Video motion analysis for the synthesis of dynamic cues and Futurist art

J. P. Collomosse [a] P. M. Hall [a]

[a] *Department of Computer Science*
University of Bath
Claverton Down
Bath, England

Abstract

This paper presents new methods for stylising video to produce cartoon motion emphasis cues and modern art. Specifically, we introduce "dynamic cues" as a class of motion emphasis cue, encompassing traditional animation techniques such as anticipation and motion exaggeration. We describe methods for automatically synthesising such cues within video premised upon the recovery of articulated figures, and the subsequent manipulation of the recovered pose trajectories. Additionally, we show how our motion emphasis framework may be applied to emulate artwork in the Futurist style, popularised by Duchamp.

Key words: Non-photorealistic Rendering, Motion stylisation, Futurism

1 Introduction

The paper addresses the problem of stylising real-world video sequences to create animations. This problem comprises two principal technical challenges. First, how to generate stable artistic stylisations over the video (for example, an oil painterly effect)? Second, how to emulate the *motion emphasis cues* used by traditional animators? Early attempts to solve the first problem suffered from a distracting flickering [1,2] that more recent approaches suppress [3,4]. This paper focuses on the second problem of motion emphasis which, until recently, has received little attention in the non-photorealistic rendering (NPR)

Email addresses: jpc@cs.bath.ac.uk (J. P. Collomosse), pmh@cs.bath.ac.uk (P. M. Hall).

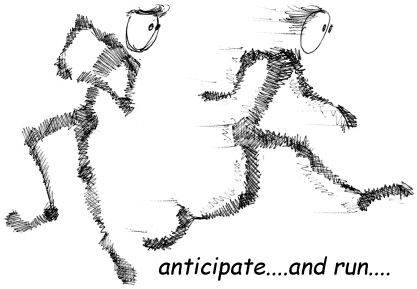

Fig. 1. Anticipation is a common dynamic cue; ghosting and streak-lines are also shown, as is some deformation.

literature. A limited range of motion emphasis effects have been produced from three dimensional computer graphics models [5,6], by motion capturing cartoons [7], or interactively from drawings [8] and video [9]; see [10] for a wider review. Of greatest relevance to this paper is previous work by the authors addressing the production of both *augmentation cues* and *deformation cues* in real video [11]. The contribution of this paper is to extend the analytic framework required for augmentation and deformation cues so that *dynamic cues* can be automatically produced. Furthermore the *Futurist* school of painting, typified by Duchamp, can be emulated; this too is a unique contribution to NPR.

Traditional animators emphasise motion with a variety of cues that are familiar to anyone who has watched animations. Streak-lines depicting the paths of objects, and ghosting effects that echo trailing edges, are both examples of what we call *augmentation cues*: the animation is visually augmented with marks of some kind. Animated objects may stretch as they accelerate, squash as they slow down, or bend to show drag or inertia — we call these *deformation cues*. Furthermore objects may "anticipate" movement by a slight prior movement backwards, or move in a characteristic way that exaggerates ordinary motion. These latter cues we call *dynamic cues*. Examples of these cues are illustrated in Figure 1. A deeper understanding of the differences between them relies on a definition of pose trajectory, as we now explain.

At any given instant in time an object has a particular *pose*, typically specified by a vector of numbers (for example, inter-joint orientations and world position). As this pose vector changes in time we obtain a *pose trajectory*. Augmentation cues and deformation cues are rendered as a function of pose trajectory. Dynamic cues differ because they alter the pose trajectory. This makes rendering dynamic cues very difficult because both the pose and timing of the object may change: poor rendering could leave "gaps" in the video, for example. Furthermore generating dynamic cues is non-trivial: a cartoon character can "wind up to run" in a way that is unique to them. The essential simplicities that bind the set of dynamic cues are very difficult to find.

Our purpose here is to provide an initial in-road into an understanding of dynamic cues. To this end we show how to generate and analyse a pose trajectory to produce:

- anticipation effects;
- motion "caricaturing" e.g. exaggeration effects;
- Futurist-like stills, in a style reflecting that of Duchamp.

Our broad approach is to track polygons fitted around rigid objects so as to estimate their pose trajectory. This is analysed to construct a hierarchical articulated figure of rigid parts, with its pose trajectory (Section 2). The dynamic cues we produce from this (Section 3) integrate fully with our early published framework for synthesising augmentation and deformation cues [11]. Further, all motion emphasis cues integrate with our stable video stylisation technique [3]. Therefore, the contribution of this paper completes our work in the automated production of animations from real-world video, see [10] for a full description of our *Video Paintbox*.

2 Recovering Articulated Structure

Our problem is to recover the motion of a articulated figure — a *doll* — from monocular video. The doll is to be built from rigid parts and have a hierarchical structure. The hierarchy is a tree in which each part corresponds to a tree node. Two nodes are linked in the tree if they are physically connected by a pivot.

Humans are an important class of articulated figures, and the recovery of human motion from video sequences is a well researched problem, see Hicks for a review [12]. Briefly, most techniques use a constraint in the form of many cameras or a prior model of human motion, neither option is open to us for we have one camera and cannot guarantee that a human is the articulated figure. The constraint we use is that the object moves in a plane (more formally: the motion vectors can be sufficiently well represented by a two-dimensional vector space).

