Computer Animation

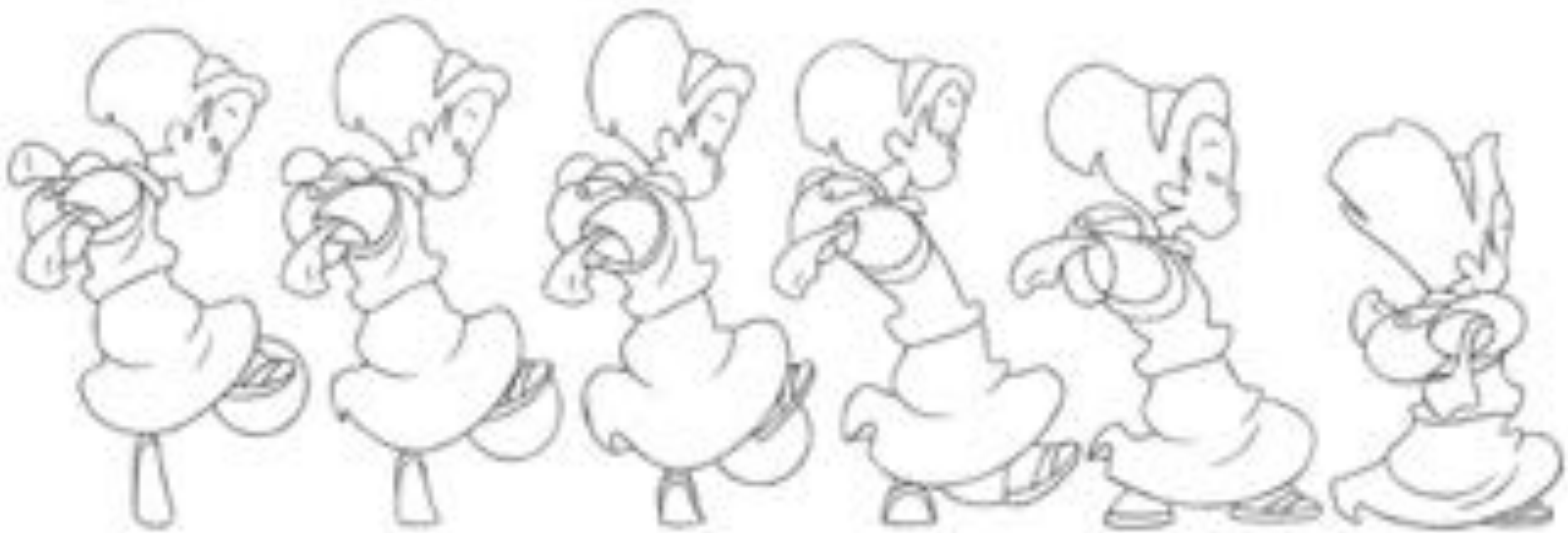
Lesson 2 - Keyframe animation

Remi Ronfard, Nov 2019

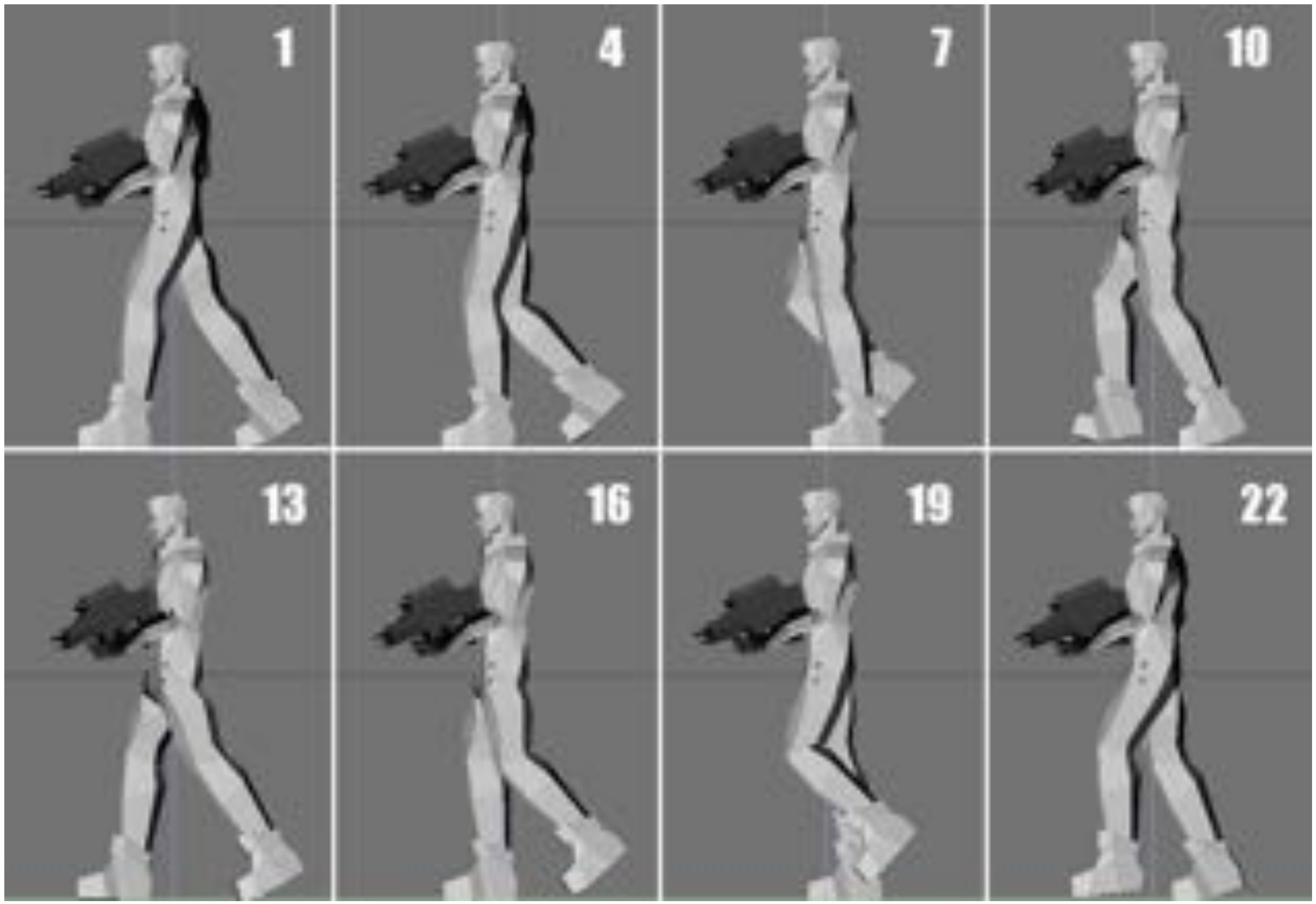
