Cartoon Motion Capturing and Retargeting by Rigid Shape Manipulation

Mohammad Rastegari
Science & Research Branch Islamic Azad University
Institute for Research in Fundamental Science (IPM)
Tehran, Iran
mohras@csc.kth.se

Mohammad Rouhani
Computer Vision Center (cVc)
Universitat Autonoma de Barcelona (UAB)
Barcelona, Spain
rouhani@cvc.uab.es

Niloofar Gheissari
Center for Vision, Speech and Signal Processing (CVSSP)
University of Surrey
Guildford, Surrey
n.gheissari@surrey.ac.uc

Mir Mohsen Pedram
Dept. of Computer Eng.
Tarbiat Moallem University of Tehran
Tehran, Iran
pedram@tmu.ac.ir

Abstract

Motion capture from live performance has received a lot of attention in the past few years. However, little work has been done on capturing the motions from existing cartoons. This paper presents a novel approach for capturing motion from existing cartoons and retargeting it to new characters in order to animate them. Most current approaches rely on the identification of key shapes and they fail to directly handle articulated shapes or local deformations. In contrast, we propose to use key-points as more efficient descriptors of motion. We use shape context to capture the motion between successive source frames. Then for retargeting the captured motion to a new character, we use an efficient rigid shape manipulation method that handles local deformations. The proposed method relies on user interaction only for the first source frame. Our algorithm has been applied to a set of test cases and the results shows improved performance in preserving the target's visual style particularly for articulated objects.

I. INTRODUCTION

Motion capture and retargeting is the process of recording movements of a source character and retargeting it to another character (in order to animate it). The objective of this research is to develop methods for re-using old cartoons in creating new cartoons. Cartoon motion capture extracts the motion style from old cartoons and cartoon retargeting retargets it to new characters in order to animate them.

It facilitates the process of reusing old cartoons to generate new animations that have the same motions as the old ones but in visually different styles. It can reduce the cost of animation studios and computer game producers significantly. It is also an important tool in editing current animations where the visual look of a character needs to be changed without changing the motion style.

Although motion capture based on live performances has attracted considerable attention in the past few years [1,2,3], little research has been conducted on cartoon motion capture. In fact, to the best of our knowledge only three research papers have been published specifically on cartoon motion capture [4,5,15]. However, there are a number of papers addressing 2D and 3D shape interpolation [11,12,13] or semi-automatic segmentation and inbetweening [14] in order to reuse old cartoons.

A pioneering work on cartoon motion capture is introduced by Bregler et al. [4]. They represent the animation as an interpolation of key-shapes (similar to facial expression animation applications [6,7]). They parameterise a motion by a combination of affine transformations and key-shape interpolations. Then the retargeting phase applies this information to animate the target characters. The result is an animation with a new visual look but the same motion style as of the original cartoon. A drawback of this method is that it requires efficient and reliable non-rigid point matching for *each* frame of the sequence [8]. It also fails to preserve the visual style of target character for articulated objects. Therefore, the authors split the articulate character into sub-parts and deform them separately, which demands significant post processing.

In [5] Wang and Li use thin plate splines (TPS) and shape context both for capturing a motion and retargeting it. An advantage of their approach is that it

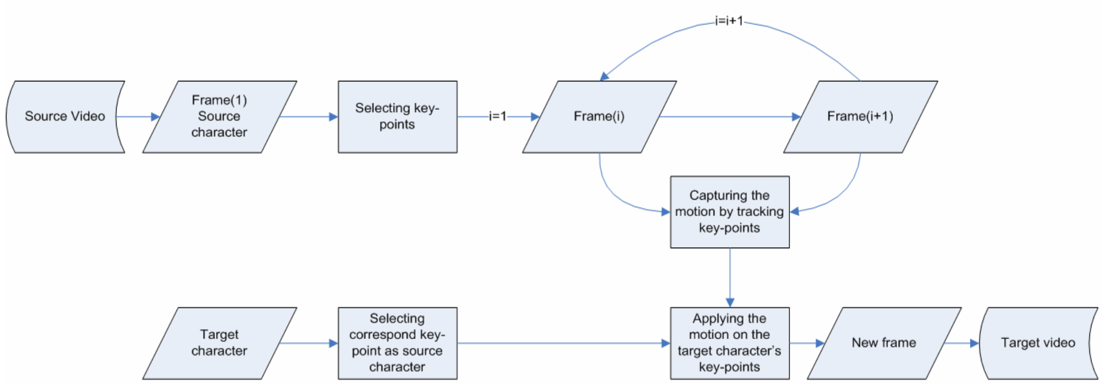

Figure 1 : Outline of the proposed approach.