The underlying idea is to consider pairs of rigid parts and observe the motion of one relative to the other. This allows us to estimate the centre of rotation, if it exists, at an instant in time. By holding fixed first one object and then the other we estimate two centres of rotation. If these are sufficiently close and both lie within the intersection of the polygons associated with the rigid parts, then we decide that the two objects are pivoted and select the rotation centre computed when the parent was held still as the pivot point. The root is arbitrarily assigned, its parent is the world frame. A depth ordering between

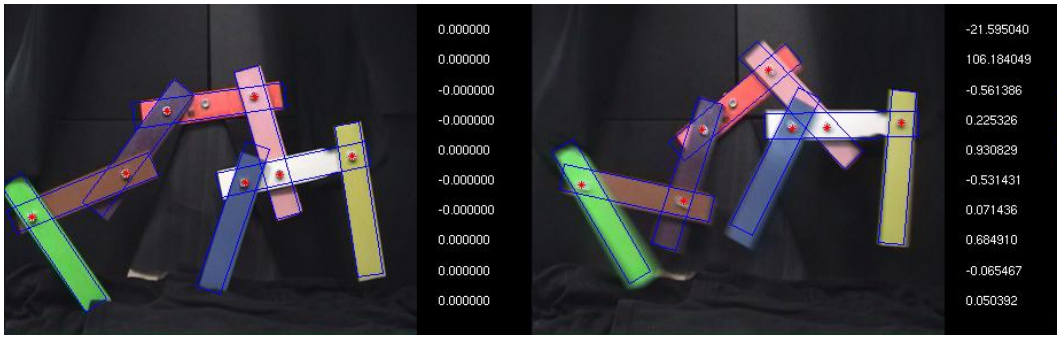

Fig. 2. An articulated contraption used for experiments. The full pose vector at various instants in time is shown alongside.

the parts of the figure is assigned using occlusion information available from the video, useful when later compositing features. The tracking and depth recovery processes are beyond the scope of this paper and the reader is referred elsewhere for details [13].

Our focus here is to recover the pose trajectory, $\mathbf{p}(t)$ of an articulated object:

$$\mathbf{p}(t) = \begin{bmatrix} \mathbf{c}(t) \\ \theta_1(t) \\ \dots \\ \theta_n(t) \end{bmatrix} \tag{1}$$

where $\mathbf{c}(t)$ is the location of some identifiable point on the object's root node, and the $\theta_i(t)$ specify the orientation of each branch node relative to its parent; $\theta_1(t)$ orients the whole articulated object using $\mathbf{c}(t)$ as a pivot.

We begin by tracking points on polygons. The *state* of a particle (point) at any time instant, t, is a vector comprising position, $\mathbf{x}(t)$, velocity $\mathbf{v}(t)$ and acceleration $\mathbf{a}(t)$.

$$\mathbf{s}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \\ \mathbf{a}(t) \end{bmatrix} \tag{2}$$

Each state is a particle in *state space*. This state is used by the Kalman filter [14] to track objects in video. The reader is referred elsewhere [10] for full details of tracking, which are beyond the scope of this paper — in brief, tracking operates as follows. Users identify objects in the video by drawing contours which are "shrink wrapped" to objects in the first frame of video

using snake relaxation [15]. We assume contour motion may be modelled by a linear conformal affine transform in the image plane, allowing planar motion plus scaling of objects. An estimate of this inter-frame transformation is estimated using a RANSAC search of putative feature correspondences within the tracked objects, obtained using a Harris corner detector.

Given the state of particles on a rigid body (polygon) we estimate the translation and rotation of the body in the following manner. At some time t let $\mathbf{x}_i$ be the ith identifiable point of a rigid body. Given three such points these transform, under an instantaneous rotation $\mathbf{R}$ and translate under an instantaneous displacement $\mathbf{u}$. In homogeneous coordinates:

$$\begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{u} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x} \\ 1 & 1 & 1 \end{bmatrix} \tag{3}$$

Each matrix is (3×3) so the unknown transform is straightforward to compute

$$\begin{bmatrix} \mathbf{R} & \mathbf{u} \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \\ 1 & 1 & 1 \end{bmatrix} \tag{4}$$

Hence we can compute instantaneous changes in orientation and location. It is a simple matter to integrate these to obtain a change relative to the starting orientation to acquire $[\mathbf{c}(t), \theta(t)]^T$, relative to the starting position.

We next consider whether a given pair of rigid objects are pivoted. Given two rigid objects, A and B, we assume the pose trajectory for each of them, $\mathbf{p}_A(t)$ and $\mathbf{p}_B(t)$. Consequently the motion of B relative to A is easy to estimate, being characterised completely by the difference in pose trajectories $\mathbf{p}_B(t) - \mathbf{p}_A(t)$. Therefore we can observe the movement of B in the reference frame of A, which reduces the problem of finding a mutual pivot to one of finding a fixed point about which B rotates (if it rotates at all).

