

# Motion Analogies: Automatic Motion Transfer to Different Morphologies

## Abstract

We present a novel approach to transfer motion between two characters with drastically different morphologies. We use a pair of example motions to establish an analogy between the source and target characters. Given a new motion for the source character, our method automatically infers the new motion for the target character using the analogy. Our algorithm is based on the hypothesis that motion similarity among different morphologies can be well represented in the eigenmotion. We cast the retargetting problem as an optimization that solves for the appropriate eigenvectors that represent the coordinations of the character. Our method provides a completely generic and automatic motion retargetting tool for rapid prototyping of character animation with arbitrary morphologies.

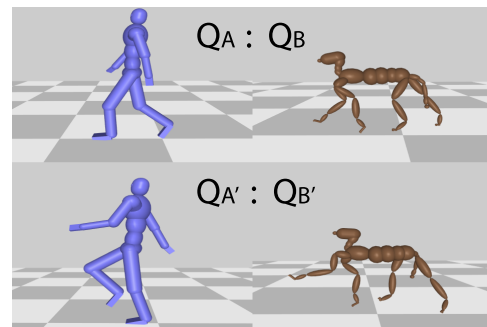
Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Animation—

## 1. Introduction

Automatic synthesis of motion for articulated characters remains a difficult problem to date. Motion capture technology provides a partial solution as it largely simplifies the process of manual creation of expressive and natural human motion. However, without proper authoring tools, existing motion can not be adapted to new situations or different characters with ease. To utilize existing motion capture data, in practice, motion retargetting is of particular importance among all animation techniques. Often, the actor, for whom the motion is captured, and the virtual character do not share the same body type or bone lengths. Directly mapping the motion onto a dissimilar virtual character results in unrealistic motion, even though they share similar skeleton structure. Consequently, most motion retargetting methods focus on retargetting stylistic motion on similar characters or require authoring tools for animators to retarget on more varied characters.

In this paper, we describe an automatic method for motion transfer to different morphologies using an *analogy*, i.e. given two motions  $\mathbf{Q}_A$  and  $\mathbf{Q}_B$  of two completely different characters  $C_A$  and  $C_B$  (e.g. human and a dog) exhibiting similar behavior like walking, we automatically synthesize new motion  $\mathbf{Q}_{B'}$  for  $C_B$  that *mimics* a new given motion  $\mathbf{Q}_{A'}$  for  $C_A$  (e.g. marching) (see Figure 1). We aim for a completely generic algorithm that does not make any assumption about the morphologies of  $C_A$  and  $C_B$  or require authoring tools to

define any correspondence between them. Our method can be used as a rapid prototyping tool that eases the effort in adapting the existing motion to new characters.



**Figure 1:** Automatically synthesized marching motion for dog,  $\mathbf{Q}_{B'}$ , using an analogy of walking motions.

The key insight of our approach is derived from the analysis on the animal motion measured from the real world. We compared walking cycles of a human and a cat using eigen analysis on the motion. We found that the eigenmotion (low-dimensional motion in the space defined by eigenvectors), corresponding to the first few eigenvectors, is strikingly similar between the human and the cat, even though the eigenvectors are in completely different dimensions. Furthermore, when we combined the eigenmotion of the cat with the eigenvectors of the human, we obtain a human walk

in a very cat-like manner. These observations inspired us to revisit the problem of motion retargeting from a completely different perspective: instead of solving for unknown motion on the new character based on the existing motion, our new approach transfers the existing eigenmotion to a new character with unknown eigenvectors.

Intuitively, the first few eigenvectors capture the main *co-ordinations* of a particular movement, and the eigenmotion indicates the execution of those coordinations over time. Based on our observation, we hypothesize that the same movement can be executed in a similar way (the same eigenmotion) by highly varied morphologies with inherently different coordinations (different eigenvectors).

The challenge in our approach is to compute the most important coordinations for synthesizing new motion automatically from the given input sequences. We analyze the two input motions  $\mathbf{Q}_A$  and  $\mathbf{Q}_B$  representing similar content and establish correspondence between the movement of the two characters. We perceive the similarity in the motion by the movement of the point cloud on each of the characters rather than their joint angles. We then use this correspondence to compute the coordinations for motion  $\mathbf{Q}_{B'}$ .

In this paper, we demonstrate transfer of motion in different styles from human character to other creatures e.g. a dog or a spider. These characters are markedly different in number of degrees of freedom (DOFs), body parts and skeleton structures. Our method automatically synthesizes motion for these characters by deriving correspondence from input motions and incorporating any user-specified constraints.

## 2. Related Work

**Motion Retargetting.** Motion retargetting is one of the challenging problems in computer animation. Gleicher [Gle98] first defined the problem as adaptation of an animated motion from one character to another. In this work, motion is retargetted primarily to similar skeleton structures but different bone lengths by solving a space-time optimization problem. Lee and Shin [LS99] proposed further improvements using hierarchical representation for motion and new per frame inverse kinematics (IK) solver along with curve-fitting for smoothness. We also formulate a space-time optimization problem but solve for the coordinations rather than joint trajectories. We tackle a much more challenging problem where the target character can have completely different morphology from the source character. Other methods [CK99, CK00, SLGS01] focus on real-time retargetting on characters with identical topologies, with differences in bone lengths, by tracking the end-effector trajectories and doing per-frame IK. Kulpa *et al.* [KMA05] present a cartesian-space constraint representation of motion which allows to adapt motion to different characters in real-time. In this work, motion retargetting is restricted to human-like figures with different number of bones and bone lengths.

