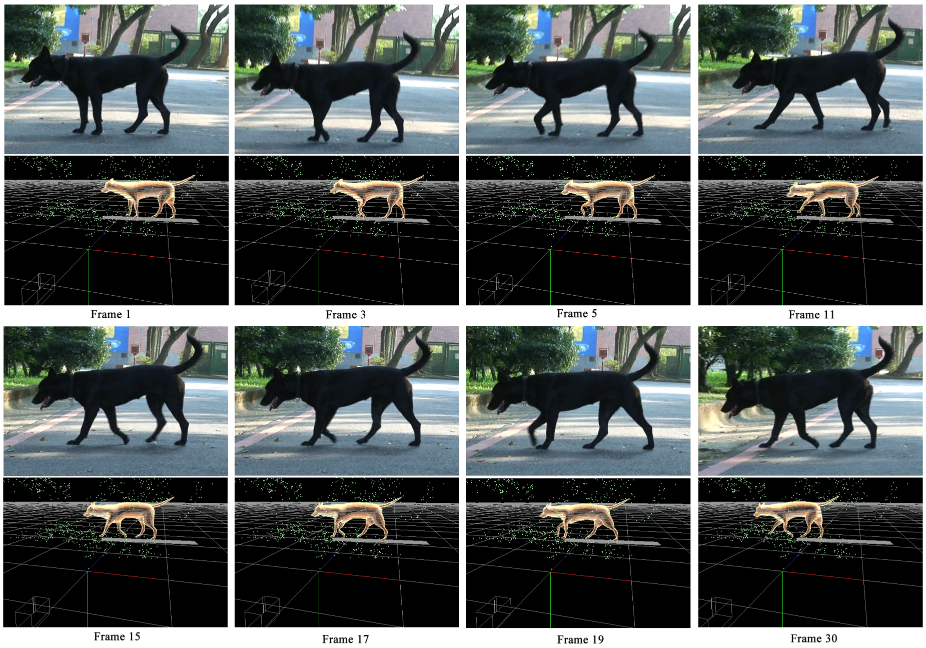


Fig. 10. Result : walk.

10. Agarwala, A., Hertzmann, A., Salesin, D.H., Seitz, S.M.: Keyframe-based tracking for rotoscoping and animation. ACM Transactions on Graphics **23**(3) (2004) 584–591 (SIGGRAPH 2004 Conference Proceedings).
11. Huang, J., Shi, X., Liu, X., Zhou, K., Wei, L.Y., Teng, S.H., Bao, H., Guo, B., Shum, H.Y.: Subspace gradient domain mesh deformation. ACM Transactions on Graphics **25**(3) (2006) 1126–1134 (SIGGRAPH 2006 Conference Proceedings).