Let $\mathbf{x}_i$ be a point on B, measured in the fixed reference frame of A. Suppose B rotates about the fixed point $\mathbf{f}$, relative to A. If motion is uniform, then after a short time interval dt this point appears at $\mathbf{y}_i$

$$\mathbf{y}_i = \mathbf{R}(\mathbf{x}_i - \mathbf{f}) + \mathbf{f} \tag{5}$$

The problem is to estimate $\mathbf{f}$ given a sufficient number of $\mathbf{x}_i$ and $\mathbf{y}_i$. This problem differs Equation (3) because there rotation about the origin was sufficient, and we computed a translation too; here we seek rotation about an

unknown point. We will later discuss the relationship between these two problems in greater depth. The important principle here is $\mathbf{f}$ is a singularity of the transform, therefore we cannot invert the system of equations.

We proceed by solving a system of homogeneous linear equations. Writing x_j for the j^{th} element of some point $\mathbf{x}$, at time t and y_j for the corresponding element at time $t + dt$. Equation (5) becomes

$$y_1 = r_{11}x_1 - r_{12}x_2 + u_1 \tag{6}$$
$$y_2 = r_{21}x_1 - r_{22}x_2 + u_2 \tag{7}$$

in which

$$\mathbf{u} = (\mathbf{I} - \mathbf{R})\mathbf{f} \tag{8}$$

We can now write

$$\begin{bmatrix} x_1 & -x_2 & 0 & 0 & 1 & 0 & -y_1 \\ 0 & 0 & x_1 & -x_2 & 0 & 1 & -y_2 \end{bmatrix} \begin{bmatrix} r_{11} \\ r_{12} \\ r_{21} \\ r_{22} \\ u_1 \\ u_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{9}$$

We may extend the left-most matrix because each identifiable point in B provides two rows, yielding a design matrix $\mathbf{M}$. The smallest right-singular vector of $\mathbf{M}$ is a suitable solution in the least squared sense. This is normalised so that its seventh element is unity and in this way we obtain the rotation matrix elements r_{kl} and a displacement $\mathbf{u}$. The pivot $\mathbf{f}$ is obtained from Equation (8) as

$$\mathbf{f} = (\mathbf{I} - \mathbf{R})^{-1}\mathbf{u} \tag{10}$$

Because this estimate of $\mathbf{f}$ is obtained using all identifiable points of B it tends to be robust to measurement error. If there is no rotation, then $\mathbf{R} = \mathbf{I}$, indicating there is no pivot. We decide that B has a pivot relative to A only if a pivot $\mathbf{f}$ exists that lies within the intersection of A and B.

To further improve robustness we reverse the roles of A and B, recomputing the pivot point. Furthermore, we compute the pivot for all time instants t,

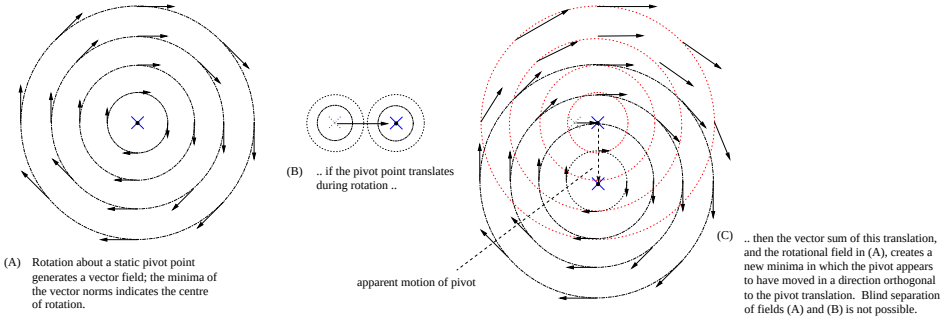

Fig. 3. Illustrating the difficulty of recovering pivot point location from rotation, in the case of a moving pivot. Under instantaneous motion, the combination of rotation and pivot shift causes an apparent translation of pivot location orthogonal to the direction in which the pivot have moved.

each over a fixed interval dt. We insist that the pivot remains within the intersection of A and B over all time. Figure 2 illustrates the fact that we can recover complex articulated structures in this way.