does not need a prior model and can capture local deformations. However since they retarget the motion only by applying the deformations found by shape context on the source character, they fail to preserve the visual style of the target character. It is a major problem if the target and source characters are very different. This algorithm also requires the source character to pose similar to the target character in the initial frame.

In [15] Rastegari et al. presented a method motivated by [4]. To capture the non-rigid deformations, they first interpolate a given source shape by a linear combination of Fourier descriptors [10] of source key-shape contours. This eliminates the need for finding corresponding points between each consecutive frame. An important aspect of their method is that it is performed in a multi-scale framework that helps preserve the target's visual style. Therefore, unlike the method of Bregler et al[4]. they do not need to split an articulated character into sub-parts. However, since they use global descriptors (i.e. Fourier descriptors), they can only capture global deformations and they fail to handle local deformations successfully. In addition, their method still needs a set of key-shapes as the input.

To overcome the aforementioned drawbacks of the existing methods, we propose using key-points instead of key-shapes to capture the motion deformation. However, for the retargeting phase we use a mesh manipulation method that employs the deformation found by the motion capture phase only to constrain some vertices on the mesh. We summarize the proposed algorithm here and describe it in more details in next sections.

To capture non-rigid deformations, a user first selects a number of key-points only for the first frame (e.g. points on articulation joints). Then the algorithm tracks these key-points through the sequence using shape context and thin-plate spline [16]. This process is described in section II.

In order to retarget the captured motion to another shape (or character), we use the algorithm of Igarashi et al [17] which the authors called: "as-rigid-as possible shape manipulation". This algorithm represents the shape by a triangular mesh. The user moves a number of mesh vertices (used as constrained handles). The algorithm estimates the positions of the remaining free vertices by minimizing the distortion of each triangle. It should be emphasized that in our approach only use user interaction is necessary only for the first frame as the shape context automatically detects point correspondences for the successive ones. The retargeting procedure is summarised as follows.

For retargeting we construct a mesh based on the target shape and ask a user to identify the key-points on the target mesh (corresponding to ones selected on the first source frame). Then we apply the deformation transform computed by TPS to move the key-points selected on the mesh. Our algorithm automatically deforms the remaining free vertices of the mesh. To generate the next frames, we follow the same procedure using the corresponding points automatically found by shape-context [16] (without any user interaction). Section III explains the retargeting process in full detail. The outline of the algorithm is illustrated in Figure 1.

We show empirically that the proposed algorithm preserves the visual style effectively.

II. MOTION CAPTURE BY KEY-POINTS

An advantage of the proposed method is that it only requires capture of the motion of key-points and not key-shapes. This makes the algorithm potentially faster and more feasible for interactive computer games. To capture the motion of selected-key points, we extract contour of the source character for all frames. In our experiments we used the background subtraction method proposed by Juan et al [15].

Lets assume we are given two contours which are extracted from two consecutive frames. We denote them by $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. Each contour is a set of points $\boldsymbol{P}_i$. We have:

$$\boldsymbol{P}_i = [x_i, y_i],$$
$$\boldsymbol{C}_1=\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\},$$
$$\boldsymbol{C}_2=\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_m\}.$$

To compute the deformation between $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ we require a number of point correspondences. We computed the shape context histogram [16] for each point in $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ (shown in Figure 2) as shape descriptors.