Tak and Ko [TK05] propose physics-based motion retargetting on similar characters using a per-frame Kalman filter framework. Popović and Witkin [PW99] use physics-based method to transform motion of the given character. They present a method for mapping motion onto a simplified representation of the same character. Hodgins and Pollard [HP97] adapted dynamic controllers to characters of similar topology with different physical properties. In our work, we use a kinematic approach and focus on retargetting on highly different morphologies. We show results on morphologies which are radically different from human characters and direct mapping between body parts is ill-defined.

Our work is closely related to Hecker *et al.* [HRE\*08]. They describe an authoring system to create animation for highly-varied morphologies. Characters are created by the user using a palette of available body parts in the authoring environment. To create animations, detailed authoring is required using the available pre-defined body parts. Motion is stored independent of any specific character and can be specialized using IK. In contrast, our algorithm can work with arbitrary animation without needing the characters or the motion to be authored. We automatically compute correspondence of the characters from their respective motions for prototyping new animation.

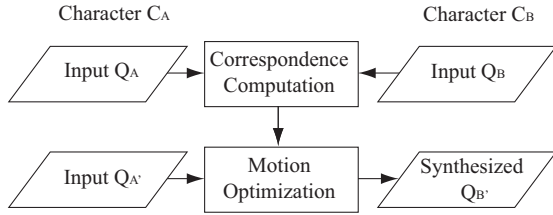
**Identification of Important Components.** Many researchers have explored the idea of analyzing motion to separate the important components of movement. In an earlier work in motion editing [BW95], multi-resolution filtering is used to extract important frequency bands that can be edited to change the style. Unuma *et al.* [UAT95] also proposed similar control for editing style. Shapiro *et al.* [SCF06] extract style components from the motion using Independent Component Analysis (ICA). They demonstrated their results of mixing styles of two motions for similar characters. Our work shares similar idea of decomposing the motion to find the most important coordinations of movement. We synthesize new motion for a different morphology by computing its most important coordinations of movement and use the same style as of the given motion of a different character.

Some recent research work [YL08, KRFMP09] have also exploited the idea of synthesizing or editing motion using important modes from dynamic data. Ye and Liu [YL08] perform eigen analysis of the muscle forces used in a given motion and synthesize perturbed motion by simulating the least important coordinations. Kry *et al.* [KRFMP09] extract the principal vibration modes from dynamic model of the character and synthesize motion (kinematically for complex characters) by combining some vibration modes with selected frequencies. The choice of important modes and their frequencies is left to the user to control the animation. Our alternative approach takes advantage of automatically establishing correspondence and utilizing the eigenmotion of the given animation, to compute the

important coordinations such that the resulting motion exhibits desired similarity in the style of motion.

**Examples or Analogies.** A wide literature of work in graphics [HJO\*01, FTP03, HPP05] has used examples or analogies to transfer style. Tenenbaum and Freeman [TF97] identify handwriting style and faces by separating style and content. Dontcheva *et al.* [DYP03] present an animation system that applies the difference of motion of a real actor and a virtual character by deriving implicit relationships between the actor and character. Young *et al.* [YIS08] extended *Image Analogies* [HJO\*01] to synthesize expressive motion for simple 2D figures. Hsu *et al.* [HPP05] demonstrated translation of style for human motion by computing stylistic differences between motions and applying it to synthesize new motion. They demonstrated the results on identical topologies. To establish correspondence, motion alignment is done by dynamic time warping. In our work, we demonstrate an automatic method that transfers motion to a new morphology while retaining the original style. In addition, we establish correspondence that is independent of the timing and scale of the movements, therefore avoiding time alignment.

### 3. Overview



**Figure 2:** The algorithm takes two similar motions  $\mathbf{Q}_A$  and  $\mathbf{Q}_B$  for different characters  $C_A$  and  $C_B$  respectively, and establishes correspondence between the characters. Then, given a new input motion  $\mathbf{Q}_{A'}$  for  $C_A$ , it synthesizes motion  $\mathbf{Q}_{B'}$  for  $C_B$  which exhibits similar behavior as the new motion.

Figure 2 illustrates our motion transfer algorithm that synthesizes new motion for character  $C_B$  using the input motions. Our method comprises of two main steps:

1. **Correspondence Computation:** Establish correspondence between the two characters based on the movement of the point cloud on the characters in the respective example motions  $\mathbf{Q}_A$  and  $\mathbf{Q}_B$  (Section 4).
2. **Motion Optimization:** Given the character correspondence and a new motion  $\mathbf{Q}_{A'}$ , solve for  $\mathbf{Q}_{B'}$  by finding the important coordinations for the character  $C_B$  performing the same behavior as  $\mathbf{Q}_{A'}$  (Section 5). The correspondence can be obtained from step 1 or specified manually if motion  $\mathbf{Q}_B$  is not available (Section 5.4).