KEYFRAME ANIMATION

Copyright Preston Blair

Vertex tweening

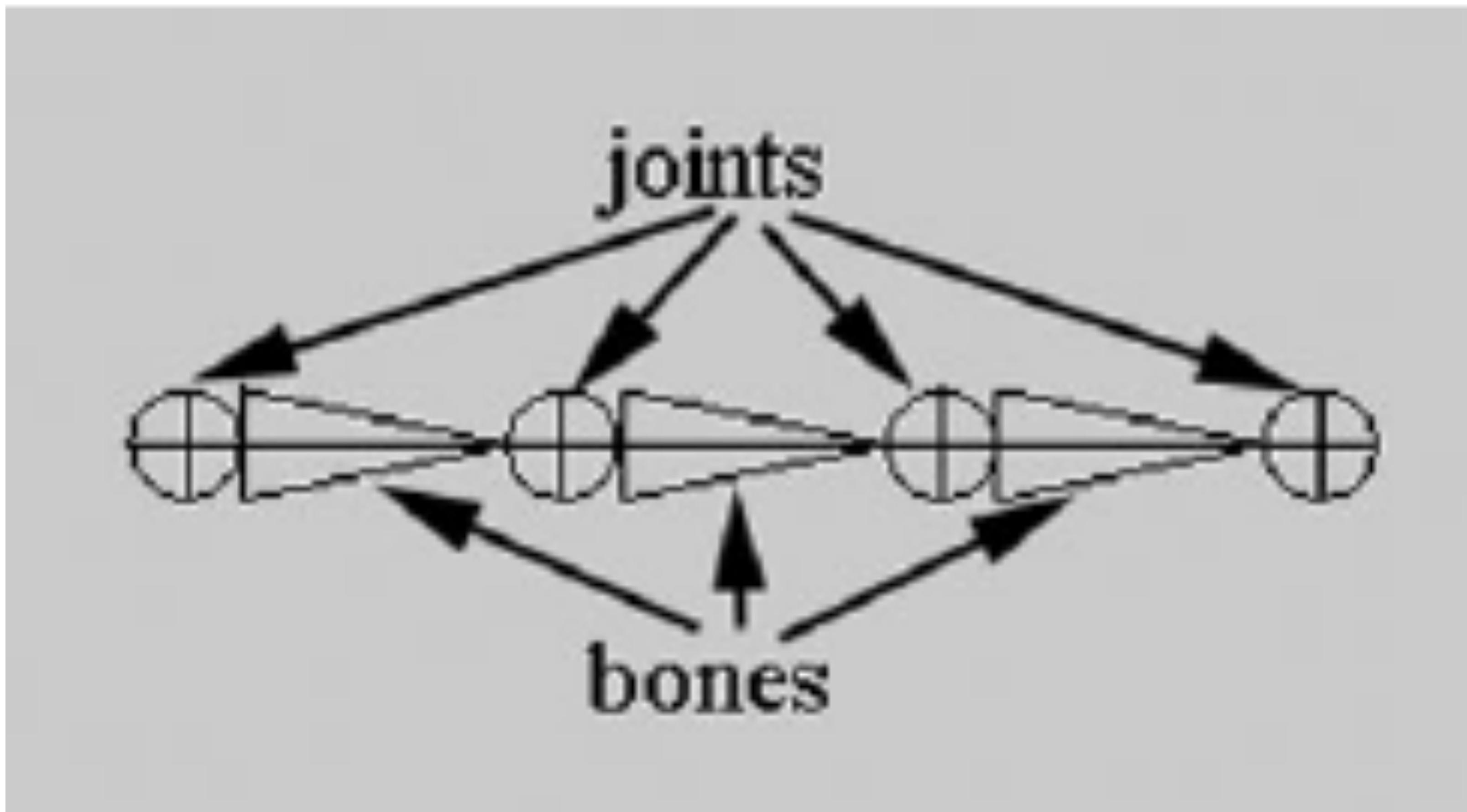


Vertex inbetweening