We now return to the relationship between Equations (3) and (5). The first of these computes rotation about the origin and an accompanying displacement, the second computes rotation about an unknown fixed pivot. We claim *it is not possible to simultaneously compute a rotation, a pivot and a displacement.* As proof we consider the point $\mathbf{x}$ rotating about the origin with constant angular velocity ω. The tangential velocity of this point is $\dot{\mathbf{x}} = \omega(\mathbf{x} \otimes \mathbf{n})$, where $\mathbf{n}$ is a normal to the plane of rotation and $\otimes$ is vector cross product (it is not necessary for this to obey the right-hand screw rule). Now suppose that $\mathbf{x}$ not only rotates about the origin but translates too, with a linear velocity $\dot{\mathbf{u}}$. The governing equation now is $\dot{\mathbf{x}} = \omega(\mathbf{x} \otimes \mathbf{n}) + \dot{\mathbf{u}}$. Since $\dot{\mathbf{u}}$ is a constant we can always write it in the form $\dot{\mathbf{u}} = \omega(\mathbf{d} \otimes \mathbf{n})$, and therefore obtain $\dot{\mathbf{x}} = \omega(\mathbf{x} \otimes \mathbf{n}) + \omega(\mathbf{d} \otimes \mathbf{n})$. Appealing to the fact that addition distributes over the cross product operator we obtain $\dot{\mathbf{x}} = \omega((\mathbf{x} + \mathbf{d}) \otimes \mathbf{n})$ from which we conclude that effective centre of rotation has been shifted as a consequence of the displacement, in a direction perpendicular to it. This result is analogous to the phenomenon observed in a gyroscope, which when suffering a force in the plane of its rotation moves, in the plane, in a direction orthogonal to the applied force. Here it shows that if we choose an arbitrary pivot we can always determine a compensating displacement, and vice-versa. Therefore we cannot unambiguously estimate both at once; this is an in-principle restriction.

Given a pose trajectory for each rigid body, and a pivot for each pair of linked rigid bodies, it is a matter of book-keeping to assemble a hierarchical articulated figure, complete with a full pose trajectory of the form in Equation (1); we have automatically assembled a doll from video data.

3 Dynamic cues and modern art

Given a recovered doll, we can produce not only dynamic cues as seen in traditional animations, but also emulate the Futurist style of modern art. So far as we are aware, both represent unique contributions.

As mentioned the general form of dynamic cues is to map one pose trajectory into another:

$$\mathbf{p}'(t) = \mathcal{F}[\mathbf{p}(t)] \tag{11}$$

The new pose trajectory is used to govern all other cues, so that objects can be augmented and deformed. Again as mentioned, a full understanding of dynamic cues eludes us at the present time, but we can make some progress by considering two important classes of dynamic cue: anticipation and motion exaggeration. We now consider each in turn, followed by a discussion on emulating Futurist art.

3.1 Motion Anticipation

Anticipation is an animation technique applied to objects as they begin to move; the technique is to create a brief motion in the opposite direction, which serves to emphasise the subsequent large scale movement of an object. The anticipation cue communicates to the audience what is *about to* happen. Anticipation acts upon a subject locally — only within a temporal window surrounding the beginning of the movement to be emphasised, and only upon the feature performing that movement.

We have implemented anticipation as a 1D signal filtering process. Each individual, time varying component of the pose vector $\mathbf{p}(t)$ (for example, the angle a metronome beater makes with its base) is fed through an "anticipation filter", which outputs an "anticipated" version of that pose signal. The filter also accepts six user parameters which control the behaviour of the anticipation motion cue. The filtering process operates in two stages. First, the 1D signal is scanned to identify the temporal windows over which anticipation should be applied. Second, the effect is applied to each of these windows independently.

3.1.1 Identifying Temporal Windows for Anticipation

Given a 1D input signal, the filter first identifies temporal windows for application of anticipation. These are characterised by the presence of high acceleration magnitudes (above a certain threshold), which exist for a significant

number of consecutive frames (a further threshold) in the signal. These two thresholds form part of the set of user parameters that control the effect. This process allows us to identify a set of temporal windows corresponding to large motion changes, which an animator would typically emphasise using anticipation. A high acceleration magnitude may or may not generate a change of direction in the signal and we have determined that the manifestation of the anticipation cue differs slightly between these two cases:

(1) First, consider the case where acceleration causes a change of direction in the 1D signal; for example, a pendulum at the turning point of its swing. Regardless of the acceleration magnitude of the pendulum beater (which may rise, remain constant, or even fall during such a transition), the anticipation effect is localised to the instant at which the beater changes direction i.e. the turning point of the signal; the *minimum of the magnitude of the first derivative with respect to time*. In the case of the *METRONOME* sequence (Figure 5), a brief swinging motion would be made, say to the left, just prior to the recoil of the metronome beater to the right. The object then gradually "catches up" with the spatio-temporal position of the original, un-anticipated object at a later instant.

(2) Now consider the second case where acceleration does not cause change of direction in the 1D signal; for example, a projectile already in motion, which acquires a sudden burst (or decrease) in thrust, i.e. a change in acceleration magnitude. The anticipation effect is manifested as a short lag just prior to this sudden acceleration change; i.e. at the *maximum in the magnitude of the third derivative with respect to time*. As with case 2, the projectile swiftly accelerates after anticipation to catch up with the spatio-temporal position of the original, unaffected projectile. Interestingly a projectile moving from rest is equally well modelled by either the first or second case, since the locations of zero speed (minimum first derivative) and maximum acceleration change (maximum third derivative) are coincident.