**Notation.** We use the following notation and conventions throughout the paper:

1. For any matrix  $\mathbf{A}$ ,  $\mathbf{a}^j$  denotes the  $j^{\text{th}}$  column and  $\mathbf{a}_i$  denotes the  $i^{\text{th}}$  row of  $\mathbf{A}$ .
2. For a character  $C_Y$ ,  $Y \in \{A, B\}$ , we denote the quantities in the space of joint angles (DOFs) and 3D marker positions as:

Character $C_Y$	Joint space	Cartesian space
Motion Matrix	$\mathbf{Q}_Y$	$\mathbf{X}_Y$
Eigenvectors	$\mathbf{F}_Y$	$\mathbf{E}_Y$
Eigenmotion	$\mathbf{N}_Y$	$\mathbf{M}_Y$
Mean Pose	$\mathbf{v}_Y$	$\boldsymbol{\mu}_Y$

3. See Figure 3 for description of eigenvector matrix  $\mathbf{E}_Y$ .

$$\mathbf{E}_Y = \begin{bmatrix} p_{1,Y} \\ \vdots \\ p_{i,Y} \\ \vdots \\ p_{m,Y} \end{bmatrix} \begin{bmatrix} l & l & k \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{matrix} \rightarrow \mathbf{e}_{Y,p_i}^l \\ \rightarrow \mathbf{E}_{Y,p_i} \\ \rightarrow \mathbf{e}_Y^l \end{matrix}$$

**Figure 3:** For motion in marker space  $\mathbf{X}_Y$ ,  $\mathbf{E}_{Y,p_i} \in \mathbb{R}^{3 \times k}$  is the sub-matrix of the eigenvector matrix  $\mathbf{E}_Y$  corresponding to the  $i^{\text{th}}$  marker on the character  $C_Y$ ,  $p_{i,Y}$ . Likewise,  $\mathbf{e}_{Y,p_i}^l \in \mathbb{R}^3$  denotes the  $l^{\text{th}}$  column of this sub-matrix i.e. components corresponding to marker  $p_{i,Y}$  in the  $l^{\text{th}}$  eigenvector.

### 4. Correspondence between Motions

We represent an articulated character's skeleton in reduced coordinates as a transformation hierarchy of body nodes with 1 to 3 rotation degrees of freedom (DOFs) for each joint and 6 DOFs for the root of the hierarchy representing the global translation and rotation. The motion for a character is represented as a sequence of DOFs in time.

Before finding the correspondence, we align the global orientation of the input motions  $\mathbf{Q}_A$  and  $\mathbf{Q}_B$ , by aligning the trajectories of the root of the characters. Further, we zero the translation DOFs of the root, since we are interested in the coordinated movement of the various body parts of the articulated character.

To compare the movements of the characters, we need to represent their respective motions in one common space. Therefore, from the joint angle (or DOFs) representation of the motion, we compute the motion,  $\mathbf{X}_A$  and  $\mathbf{X}_B$ , of some points or *markers* spread across on the respective skeletons. Each skeleton can have different number of markers that are chosen arbitrarily such that they span all the body parts of the skeleton.

#### 4.1. Marker Correspondence

We aim at establishing correspondence between the characters from the respective input motions based on the simi-

larity of movement of the markers. One potential method of analyzing similarity would be to do compute correlation of movement of every pair of markers on the two skeletons. However, the timing of the motions and the magnitudes of the markers' movements might be very different and would therefore require both spatial and temporal alignment of the motions. To alleviate this problem, we do an eigen analysis (see Appendix A) and separate the coordinations of the movement (eigenvectors) and their execution over time (eigenmotion). Eigenvectors capture the important *directions* of movement of the markers and eigenmotion capture *how much* and *when* the markers are moved along these directions. Therefore to establish correspondence between the motions, we compare the eigenvectors of the two motions.

Since we have decoupled coordinations and their execution in the given motions, we analyze how these coordinations of the two motions are related. Suppose we apply eigenmotion  $\mathbf{M}$  to eigenvectors of both the motions, then the motion for the characters with zero mean can be written as  $\mathbf{E}_A \mathbf{M}$  and  $\mathbf{E}_B \mathbf{M}$  respectively. Note that  $\mathbf{e}_A^l$  and  $\mathbf{e}_B^l$  represent the “direction” of the movement of markers of  $C_A$  and  $C_B$  respectively projected on the  $l^{th}$  eigenvector. Therefore, the actual movement of the markers in these directions is given by  $\mathbf{e}_A^l \mathbf{m}_l$  and  $\mathbf{e}_B^l \mathbf{m}_l$  respectively.

Let us consider the  $l^{th}$  eigenvector. For a marker  $p_{j,A}$  of character  $C_A$  and  $p_{i,B}$  of  $C_B$ ,  $\mathbf{e}_{A,p_j}^l$  and  $\mathbf{e}_{B,p_i}^l$  represent the three components in the  $l^{th}$  eigenvector corresponding to the respective markers (see Figure 3). If the angle between  $\mathbf{e}_{A,p_j}^l$  and  $\mathbf{e}_{B,p_i}^l$  is close to zero, the markers  $p_{j,A}$  and  $p_{i,B}$  always move in the same direction, since the motion trajectory  $\mathbf{e}_{B,p_i}^l \mathbf{m}_l$  will just be a scaled version of  $\mathbf{e}_{A,p_j}^l \mathbf{m}_l$ .