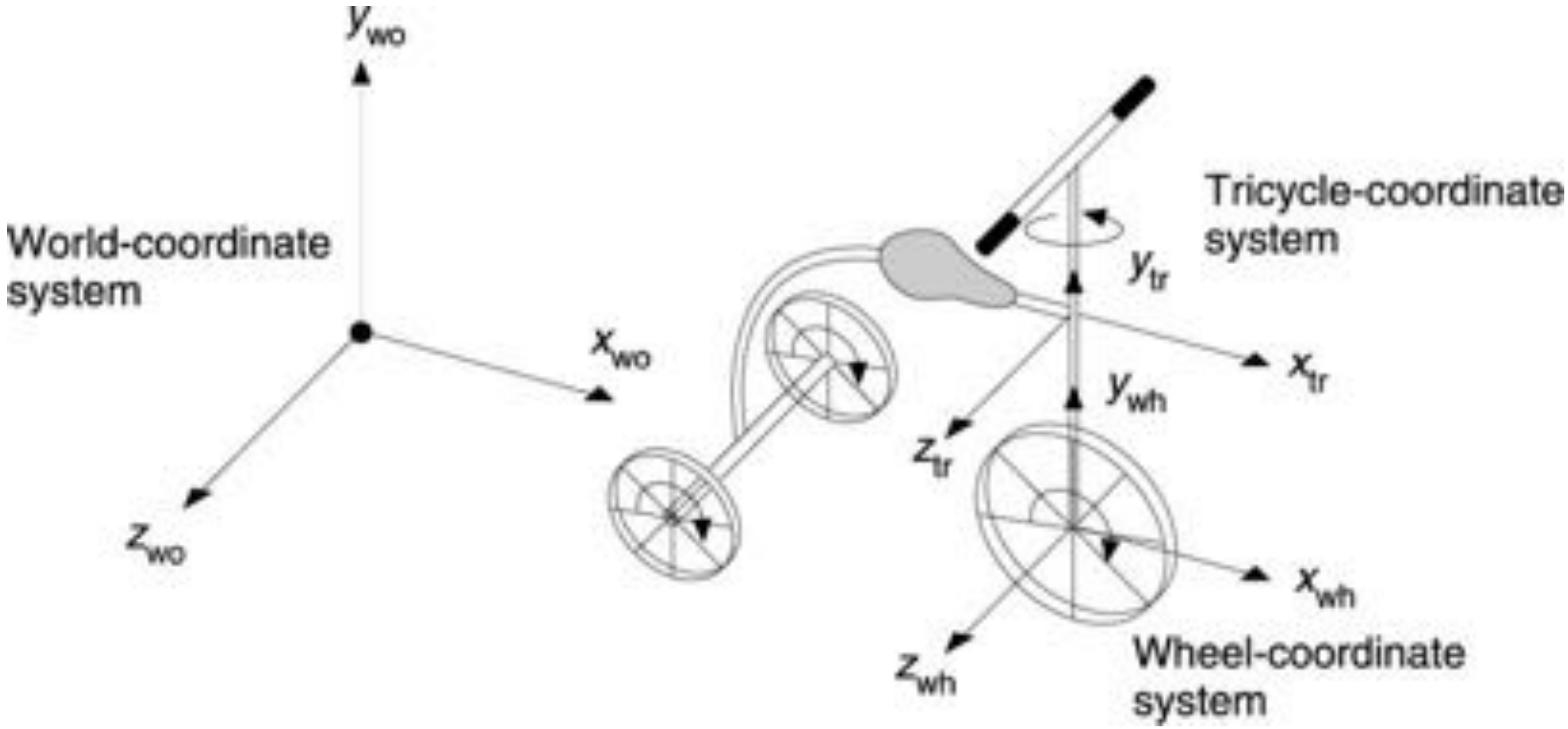
Rigging and skinning

Rigging and skinning

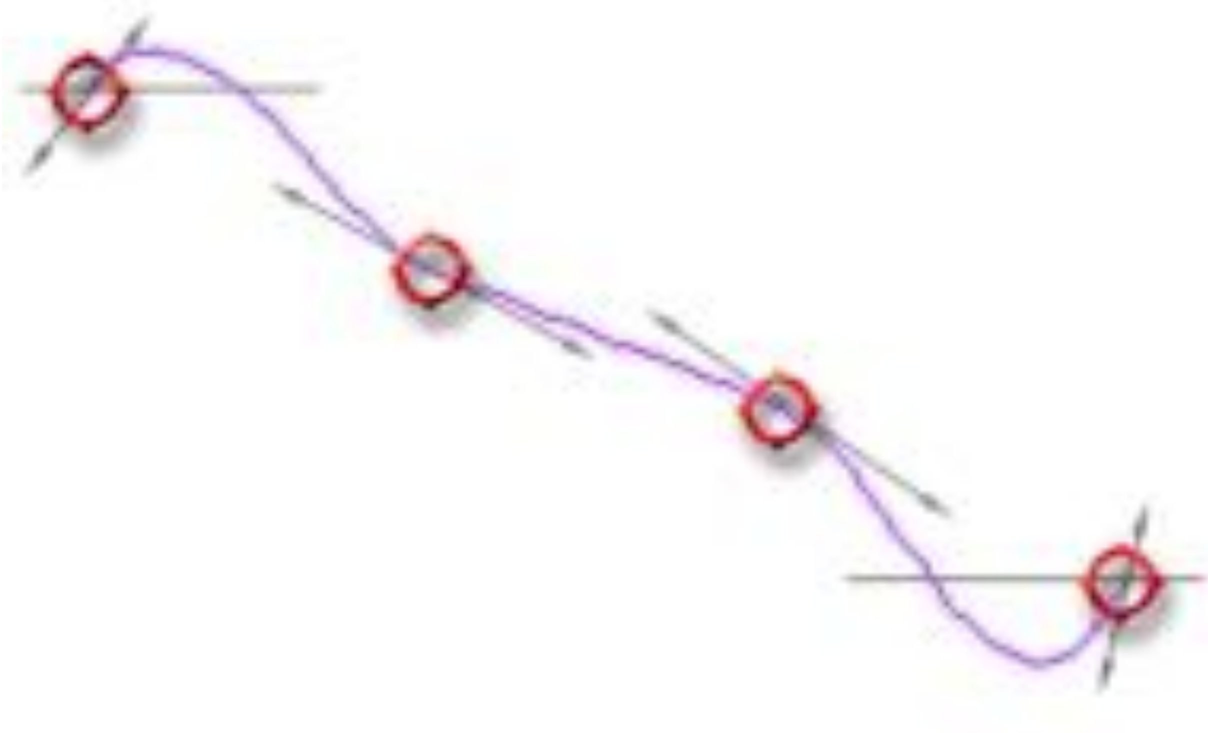