3.1.2 Synthesising Anticipation within the Temporal Window

Each temporal window identified for application of the anticipation cue is processed independently, and we now consider manipulation of one such window. The first task of the "anticipation filter" is to scan the pose signal to determine whether a change of direction occurs within the duration of the temporal window. This test determines which criterion from the respective case (1 or 2) is used to determine the instant at which anticipated motion should be "inserted" into the sequence; we denote this time instant by τ. We define a temporal "working interval" as the time window within which the pose is varied from the original signal, in order to introduce anticipation. This working

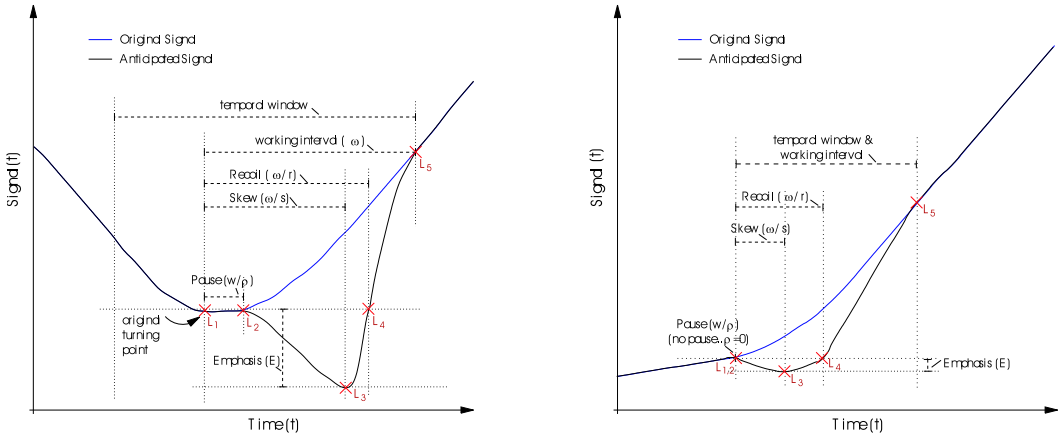

Fig. 4. Schematic examples of the anticipation filter under case one (signal direction of motion changes) and case two (signal direction of motion unaffected). Case two has been illustrated with pause parameter $\rho = 0$. Subsection 3.1.2 contains an explanation of the user parameters ρ, s, r, and ϵ which influence the behaviour of the effect.

interval extends from time τ to the end of the temporal window, which we write as $\tau + \omega$. In all cases the direction of the anticipatory motion will be in opposition to the direction in which acceleration acts. We refer the reader to Figure 4 to assist in the explanation of the subsequent signal manipulation.

We create the anticipation effect by modifying the 1D pose signal to follow a new curve, interpolating five landmark points $\mathbf{L}_{1..5.5}$ in space, interpolated using cubic Catmull-Rom spline functions. Aside from the two parameters used to control activation of the effect, there are four user parameters ρ, s, r, and ϵ (where $\rho \leq s \leq r$). These influence the location of the five landmark points $[\mathbf{L}_{1...5}]$, which in turn influences the behaviour of the anticipation. We now explain the positioning of each of the five landmarks and the effect the user parameters have on this process. Throughout, we use the notation $\theta(t)$ to indicate the original (unanticipated) 1D pose signal at time t (taken from pose vector $p(t)$), and $\theta'(t)$ to denote the new, anticipated signal.

- $\mathbf{L}_1$. The first landmark marks the point at which the original and anticipated pose signals become dissimilar, and so $\mathbf{L}_1 = (\tau, \theta(\tau))^T$. Recall τ is the determined by the algorithm of either case 1 or 2, as described in the previous subsection.
- $\mathbf{L}_2$. At the instant τ, a short pause may be introduced which "freezes" the pose. The duration of this pause is a fraction of the "working interval" — specifically ω/ρ frames, where ρ is a user parameter. The second landmark directly follows this pause, and so $\mathbf{L}_2 = (\tau + \omega/\rho, \theta(\tau))^T$.
- $\mathbf{L}_3$. Following the pause, the pose is sharply adjusted in the direction opposite to acceleration, to "anticipate" the impending motion. The magnitude (E), and so the emphasis of, this anticipatory action is proportional to the magnitude of acceleration: $E = \epsilon|\ddot{\theta}(\tau)|$. Here ϵ is a user parameter

(a constant of proportionality) which influences the magnitude of the effect. A further user parameter, s, specifies the instant at which the anticipation is "released" to allow the movement to spring back in its original direction. We term s the "skew" parameter, since can be used to skew the timing of the anticipation to produce a long draw back and quick release, or a sharp draw back and slow release. Referring to cartoonist Richard Williams' guidelines for anticipation [16], one would typically desire the former effect ($s > 0.5$), however our framework allows the animator to explore alternatives. The third landmark is thus located at the release point of this anticipated signal, and so $\mathbf{L}_3 = (s, E)^T$.

$\mathbf{L}_4$. The rate at which the feature springs back to "catch up" with the unanticipated motion is governed by the gradient between the third and fifth landmarks. This can be controlled by forcing the curve through a fourth landmark $\mathbf{L}_4 = (\tau + \omega/r, \theta(\tau))^T$.

$\mathbf{L}_5$. Finally the point at which the anticipated and original pose signals coincide is specified by the final landmark, $\mathbf{L}_5 = (\tau + \omega, \theta(\tau + \omega))^T$.