Having found a number of point correspondences (lets denoted by $\boldsymbol{p}_i$ in $\boldsymbol{C}_1$ and $\boldsymbol{q}_i$ in $\boldsymbol{C}_2$), we seek a function $\boldsymbol{F}$ that minimizes the energy function $\boldsymbol{E}$ in equation **(1).** .

$$\min_{\boldsymbol{F}}\{E(F) = \sum_i \left\| \boldsymbol{p}_i - \boldsymbol{F}(\boldsymbol{q}_{\pi(i)}) \right\|^2\} \qquad (1).$$

The function F is modelled by a thin-plate spline (TPS) [16] $\boldsymbol{\pi}$ is a permutation function which indexes the matched points. We capture the entire motion of the source character, by computing F for all successive frames. As a results, the selected key-points in the source character are tracked throughout the sequence.

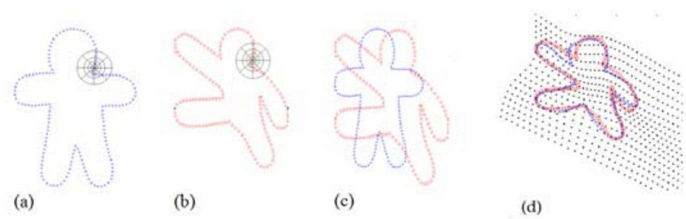

Figure 2: This figure illustrates the deformation of a shape. The shape-context of each point is extracted (a,b) and the deformation is computed by TPS (c,d).

III. Motion Retargeting by Mesh Manipulation

So far, we have captured the motion a number of source key-points selected on the first frame by a user and tracked through the sequence by shape matching. Now we retarget this motion to a new character so that the visual style of the target shape is preserved.

As mentioned in section 1, we use the algorithm Igarashi of et al [17]. The shape is represented by a triangular mesh. A user can move a number of vertices interactively, while the algorithm considers the moved vertices as constrained handles and deforms the remaining free vertices accordingly. This is performed by minimizing the distortion of each triangle in a two-step closed-form algorithm that first finds the rotation for each triangle and then adjusts its scale. Since this algorithm uses quadratic error metrics the minimization problem is reduced to a system of linear equations.

It should be noted here that in our application we only ask the user to select a number of key-points on the first frame of the source sequence. These points are moved automatically by applying the deformation matrix computed in the motion capture phase (by shape context). This means to generate the subsequent frames we do not need any user interaction at all. Figure 3 explains this process.

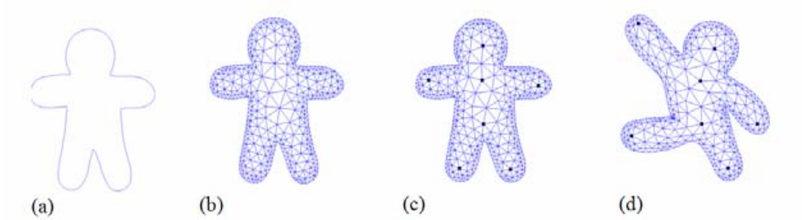

Figure 3: This figure illustrate the manipulation of a shape. (a) is the shape's boundary. (b) is triangular mesh of the boundary. In (c) key-points are selected. (d) shows how moving the key-points result in rigid deformation.

If the pose of the source character in the first frame dramatically differs from that of the target shape, our method is still capable of handling it. This is because using the above mesh manipulation procedure, the user can also manipulate the target mesh so that it complies with the first frame of the source sequence (if required).

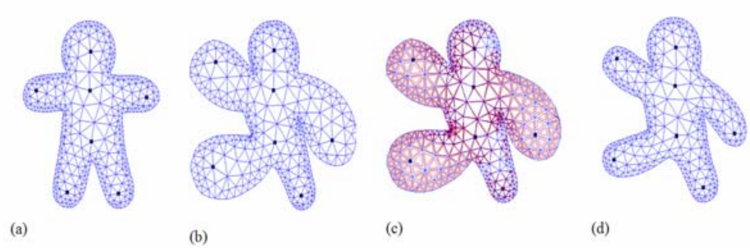

Figure 4: This figure indicate two steps of deformation. Scale-free as step one (b). Scale adjustment as step two which has two part triangles fitting (c) and final result (d).