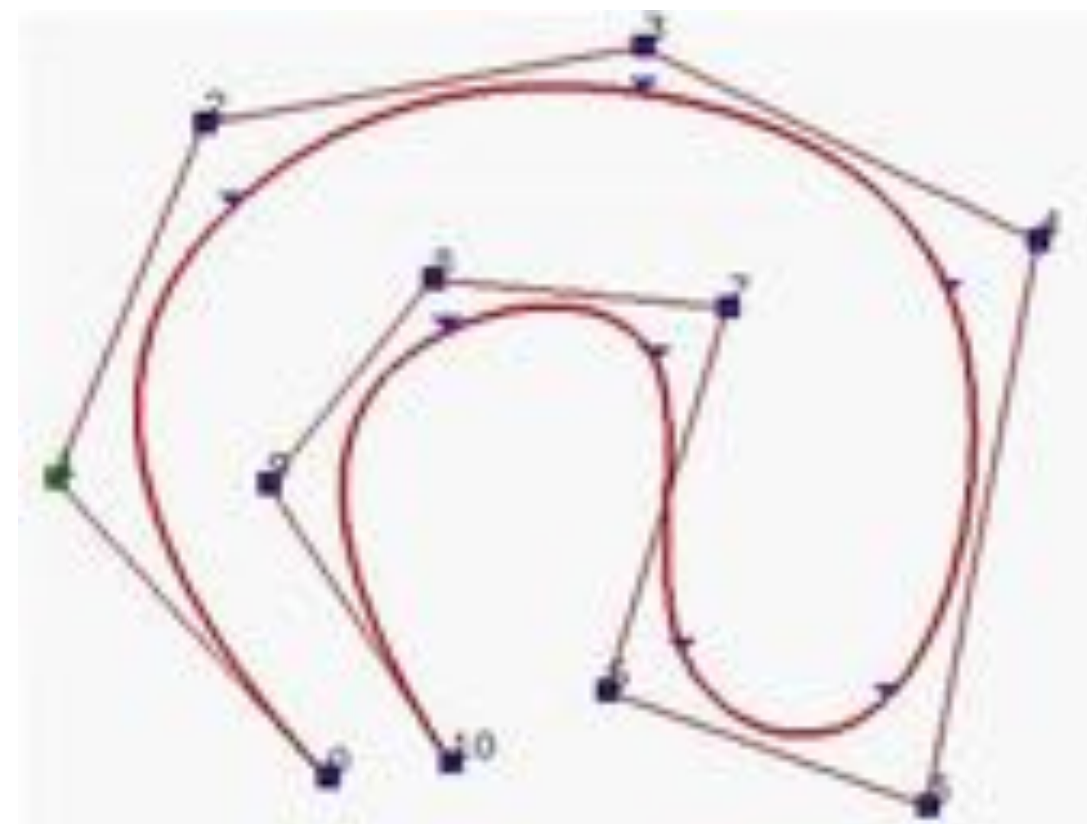
Keyframe animation

Keyframe animation



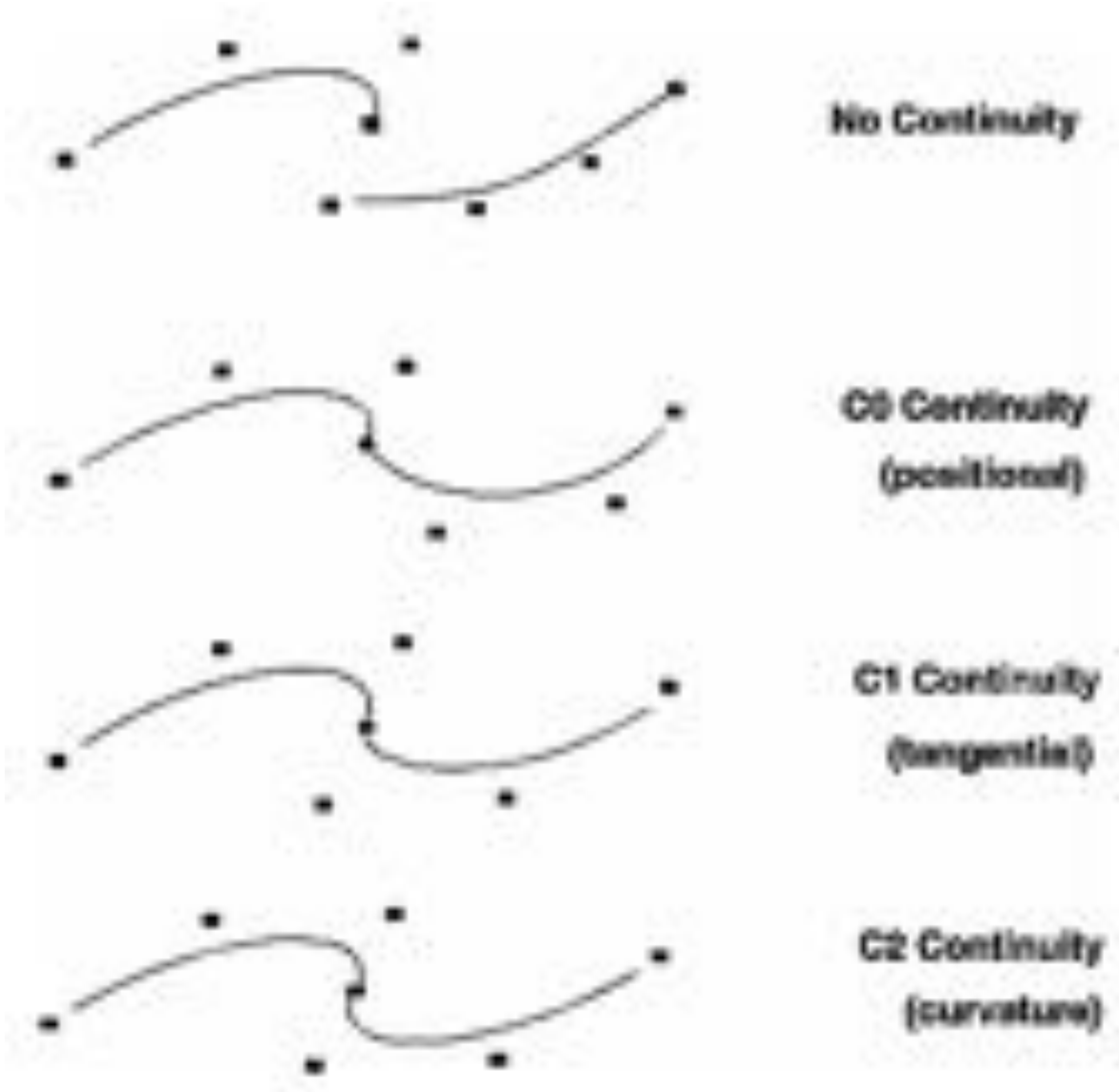