Figure 5 shows animation frames of a metronome anticipating motion by "snapping" [16]. The bending is due to our deformation effects acting on the modified pose trajectory and indicates inertia, which is why the beater bends as if to oppose motion.

3.2 Motion Caricaturing

Motion exaggeration is another form of dynamic cue. From an animators point of view, motion exaggeration characterises the way an object moves much as a newspaper caricaturist might exaggerate facial features or an impersonator exaggerates vocal idioms. Intuitively, these characteristics are outliers compared to a distribution of common cases.

This principle has been put to use to produce cartoon-like versions of a face [17], as follows. An eigenmodel is generated from mug-shots of many people by considering each image as a vector in some high-dimensional space. An individual mug-shot is projected into this eigenspace, scaled away from the mean, and then reconstructed to reveal a "cartoon". We might proceed by analogy, at least for cyclic motions such as a walk. The set of pose trajectory for a walking motion must lie on a annular manifold embedded within pose space (the space comprising all possible pose vectors). The eigenvectors of this trajectory point in the most important directions. We can scale a pose vector await from the mean, in proportion to the eigenvalues associated with the eigenvectors, thus scaled more along the important directions. This approach is poor: (1) The doll can contradict physical constraints, so that feet appear to slide along the floor or look as they *should* penetrate the ground, for ex-

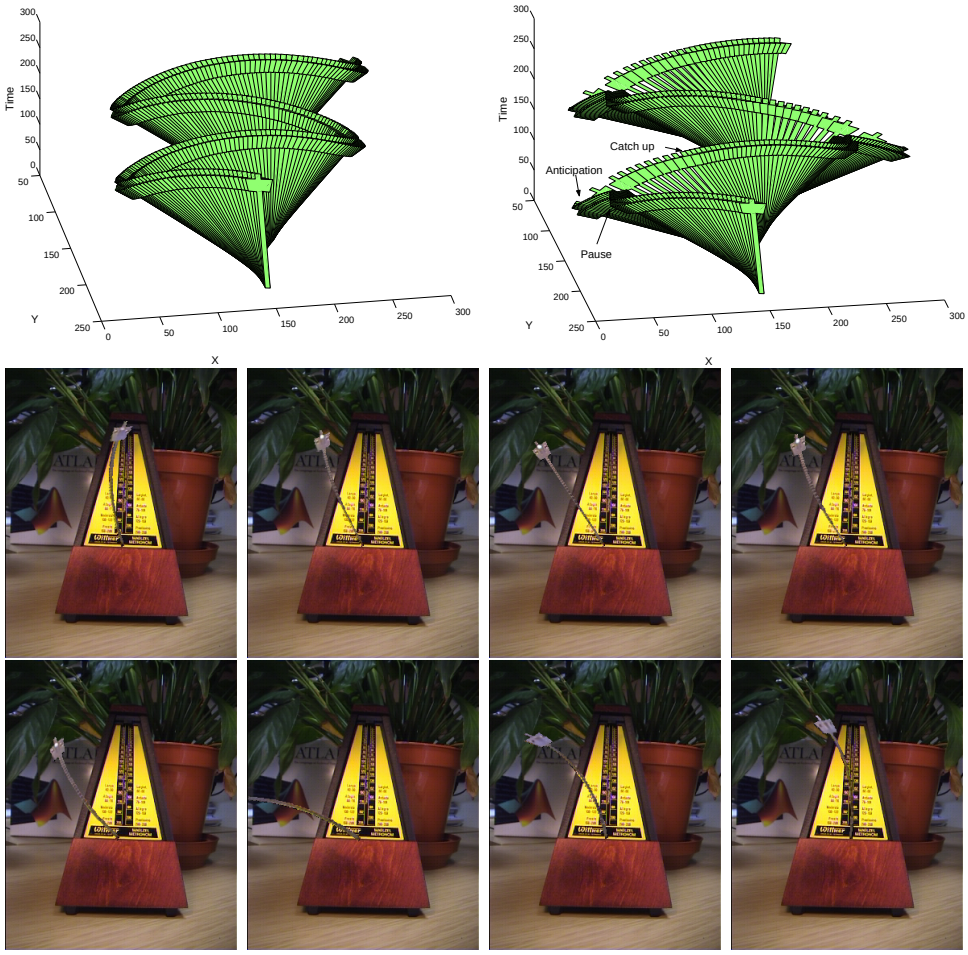

Fig. 5. Top: Time lapse representation of the beater in the *METRONOME* sequence, before (left) and after (right) application of the anticipation filter to the pose vectors. Bottom: Stills taken from a section of the rendered *METRONOME* sequence, exhibiting the anticipation cue combined with a deformation motion cue emphasising drag (described in [11]).

ample; (2) The output can be aesthetically displeasing — which is difficult to quantify but is important nonetheless; (3) It offers little scope for animator control, which is probably related to point 2. We have only indirect evidence to support this: animators produce output with an aesthetic value greater than any machine can manage at this point in time.