Based on this observation, we compute a set of corresponding markers of  $C_A$  for markers of  $C_B$ . Matching criterion or *match score* is defined as the cosine of angle between the components of the eigenvector corresponding to the markers:

$$m(\mathbf{e}_{B,p_i}^l, \mathbf{e}_{A,p_j}^l) = \frac{\mathbf{e}_{B,p_i}^l \cdot \mathbf{e}_{A,p_j}^l}{\|\mathbf{e}_{B,p_i}^l\| \|\mathbf{e}_{A,p_j}^l\|} \quad (1)$$

If the absolute value of the score is greater than a certain threshold, the markers are said to match. The algorithm for establishing correspondences based on  $l^{th}$  eigenvectors is described below. It compares each marker  $p_{i,B}$  of  $C_B$  with each marker  $p_{j,A}$  of  $C_A$  and returns a set of matches  $S_{A,B}$ .

```

proc FINDCORRESPONDENCE( $\mathbf{e}_A^l, \mathbf{e}_B^l, T$ )
   $S_{A,B} \leftarrow \emptyset$ 
  for each  $p_{i,B}$  of  $C_B$ , do
     $S_{A,i} \leftarrow \emptyset$ 
    for each  $p_{j,A}$  of  $C_A$ , do
      if  $|m(\mathbf{e}_{B,p_i}^l, \mathbf{e}_{A,p_j}^l)| \geq T$ , add  $p_{j,A}$  to  $S_{A,i}$ 
    end for
  end for

```

```

    if  $S_{A,i} \neq \emptyset$ , add  $S_{A,i}$  to  $S_{A,B}$ 
  end for
return  $S_{A,B}$ 
end proc

```

We keep the threshold  $T$  between 0.8 to 0.9, depending on the number of matched markers  $p_{i,B}$ . Note that the cosine angle can be negative which simply means that the markers move in the opposite direction. Further, not all markers in  $C_B$  necessarily match to any marker in  $C_A$  and a marker  $p_{i,B}$  can match to a set of markers  $\{p_{j,A}\}$  or  $S_{A,i}$ .

**Scaling between marker movements.** Match score defined in Equation (1) captures the closeness in the directions of movement of markers  $p_{i,B}$  and  $p_{j,A}$ . To capture the magnitude of relative movement of the markers, we scale the match score as:

$$s(\mathbf{e}_{B,p_i}^l, \mathbf{e}_{A,p_j}^l) = \frac{\|\mathbf{e}_{B,p_i}^l\|}{\|\mathbf{e}_{A,p_j}^l\|} m(\mathbf{e}_{B,p_i}^l, \mathbf{e}_{A,p_j}^l) \quad (2)$$

Intuitively, the magnitude  $\|\mathbf{e}_{A,p_j}^l\|$  or  $\|\mathbf{e}_{B,p_i}^l\|$  captures the magnitude of the movement of the marker in 3D space. For example, this magnitude would be greater for a marker of an adult compared to the corresponding marker of a child in their normal walking motions.

For any matched pair of markers  $(p_{i,B}, p_{j,A})$  in  $S_{A,B}$ , the cosine of the angle is high, therefore, the scale essentially captures how much  $p_{i,B}$  moves with respect to  $p_{j,A}$ . For markers moving in opposite directions, the scale is negative.

## 4.2. Refinement

The correspondence established using FINDCORRESPONDENCE is based on motions  $\mathbf{X}_A$  and  $\mathbf{X}_B$  only. This implies that a marker  $p_{i,B}$  can match to a set of markers  $\{p_{j,A}\}$ , that move in a similar manner in  $\mathbf{X}_A$ , but not necessarily in general. For example, markers on the arm may move similar to those on the thighs while walking but may move differently in other motions. Therefore, we use information from the motion  $\mathbf{X}_{A'}$  to select markers which exhibit similarity in both motions  $\mathbf{X}_A$  and  $\mathbf{X}_{A'}$ .

In  $S_{A,B}$ , for each marker  $p_{i,B}$ , we have a set of matched markers  $\{p_{j,A}\}$ . These markers match well with themselves in motion  $\mathbf{X}_A$  i.e. absolute value of the match score (from Equation (1)) is high, since they all match well with  $p_{i,B}$ . Therefore, we compute the similarity of markers within the set  $\{p_{j,A}\}$  based on their match score from  $\mathbf{X}_{A'}$  as well i.e. the markers with high match score in  $\mathbf{X}_A$  should also have a high score in  $\mathbf{X}_{A'}$ . In addition to the closeness in directions measured by the match score (Equation (1)), we want the scales (Equation (2)) for the markers in both the motions to agree as well. Therefore, the metric for similarity  $r_{j1,j2}$  for

each pair of markers  $(p_{j1,A}, p_{j2,A})$  in the set is defined as:

$$r_{j1,j2} = \underbrace{m(\mathbf{e}_{A',p_{j1}}^l, \mathbf{e}_{A',p_{j2}}^l)}_{\text{match score}} \underbrace{\min \left( \left| \frac{s_{j1,j2}^A}{s_{j1,j2}^{A'}} \right|, \left| \frac{s_{j1,j2}^{A'}}{s_{j1,j2}^A} \right| \right)}_{\text{scale score}} \quad (3)$$

where  $s_{j1,j2}^A$  and  $s_{j1,j2}^{A'}$  are shorthand for  $s(\mathbf{e}_{A,p_{j1}}^l, \mathbf{e}_{A,p_{j2}}^l)$  and  $s(\mathbf{e}_{A',p_{j1}}^l, \mathbf{e}_{A',p_{j2}}^l)$  respectively, computed from Equation (2) and  $m(\mathbf{e}_{A',p_{j1}}^l, \mathbf{e}_{A',p_{j2}}^l)$  is computed from Equation (1). The match score is required even when the scale includes the cosine of the angle, since both the angle and the magnitudes can change in such a way that the scale (Equation (2)) remains the same.