Step One: scale-free construction

The outcome of the first step in the application of the Igarashi et al algorithm [17] is an intermediate mesh resulting from minimization of an error function which permits only rotations and uniform scalings. This step takes the coordinates of the constrained vertices as input and outputs the remaining vertices of the intermediate mesh.

Assume that the original triangle is $\{v_0, v_1, v_2\}$ (as can be seen in Figure 5.a) and we wish to generate the target triangle $\{v_0', v_1', v_2'\}$ by deformation. We move the vertex v_1 of a triangle while keeping the vertex v_0. So the third vertex, v_2, is free to move. Our objective is to find a new coordinate for v_2 such that the shape of the triangle is maintained. To achieve this Igarashi et al [17] minimise the following error function associated with v'_2 as follows:

$$E_{\{v_2\}} = \left\| v_2^{desired} - v'_2 \right\|^2 \qquad (2).$$

where

$$v_2 = v_0 + x_{01}\overrightarrow{v_0 v_1} + y_{01} R_{90} \overrightarrow{v_0 v_1} \quad \text{and}$$

$$v_2^{desired} = v'_0 + x_{01}\overrightarrow{v'_0 v'_1} + y_{01} R_{90} \overrightarrow{v'_0 v'_1}\,.$$

The coordinates $\{x_{01}, y_{01}\}$ are relative coordinates of v_2 in the local coordinate system defined by v_0 and v_1 and R_{90} is a 90 degree rotation matrix. This step is illustrated in Figure 5. The error for entire mesh can be computed by summing of the error measurements of all triangles (Figure 4.b):

$$E_{1\{V\}} = \sum \left\| V^{desired} - V' \right\|^2 \qquad (3).$$

where $V = (v_{0x}, v_{0y}, v_{1x}, v_{1y}, \cdots, v_{nx}, v_{ny})^T$

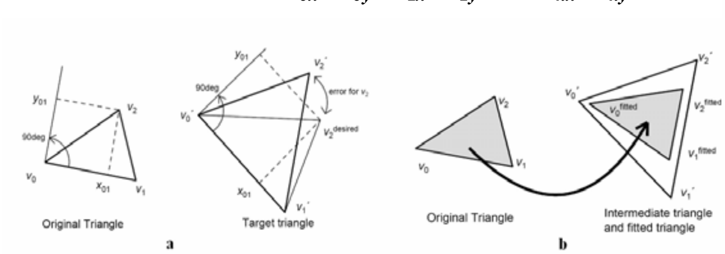

Figure 5: (a) shows the scale-free deformation of a single triangle. -(b) shows fitting of an orginal triangle on a deformed triangle ([17]).

Step Two: non-uniform scale adjustment

The input to this step is the intermediate mesh resulting from the previous step one. This step generates the final mesh by adjusting the scale of the triangles in the mesh which updates coordinates of the free vertices. Given a triangle $\{v_0', v_1', v_2'\}$ in the intermediate result, we find a new triangle $\{v_0^{fitted}, v_1^{fitted}, v_2^{fitted}\}$ that minimises the difference between the vertices of both triangles (Figure 4.c) and (Figure 5.b)

Then, to build the final mesh, we need to solve the problem of minimizing the difference between edges of the two triangles quantified as:

$$E_{2\{v_0'', v_1'', v_2''\}} = \sum_{(i,j)=\{(0,1),(1,2),(2,0)\}} \left\| \overrightarrow{v_i'' v_j''} - \overrightarrow{v_i^{fitted} v_j^{fitted}} \right\|^2 \quad (4).$$

IV. Experimental Results

In this section we present some experimental results of our approach and a comparison with other methods. We implemented our method with MATLAB software on a PC with 1.8 GHz CPU and 512 MG RAM. For shape-context we set the parameter r (number of radius level) to 8 and parameter Θ (number of angles level) to 4. The shape context automatically sampled 150 points on the shape's boundary. To produce a triangular mesh we used Delaunay triangulation from PDE toolbox in MATLAB.