Keyframe animation



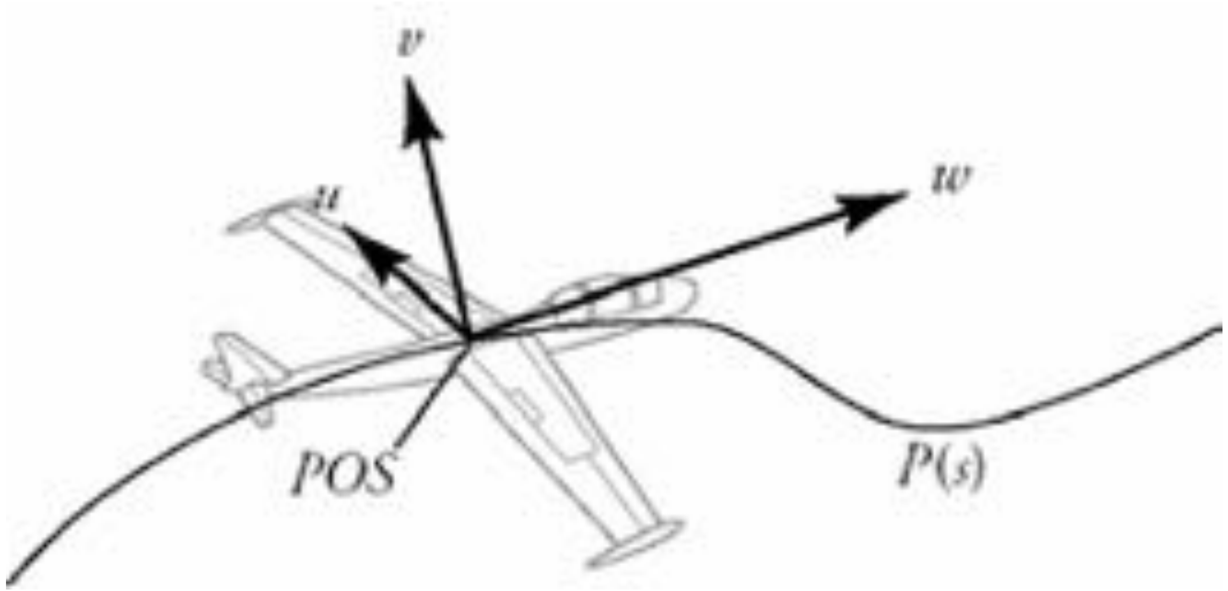


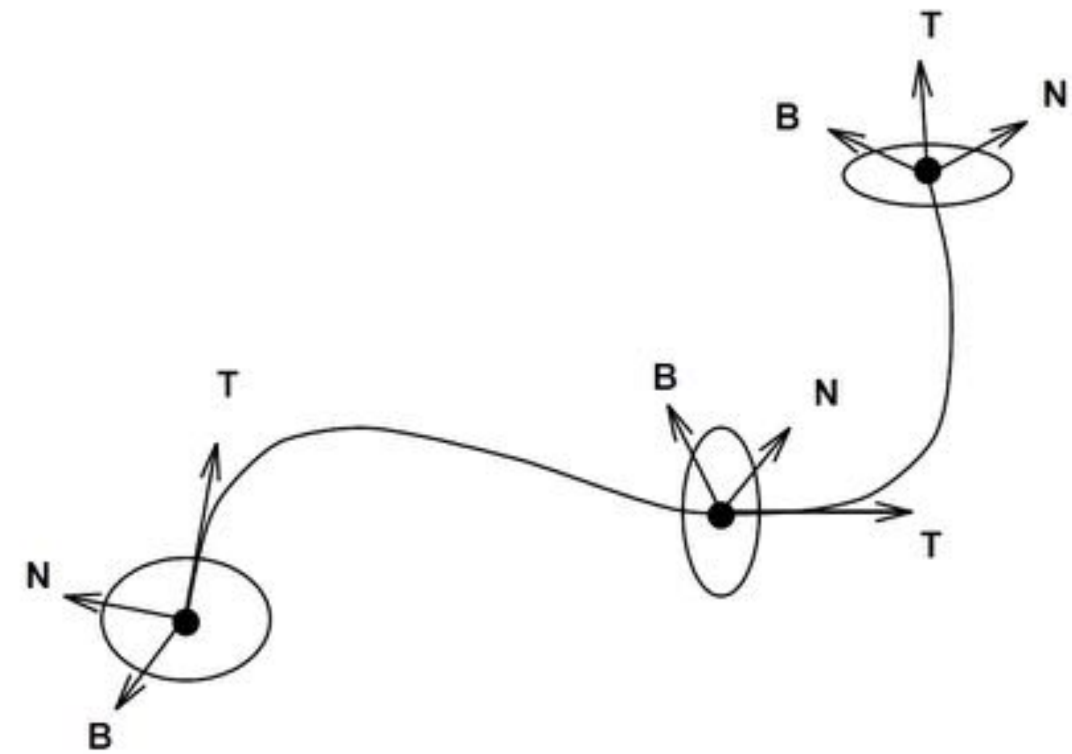
- Interpolation or approximating splines