If the metric  $r_{j1,j2}$  is above a threshold  $T_s$  (i.e. both the direction and the magnitude do not change drastically), the markers are deemed similar. We choose a threshold of  $T_s = 0.8$  in our implementation. Finally we choose maximum number of markers similar to each other as the set of refined matches for  $p_{i,B}$ .

## 5. Optimization Problem for Motion Transfer

Given a new motion  $\mathbf{Q}_{A'}$  for character  $C_A$ , we want to compute the new motion for  $C_B$ ,  $\mathbf{Q}_{B'}$ , using the correspondence established in the previous section. Rather than directly solving for the joint DOFs  $\mathbf{q}_{B'}^k$  at each frame  $k$ , we solve for the basis vectors  $\mathbf{F}_{B'}$  representing coordinations for  $C_B$  and the mean pose  $\mathbf{v}_{B'}$  while using the eigenmotion  $\mathbf{N}_{A'}$  of  $\mathbf{Q}_{A'}$ .

We formulate a space-time optimization problem to solve for the unknowns,  $\mathbf{F}_{B'}$  and  $\mathbf{v}_{B'}$ . The DOFs at time sample  $k$ ,  $\mathbf{q}_{B'}^k$  ( $k^{\text{th}}$  column of motion matrix  $\mathbf{Q}_{B'}$ ), are then reconstructed as:

$$\mathbf{q}_{B'}^k = \mathbf{v}_{B'} + \mathbf{F}_{B'} \mathbf{n}_{A'}^k \quad (4)$$

Note that the number of unknowns is independent of number of frames and just depends on number of DOFs and number of basis vectors being solved for. In all our examples, we solve for first five basis vectors.

We solve for the coordinations in joint DOFs space, since joint angles provide a minimal representation of the pose of the character, and therefore avoid the need of introducing additional constraints.

### 5.1. Correspondence Objective

Based on  $l^{\text{th}}$  eigenvector, we compute the marker correspondence,  $S_{A,B} = \text{FINDCORRESPONDENCE}(\mathbf{e}_{A'}^l, \mathbf{e}_B^l, T)$ .

For a marker  $p_{i,B}$  matched to a marker  $p_{j,A}$  in the set  $S_{A,B}$ , the direction of movement of  $p_{i,B}$  is close to that of  $p_{j,A}$  in motions  $\mathbf{X}_B$  and  $\mathbf{X}_A$  respectively. We aim at computing the new DOF motion  $\mathbf{Q}_{B'}$  such that this relation holds for markers  $p_{i,B}$  and  $p_{j,A}$  in motions  $\mathbf{X}_{B'}$  and  $\mathbf{X}_{A'}$  as well.

For each match  $S_{A,i} \in S_{A,B}$ , marker  $p_{i,B}$  matches a set

of  $N_i$  markers  $\{p_{j,A}\}$ . The position (without the mean) of each of these markers in  $\mathbf{X}_{A'}$  at time sample  $k$  is given by  $\mathbf{E}_{A',p_j} \mathbf{m}_{A'}^k$ . We define the desired position of the marker  $p_{i,B}$  (without the mean) as an average of the scaled positions of the matched markers,  $\frac{\sum_{j=1}^{N_i} s_{ij} \mathbf{E}_{A',p_j} \mathbf{m}_{A'}^k}{N_i}$ , where  $s_{ij}$  stands for  $s(\mathbf{e}_{B,p_i}^l, \mathbf{e}_{A,p_j}^l)$  defined in Equation (2).

Position of the marker  $p_{i,B}$  at time sample  $k$ ,  $\mathbf{x}_{p_i}(\mathbf{q}_{B'}^k)$ , is computed from the DOFs  $\mathbf{q}_{B'}^k$  using the transformation hierarchy of the skeleton. The mean of the computed marker positions in all the frames is denoted as  $\boldsymbol{\mu}_{p_i}(\mathbf{Q}_{B'})$ . The correspondence objective  $G_i^l$  for the marker  $p_{i,B}$  can now be defined as:

$$G_i^l = \left\| \mathbf{x}_{p_i}(\mathbf{q}_{B'}^k) - \left( \boldsymbol{\mu}_{p_i}(\mathbf{Q}_{B'}) + \frac{\sum_{j=1}^{N_i} s_{ij} \mathbf{E}_{A',p_j} \mathbf{m}_{A'}^k}{N_i} \right) \right\|^2 \quad (5)$$

where  $\mathbf{q}_{B'}^k$  is computed using Equation (4).

These objectives for different eigenvectors can be weighted and combined to form a single objective function. However, in our experiments, it was sufficient to use only the first eigenvector, since not much improvement was obtained by incorporating objectives for other eigenvectors.