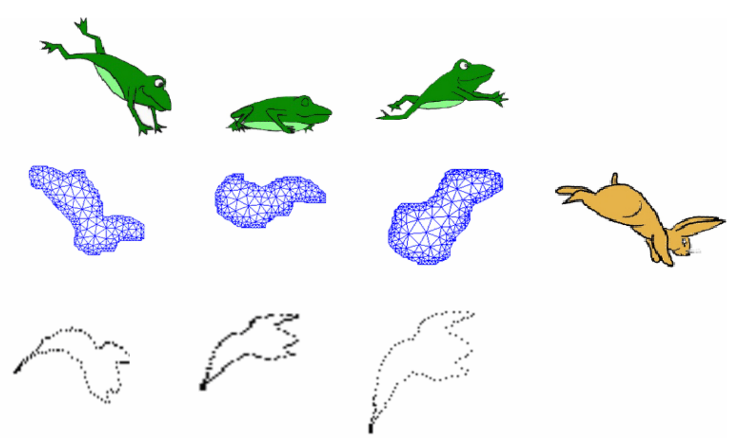

Figure 6: This figure demonstrate retagetting of a frog jump to a rabbit. Second row is our method's results and third row is Wang method's resuls. (sequence borrowed from [4])

The proposed scheme was tested on a number of cartoons and three examples are provided below. As seen in Figure 6, we capture movements of a frog and retarget it to a rabbit. The second row of the figure is

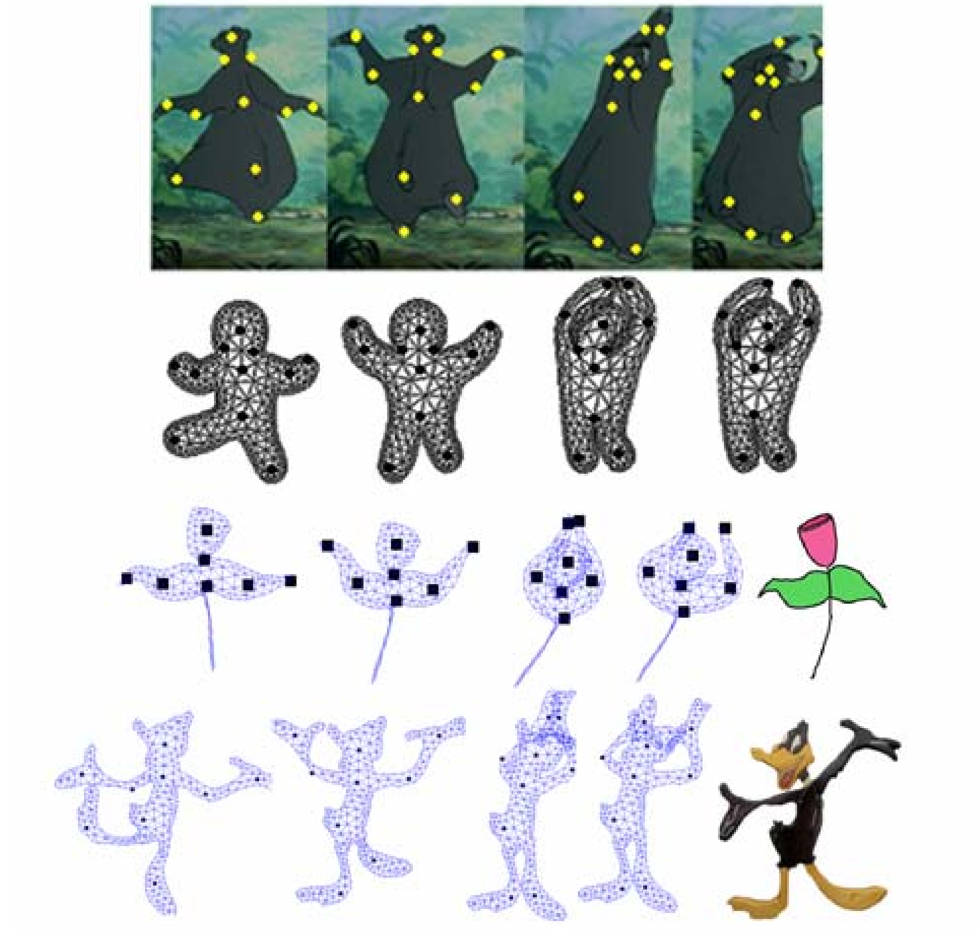

Figure 7: This figure illustrate the motion capturing of Baloo's dance and retargeting to a human shape character and a flower(sequence borrowed from [4,17]]).