Keyframe animation



Keyframe animation

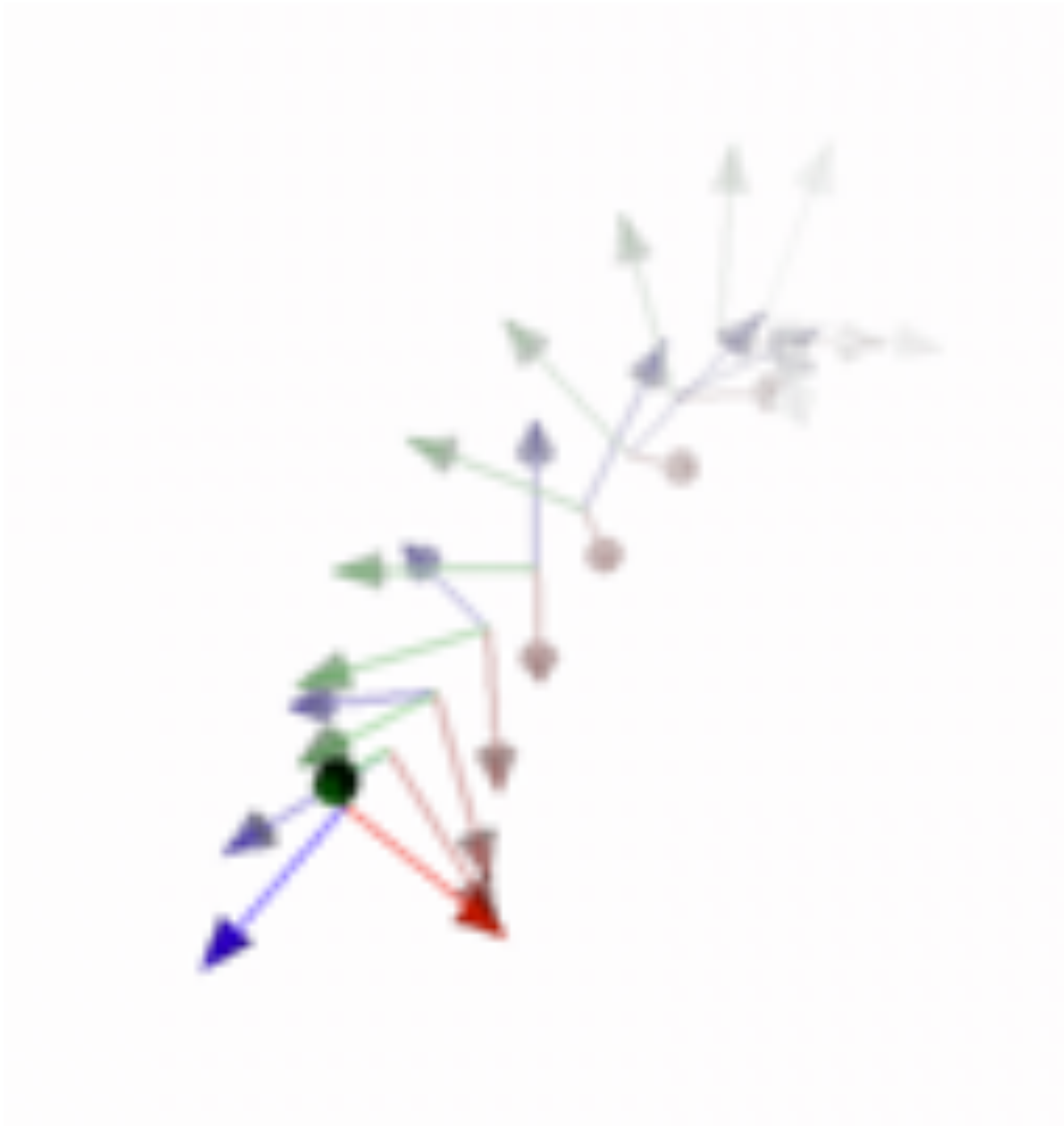




$$
\begin{aligned}
\vec{\mathbf{T}}(s) &= \frac{\vec{\mathbf{x}}'(s)}{\|\vec{\mathbf{x}}'(s)\|} \\
\vec{\mathbf{B}}(s) &= \frac{\vec{\mathbf{x}}'(s) \times \vec{\mathbf{x}}''(s)}{\|\vec{\mathbf{x}}'(s) \times \vec{\mathbf{x}}''(s)\|} \\
\vec{\mathbf{N}}(s) &= \vec{\mathbf{B}}(s) \times \vec{\mathbf{T}}(s) \ .
\end{aligned}
$$

Keyframe animation

Keyframe animation



Rotations and orientations

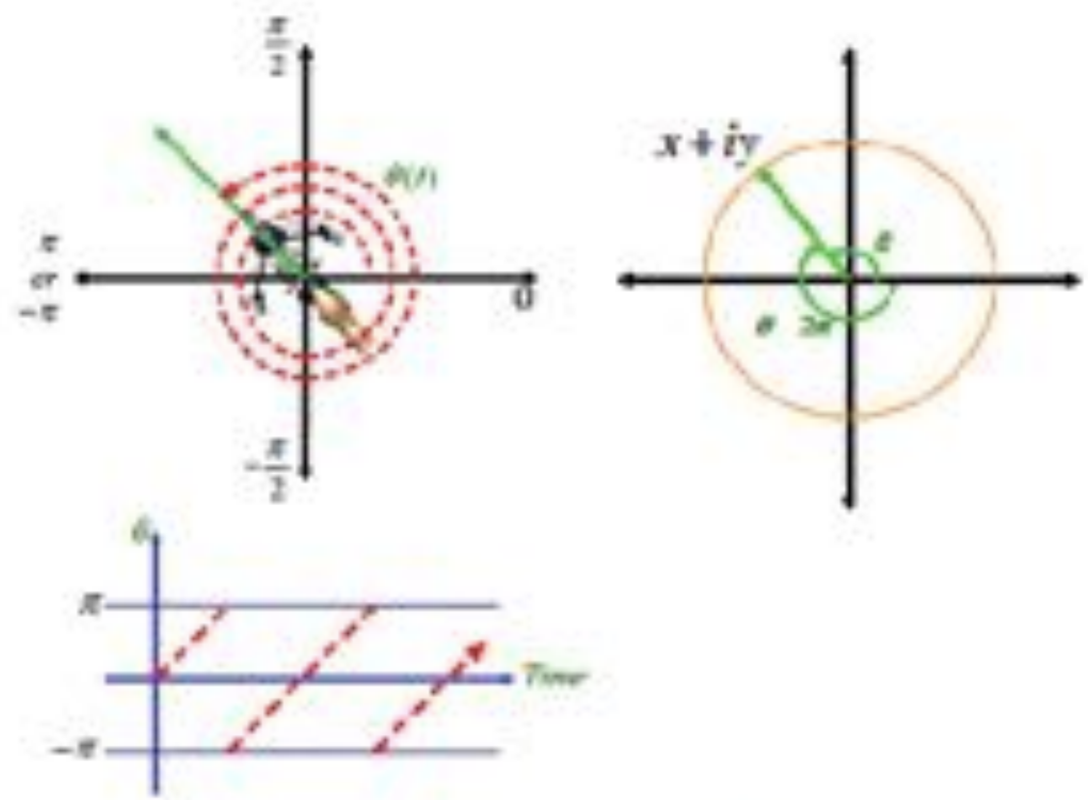


Figure 3: Rotations and orientations in two-dimensional space. The orientation of a two-dimensional rigid object can be represented by either a scalar angle from a reference orientation or a complex number of unit length. (Left and Bottom) By representing orientations by scalar angles, the orientation of a continuously moving rigid object could be mapped to a time-varying function with discontinuous jumps at the boundary of the 2π-interval. (Right) The use of complex numbers can circumvent this singularity problem because any continuous motion can be mapped to a continuous complex function. However, complex numbers may not be appropriate for parameterizing rotations because many different angular motions (such as counter-clockwise rotation by angle θ and clockwise rotation by angle $\theta - 2\pi$) are mapped to the same complex number.

Source: Jehee Lee, Representing rotations and orientations in geometric computing, CGA 2008