### 5.2. Constraints

To our non-linear optimization problem, we add some constraints to help solve the optimization better. Although the problem is highly under-constrained, the number of parameters in our method is significantly less than that in case of solving for DOFs in all the frames. As an example, for a character  $C_B$  with 50 DOFs, solving for 5 basis vectors results in 250 parameters. Synthesizing 100 frames of animation otherwise results in 5000 parameters.

**Orthonormal Constraints.** We enforce orthonormal constraints on the basis matrix,  $\mathbf{F}_{B'}$ , that we solve for:

$$\mathbf{f}_{B'}^i \cdot \mathbf{f}_{B'}^j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \quad (6)$$

The normalization constraint helps contain the magnitude of the numbers in the columns of  $\mathbf{F}_{B'}$ .

**Pose Constraints.** We can add pose constraints on DOFs or positional constraints for markers at certain times to direct the optimization. Since the problem is largely under-constrained, this flexibility gives us an advantage of being able to synthesize different looking motions yet exhibiting similar behavior present in the input motion  $\mathbf{Q}_{A'}$ . Adding more than one pose constraint can sometimes lead to infeasibility of the problem due to the orthonormal constraints. In that case, we can relax the orthonormal constraints and solve for the optimization. Relaxing the orthonormal constraints is equivalent to modifying the eigenmotion  $\mathbf{N}_{A'}$  we wanted to use to synthesize  $\mathbf{Q}_{B'}$ . This



is acceptable since eigenmotion represents the execution, which can be modified while retaining the important characteristics of the motion.

### 5.3. Initialization

Initialization plays an important role in solving the optimization, since the problem is highly non-convex. We initialize  $\mathbf{F}_{B'}$  and  $\mathbf{v}_{B'}$  as  $\mathbf{F}_B$  and  $\mathbf{v}_B$  respectively, as they provide with natural coordinations and a pose to initiate the optimization. The resulting motion would be close to the input motion  $\mathbf{Q}_B$  due to the same mean pose, though it can be modified as desired, by adding pose constraints. As an alternative, we can also modify the mean pose directly before starting the optimization to get different looking motion.

### 5.4. Solving without $\mathbf{Q}_B$

The motion  $\mathbf{Q}_B$  for character  $C_B$  can be obtained by motion capture data or is procedurally generated. For non-human characters, this motion may be hard to get. Therefore, we slightly modify our method to synthesize motion  $\mathbf{Q}_{B'}$  from  $\mathbf{Q}_{A'}$  without using information from  $\mathbf{Q}_B$  and  $\mathbf{Q}_A$ .  $\mathbf{Q}_{B'}$ , along with  $\mathbf{Q}_A$ , is used to get the marker correspondence  $S_{A,B}$  and the initialization of the parameters for optimization.

For simple characters, we can by-pass the FINDCORRESPONDENCE step (Section 4) and choose the matching markers and define their respective scales manually (since we do not have eigenvectors for any motion of  $C_B$ ).

Without a good initialization of the mean pose  $\mathbf{v}_{B'}$ , the optimization has no information or guidelines to synthesize meaningful movements for the character. Therefore, to initialize the mean pose, we can design an appropriate pose by inverse kinematics (IK). In our experience, this initialization of the mean pose is sufficient to guide the optimization process. We initialize the basis set  $\mathbf{F}_{B'}$  as a diagonal matrix with 1.0 as each diagonal element. Since the mean pose initialization might not be very accurate, we can add additional pose constraints as before to guide the optimization.

## 6. Results

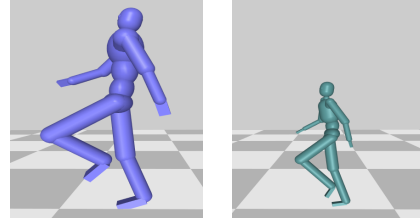
We now present our results of automatic motion transfer using the techniques described in Section 4 and Section 5. We used SNOPT [GSM96] to solve the space-time optimization problem in Section 5. Synthesis time for all the examples described in this section took under 1 minute to solve.

As mentioned earlier, we do not transfer the root translation in our optimization. Therefore, we use the translation of the motion  $\mathbf{Q}_{A'}$  for  $\mathbf{Q}_{B'}$  as well. This root translation may be scaled according to the size of the characters. The next step before outputting the final motion is to perform IK to correct the foot constraints like slipping and penetration. The timing

of these constraints is extracted from foot positions of the input motion  $\mathbf{Q}_{A'}$ , since the same eigenmotion  $N_{A'}$  is used to synthesize  $\mathbf{Q}_{B'}$ .

In all the following examples, we use a human adult walking motion as the input  $\mathbf{Q}_A$ . Adult human skeleton comprises of 18 limbs and 42 DOFs. We refer the reader to the supplemental video for full animations.

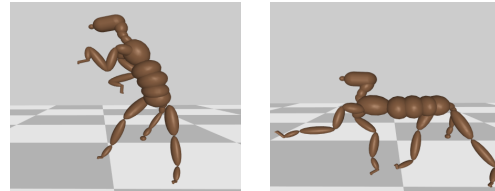
**Child march.** We choose human adult marching motion as the input  $\mathbf{Q}_{A'}$  and a child walking motion as the input  $\mathbf{Q}_B$ . This examples illustrates the motion transfer method for a child skeleton with similar morphology but has 35 DOFs and different bone lengths as compared to that of adult human. In the synthesized motion without any pose constraints, the hands do not stretch. To correct this, we provided two pose constraints designed by IK at frames 44 and 80. In addition, we relaxed orthogonal condition for the bases (Equation (6)) after adding the pose constraints. Figure 4 compares the synthesized child marching with the adult marching motion.