the results of our method and the third row is the results of Wang et al. [5]'s method. This figure shows that our method preserved the rabbit's visual style while Wang's method produced distorted results.

In Figure 7, the motion of a bear (Baloo from The Jungle Book[1]), is assigned to an abstract human character, a flower and a bird (Dffy Duck[2]). Although, the visual style of the flower and the bird is significantly different from the bear, our method can capture and retarget the motion effectively. Figure 8 shows a more complex test sequence. The movements of a walking cartoon panther (Pink Panther[3]) character is captured and retargeted to an abstract human character, to a tiger (Tigger from Winnie The Pooh)[4] and to a Barbie character. The pose of Tigger in the source image of Tigger differs from that of Pink Panther; however our method is capable of handling this situation.

This example poses a challenge to any algorithm as this character is highly articulated. This is also true for the Barbie character.

To work with articulated characters (such as this one), Bregler et al. [4] split the shape into sub-parts and deform them separately. We did not need to apply any sub-part decomposition and our method was applied directly to this sequence. This shows an important contribution of the proposed method. Our success can be attributed to the fact that unlike Bregler et al., we use key-points that can describe local deformations effectively.

V. Conclusion

While motion capture from live performance has received substantial attention during the past few years,

[1] Walt Disney Productions
[2] Warner Brothers
[3] DePatie-Freleng Enterprises
[4] Walt Disney Productions

the field of cartoon motion capture is yet to be explored. Cartoon motion capture makes it possible to re-use old cartoons to generate new ones with the same motion style but different visual characters. It is also essential for editing existing cartons. In this paper we presented a new method for cartoon motion capture based on key-point matching and mesh deformation. We used shape context to capture the motion between successive source frames.The re-targeting process is performed by the means of a deformable mesh editing method. We have applied our method to a wide range of image sequences and have shown that its performance is superior to the existing methods in the literature. In particular, we have shown that while preserving the visual style of the target character, we have achieved our goal in eliminating the need for key-shapes or the division into subparts of articulated characters.

Acknowledgement

The authors would like to thank Prof. Mehrdad Mirshams Shahshahani for his great support, valuable discussions and supervision.

References

[1] M. Gleicher. Animation from observation: Motion capture and motion editing. *ACM Computer Graphics*, 33(4):51–54, 1999.

[2] T. B. Moeshund and E. Granum. A survey of computer vision -based human motion capture. *Computer Vision and Image Understanding*, 81:231–268, 2001.

[3] M. Gleicher. Making motion capture useful. *Proc. ACM SIGGRAPH 2001 Course Notes 51*, 2001.

[4] C. Bregler, L. Loeb, E. Chuang, and H. Deshpande. Turning to the masters: motion capturing cartoons. *Proc. ACM SIGGRAPH 2002*, 2002.

[5] H.Wang, H.Li Cartoon Motion Capture by Shape Matching, *Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications* , 0-7695-1784-6/02

[6] PARKE, F. 1972. Computer generated animation of faces. In *ACM National Conferences*, 451–457.