Our approach is to allow animators to impose physical constraints, so that feet are fixed to the ground when necessary, but that the remaining motion is exaggerated by scaling away from some mean. Consider a full pose trajectory $\mathbf{p}(t) \in \Re^n$. Animators are able to specify a subspace that remains can move between times τ_1 and τ_2 using a projection matrix $\mathbf{M}(t) \in \Re^{m \times n}$ that "picks out" those dimensions of the pose trajectory that *can* be changed at some time t. Thus

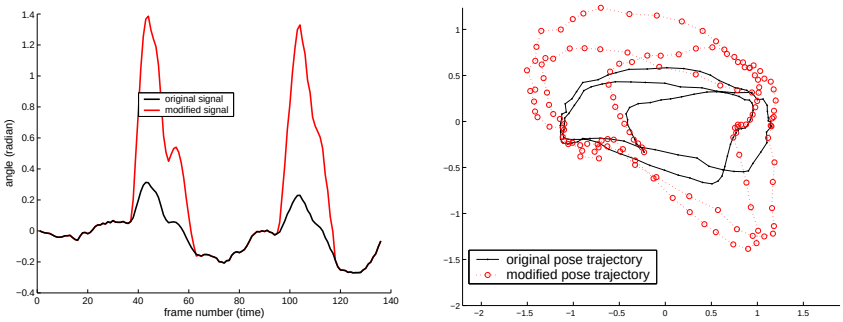

Fig. 6. Left: The change in the pose trajectory of the left upper leg. The original signal (black) is held fixed when then left foot is on the ground, but is altered (red) when the left foot is off the ground. Right: The pose complete trajectory projected into the plane defined by the largest two eigenvectors. The original pose(black) is clearly scaled subject to constraints — some points remain static — to yield an updated pose trajectory (red).

Fig. 7. A "Monty-Python" walk; the product of motion caricaturing applied to walking.

$$\mathbf{q}(t) = \mathbf{M}(t)\mathbf{p}(t) \tag{12}$$

identifies those elements of pose that can vary at time t. Typically each row of $\mathbf{M}$ is drawn from the n^2 identity matrix. We can now synthesise a new pose vector:

$$\mathbf{p}'(t) = \mathbf{p}(t) + \mathbf{M}^T(t)\mathbf{q}'(t) \tag{13}$$

where $\mathbf{q}'(t) = \mathcal{F}[\mathbf{q}(t)]$ is some modified version of the "variable" pose.

We have found that simply scaling away from the mean of the subspace yields better but nonetheless poor results. This is because scaling along eigenvectors tends to obscure those high-frequency characteristics a walk (say) as individual. Our approach is more subtle. We first transform the signal by $\mathbf{R}$ so that is principle eigenvector aligns with the 'x'-axis: $\mathbf{r}(t) = \mathbf{R}\mathbf{p}(t)$. Next we fit a piecewise curve $\mathbf{s}(t)$ smoothly approximate $\mathbf{r}(t)$. Then we measure the error signal $\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{s}(t)$. We then map as follows:

$$\mathbf{q}'(t) = \mathbf{M}^{-1}(w(t)\mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{e}(t)) \tag{14}$$

where $\mathbf{A}$, $\mathbf{B}$ are linear transforms and $w(t)$ is a smoothing function that ensures the scaling is zero at the edges of the time window $[\tau_1, \tau_2]$. Without this weighting the motion suffers a discontinuity at window boundaries. This approach has the advantage of separating high-frequency detail from low-frequency detail and the effect on a particular signal is shown in Figure 6.

We applied this mechanism to create an animated sequence, stills from which are shown in Figure 7. The pith-helmet and handle-bar moustache were painted using techniques described elsewhere [10]. A commercially available product added the 1920's cinematography effects.

3.3 Simulating Futurist Artwork

Our attempts at synthesising Futurist art from video are unique in the NPR literature, although early attempts at generating Futurist-like effect via superimposition and distortion of 3D models are briefly described in [18]. The closest alternative is the automatic production of Cubist art from three or four photographs [19]. The futurists were a group of artists working in the early part of the 20th century, Marcel Duchamp is perhaps their best known member. Futurist Art and Cubist Art share a number of visual characteristics, however the difference of relevance here is that the Cubist's depicted motion unstructured in time (an object is "here" at some unspecified time) whereas the Futurists depict structured motion (the object is "here and now").

We began by studying Duchamp's "Nude Descending the Stairs", which he painted in response to the work of motion scientist Étienne-Jules Marey [20]. "Nude Descending the Stairs" is a complicated piece of Art, a plethora of arms and legs intertwine and obscure one another; motion blurring, ghosting, streaklines, and other artifacts usually associated with animation are also present into the painting. Duchamp succeeds in creating a sense of motion without ever painting a single form that can be recognised as definitively human.

We discovered that careful analysis of pose trajectory is the key to synthetic Futurist art. More specifically, the motion of the feet can be used to control the whole process. This suggests Duchamp intuitively picked out very particular phases in the walking cycle as being salient. We now explains which particular phases Duchamp seemed to be interested in.