**Figure 4:** Comparison of synthesized child march with adult march. Pose constraints are added to help stretch the arms.

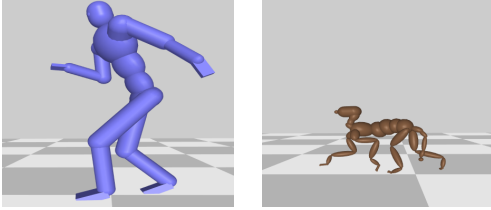
**Dog march.** We choose human adult marching motion as the input  $\mathbf{Q}_{A'}$  and a dog walking sequence as  $\mathbf{Q}_B$ . This example illustrates our method on a very different morphology with 30 limbs and 81 DOFs. We synthesize dog marching motion without any pose constraint. The result is compared in Figure 1.

**Dog march with pose.** To synthesize marching motion using only two feet, we added pose constraint at frame 10. The input motions are exactly the same as previous example for synthesizing dog marching motion. The two marching motions for dog with and without pose are compared in Figure 5. The two motions being noticeably different due to the additional pose constraint, demonstrate similar marching behavior.



**Figure 5:** Synthesized marching motions for dog with and without pose constraint.

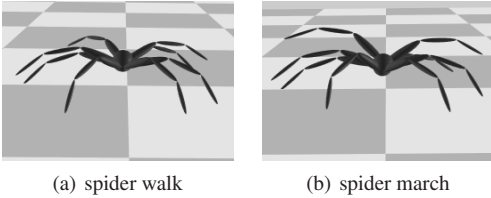
**Dog sneaky walk.** To demonstrate variability of our method, we choose a sneaky walking motion as the new input  $\mathbf{Q}_{A'}$  for the adult human skeleton. Motions  $\mathbf{Q}_A$  and  $\mathbf{Q}_B$  are same as before. We create a crouching pose for the dog skeleton and add a pose constraint at the first frame. Synthesized motion has the characteristic sneaky behavior of the human motion (Figure 6).



**Figure 6:** Comparison of human sneaky walk and synthesized dog sneaky walk.

**Spider walk using manual correspondence.** We create a skeleton for a spider that has 26 limbs and 48 DOFs. We do not have any given motion for the spider. Therefore, we manually specify the correspondence of markers of the spider and the human adult (Section 5.4). We map the tip of one set of alternate legs of the spider to the left foot of the human and other set of alternate legs to right foot. We choose a scaling of 0.8 (Equation (2)) for the matched markers. The result is depicted in Figure 7(a).

**Spider march.** Input motions  $\mathbf{Q}_A$  and  $\mathbf{Q}_{A'}$  are chosen as human adult walk and march respectively. We use the spider's walking motion synthesized in the previous example as our input motion  $\mathbf{Q}_B$  to synthesize its marching motion  $\mathbf{Q}_{B'}$  (Figure 7).



**Figure 7:** Comparison of synthesized walk and march. The spider lifts its legs higher while marching as compared to walking.

**Dog walk using manual correspondence.** To highlight the benefit of using example motion, as compared to manually specifying the correspondence, to synthesize new motion, we choose the dog (30 limbs, 81 DOFs) which is much more complex than a spider. We choose human adult walk as  $\mathbf{Q}_{A'}$  and synthesize walking motion  $\mathbf{Q}_{B'}$  for the dog. We intuitively match some markers for the dog to those of human. In our experience, it was much harder to specify this correspondence because of very different movement of dogs legs and the associated scale of movement. We get poor results

and the dog legs look rigid because of inaccurate matches and scales.

## 7. Conclusion

We proposed a novel approach to motion retargeting on different characters using motion analogies. Our algorithm is derived from the hypothesis that motion similarity among different morphologies can be well represented in the eigenmotion. Therefore, we cast the retargeting problem as an optimization that solves for the appropriate eigenvectors of the new character. Our approach is generic and automatic. We do not make any assumption about the dimensions or the skeletal structures of the characters. In essence, we present a fully automatic rapid prototyping tool that allows the animator to jump start the process of motion creation for arbitrary characters.

We discuss certain limitations that our method suffers from. Firstly, the resulting motion requires refinement using an external animation tool. Although we provide a user-in-the-loop IK method to clean the foot slide after the retargeting, further improvement is needed to bring the results to the production quality. Since the problem is largely under constrained, one promising future direction is to leverage the user's guidance in the optimization process.

Secondly, we transfer the eigenmotion of motion  $\mathbf{Q}_{A'}$  to  $\mathbf{Q}_{B'}$  without modification. However, the eigenmotions of input motions may have stylistic differences which are not retained in this process. Therefore, an appropriate transformation may be computed and applied to the eigenmotion for  $\mathbf{Q}_{B'}$ . Simple transformations like time-warping are easy to realize, though more sophisticated alignment techniques [HPP05] for these reduced space motions may be needed.

Our method may be extended to analyze the important coordinations of the muscle usage of the characters from input motions and automatically transfer the motion respecting the character's dynamic model. This helps synthesizing a wide range of realistic motions by modifying the physical properties, yet retaining similar motion style.