Rotations and orientations

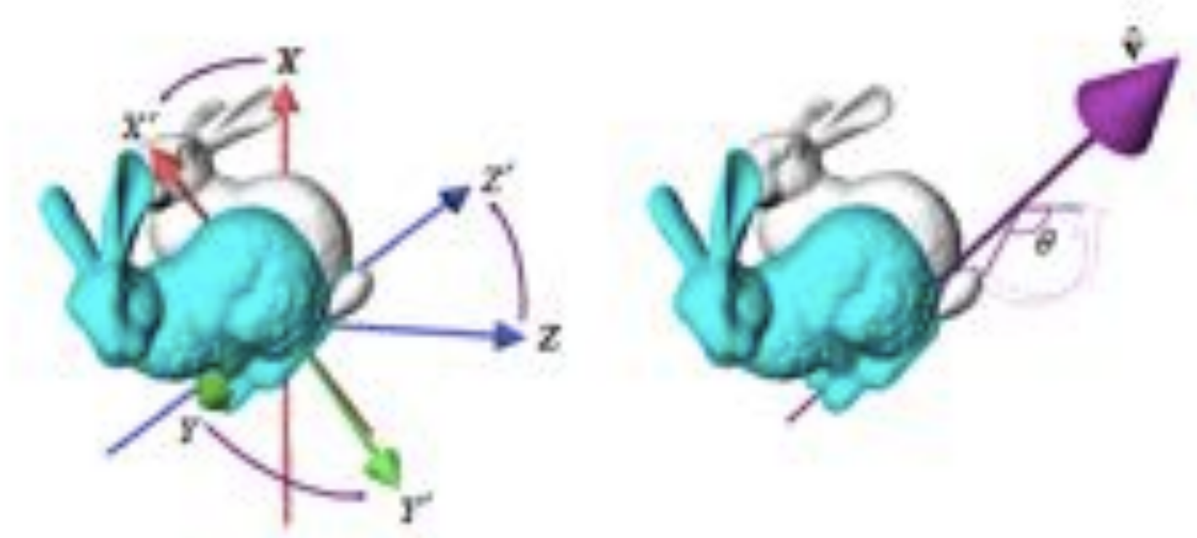


Figure 2: The reference coordinate system is depicted by the white bunny and XYZ axes. (Left) The orientation of the blue bunny can be described by a unit quaternion with respect to the reference coordinate system. $X'Y'Z'$ axes are a local coordinate system of the blue bunny. (Right) Euler's rotation theorem states that a fixed axis $\hat{\mathbf{v}}$ and an angle θ can always be found such that the rotation of the white bunny about axis $\hat{\mathbf{v}}$ by angle θ will bring it to the orientation of the blue bunny.

Source: Jehee Lee, Representing rotations and orientations in geometric computing, CGA 2008

Rotations and orientations

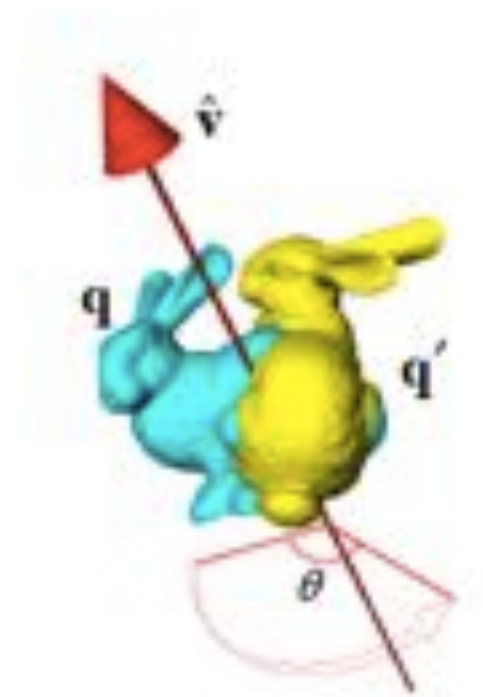


Figure 5: The relative rotation between any two orientations $\mathbf{q}$ and $\mathbf{q}'$ can be described as a vector $\mathbf{v} \in \mathbb{R}^3$ that specifies a rotation around a fixed axis $\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$ by the amount of angle $\theta = \|\mathbf{v}\|$.

Source: Jehee Lee, Representing rotations and orientations in geometric computing, CGA 2008

Rotations and orientations

Though the use of unit quaternions provides us with a useful non-singular parametrization for orientations, it cannot parameterize rotations without ambiguity. This can be shown as follows: Consider a family of rotations about axis $\hat{\mathbf{v}}$ by angle $\theta \pm 2n\pi$ for all integer n. All of these different rotational actions are mapped to a single unit quaternion or its antipode such that $\exp(0, \frac{\theta}{2}\hat{\mathbf{v}}) = \exp(0, \frac{\theta \pm 2n\pi}{2}\hat{\mathbf{v}})$ or $-\exp(0, \frac{\theta \pm 2n\pi}{2}\hat{\mathbf{v}})$. Note that both represent the same rotation. From these observations, we can conclude that we have to use vectors for parameterizing rotations and unit quaternions for orientations in three-dimensional space.

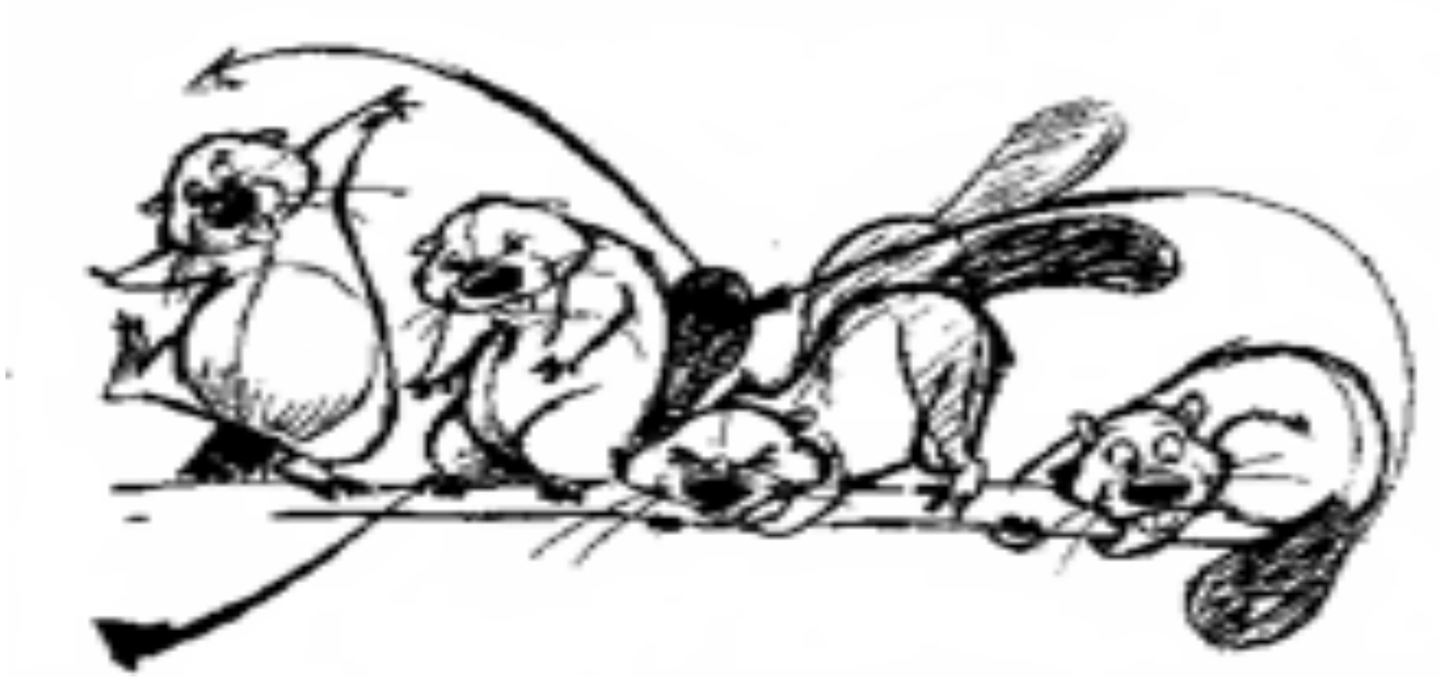
Source: Jehee Lee, Representing rotations and orientations in geometric computing, CGA 2008