[7] PIGHIN, F., HECKER, J., LISCHINSKI, D., SZELISKI,

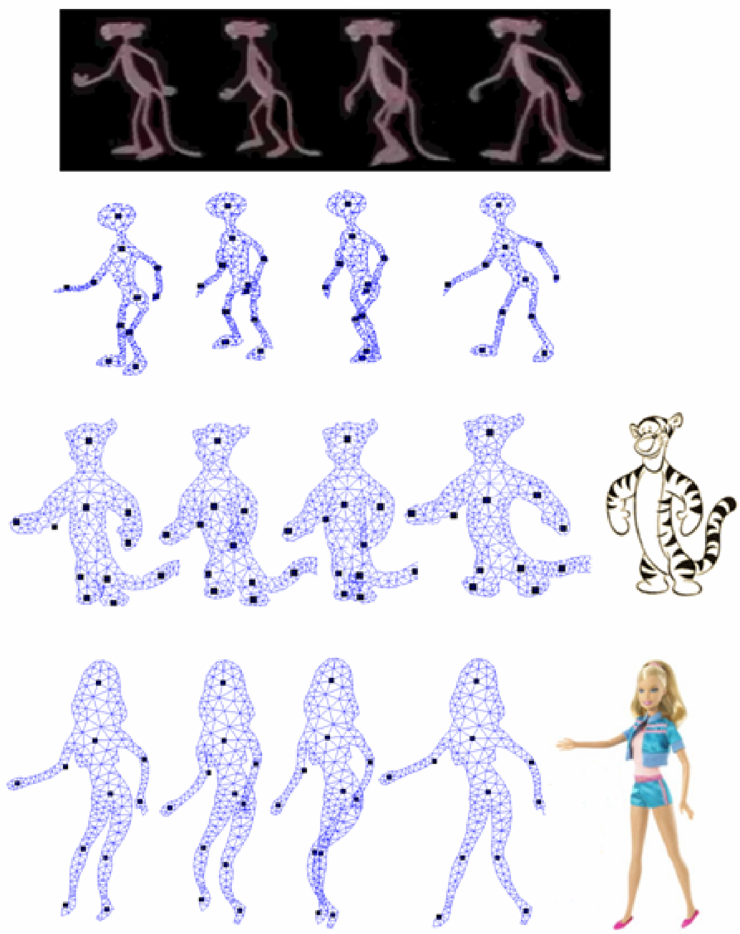

Figure 8: This figure shows motion capturing and retageting of Pink panther's walk to an abstract character and Tigger. (sequence borrowed from [4]).

R., AND SALESIN, D. H. 1998. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH 98 Conference Proceedings*, Addison Wesley, M. Cohen, Ed., Annual Conference Series, ACM SIGGRAPH, 75–84. ISBN 0-89791-999-8.

[8] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR00)*, 2:44–51, June 2000.

[9] GILL, P., MURRAY, W., AND WRIGHT, M. 1981. *Practical Optimization*. Academic Press, London, UK.

[10] D.S. Zhang, G. Lu, A comparative study of Fourier descriptors for shape representation and retrieval. *Proceedings of the Fifth Asian Conference on Computer Vision (ACCV02)*, Melbourne, Australia, January 22–25, 2002, pp. 646–651.

[11] J. P. Lewis, M. Cordner, N. Fong, Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 165 - 172.

[12] A. Hornung, E. Dekker, L. Kobbelt, Character animation from 2D pictures and 3D motion data. *ACM Transactions on Graphics (TOG)*, 2007, Volume 26 , Issue 1, Article No. 1.

[13] ALEXA, M., COHEN-OR, D., and LEVIN, D. As-Rigid-As-Possible Shape Interpolation. In *Proceedings of ACM SIGGRAPH 2000*, pp. 157-164.

[14] C. N. de Juan, B. Bodenheimer, Re-using traditional animation: methods for semi-automatic segmentation and inbetweening. *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2006, SESSION: Animation systems. pp. 223 – 232

[15] M. Rastegari, N. Gheissari: Multi-scale Cartoon Motion Capture and Retargeting without Shape Matching. *IEEE proceeding DICTA* 2008: 320-326

[16] S. Belongie, J. Malik, J. Puzicha: Shape Matching and Object Recognition Using Shape Contexts. IEEE Trans. Pattern Analysis and Machine Intelligent. 24(4): 509-522 (2002)

[17] T. Igarashi, T. Moscovich, J. F. Hughes: As-rigid-as-possible shape manipulation. *ACM Transaction on Graphics*. 24(3): 1134-1141 (2005)