## Appendix A: Eigen Decomposition of Motion

A motion matrix  $\mathbf{X}$  can be decomposed such that pose representation  $\mathbf{x}^j$  (either in DOFs or marker space) at time sample  $j$  can be represented as:

$$\mathbf{x}^j \approx \boldsymbol{\mu} + \mathbf{E}\mathbf{m}^j \quad (7)$$

where  $\mathbf{E} \in \mathbb{R}^{n \times k}$  is the eigenvector matrix with  $i^{th}$  column as the  $i^{th}$  eigenvector  $\mathbf{e}^i$ ,  $k < n$  is the number of principal components,  $\boldsymbol{\mu} \in \mathbb{R}^n$  is the mean of all time samples of the motion, and  $\mathbf{m}^j \in \mathbb{R}^k$  represents the reduced coordinates at time  $j$  in this eigenspace ( $j^{th}$  column of eigenmotion  $\mathbf{M}$ ).

## References

- [BW95] BRUDERLIN A., WILLIAMS L.: Motion signal processing. In *SIGGRAPH* (Aug. 1995), pp. 97–104. [2](#)
- [CK99] CHOI K.-J., KO H.-S.: On-line motion retargetting. In *Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 1999), IEEE Computer Society, p. 32. [2](#)
- [CK00] CHOI K.-J., KO H.-S.: On-line motion retargetting. *Journal of Visualization and Computer Animation* 11 (2000), 223–235. [2](#)
- [DYP03] DONTCHEVA M., YNGVE G., POPOVIĆ Z.: Layered acting for character animation. *ACM Trans. Graph. (SIGGRAPH)* 22, 3 (2003), 409–416. [3](#)
- [FTP03] FREEMAN W. T., TENENBAUM J. B., PASZTOR E. C.: Learning style translation for the lines of a drawing. *ACM Trans. Graph.* 22, 1 (2003), 33–46. [3](#)
- [Gle98] GLEICHER M.: Retargeting motion to new characters. In *SIGGRAPH* (July 1998), pp. 33–42. [2](#)
- [GSM96] GILL P., SAUNDERS M., MURRAY W.: *SNOPT: An SQP Algorithm for Large-scale Constrained Optimization*. Tech. Rep. NA 96-2, University of California, San Diego, 1996. [6](#)
- [HJO\*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *SIGGRAPH* (New York, NY, USA, 2001), ACM, pp. 327–340. [3](#)
- [HP97] HODGINS J. K., POLLARD N. S.: Adapting simulated behaviors for new characters. pp. 153–162. [2](#)
- [HPP05] HSU E., PULLI K., POPOVIĆ J.: Style translation for human motion. In *ACM Trans. Graph. (SIGGRAPH)* (New York, NY, USA, 2005), ACM, pp. 1082–1089. [3](#), [7](#)
- [HRE\*08] HECKER C., RAABE B., ENSLOW R. W., DEWEESE J., MAYNARD J., VAN PROOIJEN K.: Real-time motion retargeting to highly varied user-created morphologies. *ACM Trans. Graph. (SIGGRAPH)* 27, 3 (2008), 1–11. [2](#)
- [KMA05] KULPA R., MULTON F., ARNALDI B.: Morphology-independent representation of motions for interactive human-like animation. *Computer Graphics Forum* 24, 2 (2005), 343–352. [2](#)
- [KRFMP09] KRY P. G., REVERET L., FAURE F., M.-P.CANI: Modal locomotion: Animating virtual characters with natural vibrations. *Computer Graphics Forum (Eurographics)* 28, 2 (2009). [2](#)
- [LS99] LEE J., SHIN S. Y.: A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH* (Aug. 1999). [2](#)
- [PW99] POPOVIĆ Z., WITKIN A.: Physically based motion transformation. In *SIGGRAPH* (Aug. 1999), pp. 11–20. [2](#)
- [SCF06] SHAPIRO A., CAO Y., FALOUTSOS P.: Style components. In *Graphics interface* (Toronto, Ont., Canada, Canada, 2006), Canadian Information Processing Society, pp. 33–39. [2](#)
- [SLGS01] SHIN H. J., LEE J., GLEICHER M., SHIN S. Y.: Computer puppetry: An importance-based approach. *ACM Trans. on Graphics* 20, 2 (Apr. 2001), 67–94. [2](#)
- [TF97] TENENBAUM J. B., FREEMAN W. T.: Separating style and content. In *Advances in Neural Information Processing Systems* (1997), vol. 9, p. 662. [3](#)
- [TK05] TAK S., KO H.-S.: A physically-based motion retargeting filter. *ACM Trans. Graph.* 24, 1 (2005), 98–117. [2](#)
- [UAT95] UNUMA M., ANJYO K., TAKEUCHI R.: Fourier principles for emotion-based human figure animation. In *SIGGRAPH* (1995), pp. 91–96. [2](#)
- [YIS08] YOUNG J., IGARASHI T., SHARLIN E.: Puppet master: Designing reactive character behavior by demonstration. In *Symposium on Computer Animation* (2008). [3](#)
- [YL08] YE Y., LIU C. K.: Animating responsive characters with dynamic constraints in near-unactuated coordinates. *ACM Trans. Graph. (SIGGRAPH Asia)* 27, 5 (2008), 1–5. [2](#)