Principle 2 - Arcs

7. Arcs

Principle 2 - Arcs

The visual path of action from one extreme to another is always described by an *arc*. Arcs in nature are the most economical routes by which a form can move from one position to another. In animation, such arcs are used extensively, for they make animation much smoother and less stiff than a straight line for the path of action. In certain cases, an arc may resolve itself into a straight path, as for a falling object, but usually, even in a straight line action, the object rotates. [12]



Paper 2. Representing Animations by Principal Components (2000)

EUROGRAPHICS 2000 / M. Gross and F.R.A. Hopgood
(Guest Editors)

Volume 19, (2000), Number 3

Representing Animations by Principal Components

Marc Alexa and Wolfgang Müller

Department of Computer Science, Darmstadt University of Technology, Darmstadt, Germany

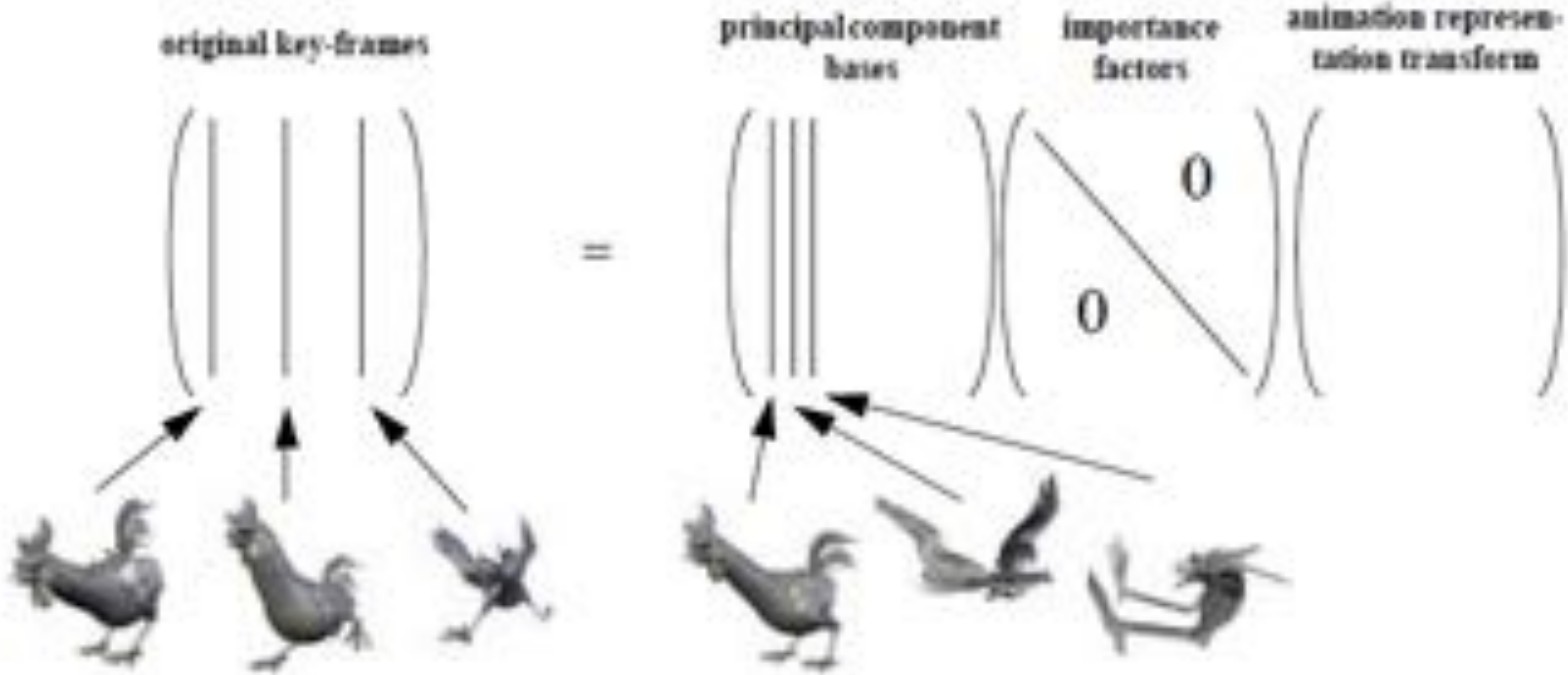

original key-frames
=
principal component bases
importance factors
0
0
animation representation transform

Figure 3: *Principal Component Analysis applied to the chicken animation. Prior to the SVD, base shapes are normalized. The sequences above show frames 0, 80, 160, 240, 320, and 400 using different number of bases in the reconstruction.*

Figure 4: *Exchanging geometry and in existing animations. a) A facial animation defined by a linear combination of base shapes. b) A featured-guided morph between the original avatar mesh and a new mesh. Topological merging is used to produce a mesh which represents both shapes. Feature control assures that the same vertices represent common features (e.g. mouth, eyes, etc.) c) The new mesh can be used with the existing animation with no additional user intervention. d) The morph can even be applied while the animation is performed.*

Paper 2. Representing Animations by Principal Components (2000)

1) Description
2) Clarity of Exposition
3) Quality of References
4) Reproducibility
5) Strengths and weaknesses
6) Rating (1-5)

Paper 2. Representing Animations by Principal Components (2000)

1) Problem statement
2) Scientific contributions
3) Experimental validation
4) Limitations
5) Impact