The angle that a foot makes to the lower-leg is, to a first approximation, sinusoidal over one cycle. The cycles of the feet are in anti-phase with respect to one another. Duchamp appears to use a particular 1/4 cycle of the nearest foot to "cue in" motion blurring, and the corresponding 1/4 cycle of the rear

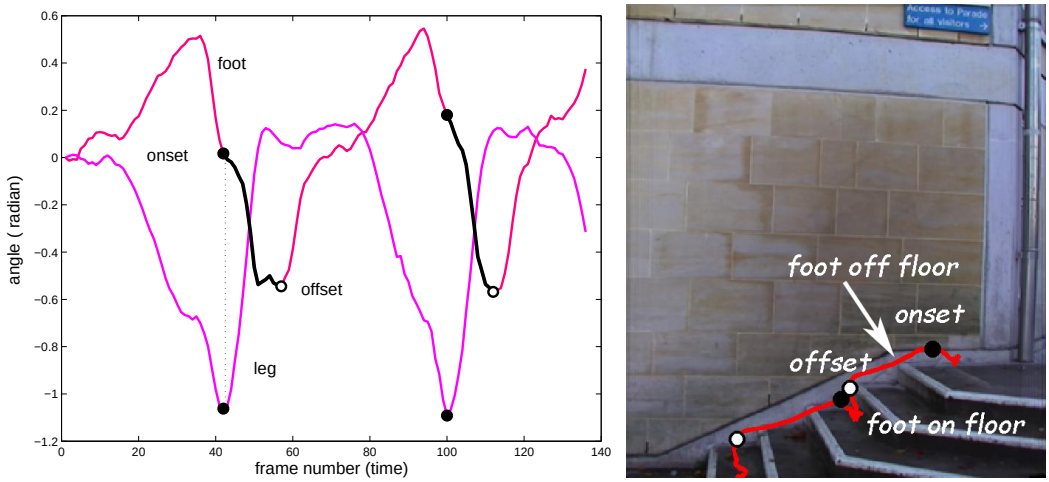

Fig. 8. Left: the foot and lower-left pose trajectories are in quadrature, making the 1/4 cycle during which the foot is off the floor relatively easy to identify. Bottom: The spatial trajectory of the near ankle (red) partitioned into sections between the onset (black) and offset (white) of the 1/4 cycle. The foot is fixed to the floor for the remaining period.

foot to cue streak-lines, as shown in Figure 8. The most robust way to identify these partial cycles is to analyse the pose trajectory of the foot and lower-leg — the limbs that are pivoted by an ankle. These 1/4 cycles correspond exactly to those time periods when the relevant foot is not on the floor, as Figure 8 also shows. In fact the start and stop of the cycle corresponds to salient points on the *spatial trajectory* of the ankle. We note that such analysis provides an opportunity to automate motion exaggeration yet further, not least because it automatically identifies when a foot is solid on the floor.

Finding the minima and maxima of a pose trajectory is complicated by the fact that the signal can be very noisy. Filtering of some kind is clearly necessary, giving rise to the important question of the size of the filter — too narrow a filter leads to too many extrema being identified, too wide a filter yields too few extrema. Ideally we want a filter of a width that is commensurate with about 1/2 cycle, but we do not know this width in advance. Therefore we need a filtering process that both filters the signal and which identifies an appropriate width.

Before explaining how we do this be must rule out low-pass filtering the signal using a linear filter, such as a Gaussian, $\exp(-0.5t^2/\sigma^2)$. Linear filters are not acceptable because they move the signal of interest. We prefer morphological filters what are non-linear, but which do nor shift signals. Influenced by the *sieves* of Harvey and Bangham [21], and also by the concept of "stability" introduced by Matas [22] that fulfils all the properties we seek, and which generalises to wider contexts.

Let $z(t)$ be a noisy discrete signal, and let τ be the half-width of the window

$[z(t-\tau)z(t+\tau)]$ which contains $2\tau+1$ points. We define transform signal $s(t;\tau)$ as

$$s(t,\tau) = \begin{cases} 1 \text{ if } \max([z(t-\tau)z(t+\tau)]) = z(t) \\ 0 \text{ otherwise} \end{cases} \tag{15}$$

which identifies all local maxima that exist within a window of width $2\tau+1$ — the maxima counts only as a maxima if it is the centre of a window. We define $r(t;\tau)$ by analogy to define all local minima in a window of the same size. We account for end conditions, where the window exceeds the boundary of the image, simply by clipping the window.

We perform the filtering over all values of τ from 1 to support width of the discrete signal $z(t)$, there is no value in considering windows of greater width. Thus we obtain a family of filtered signals that are indexed by window width τ. Turning now to use "stability". First we partition "maximum" signals $s(t,\tau)$ into equivalence classes using a suitable measure of change such as $\sum_t \left|\frac{\partial s(t,\tau)}{\partial \tau}\right|^2$. Similarly, we then partition the "minimum" signals by analogy. We then intersect the intervals obtained from both the minimum and maximum signals, and select the intersection of maximum duration. Hence τ_0 is the minimal filtering width giving the most stable filtered signal in both maxima and minima. We filter the signal using τ_0, which empirically turns out to be about 1/2 a cycle; and which therefore allows to to identity the points on the original signal $z(t)$ of interest to us.

Having reliably identified turning points in walking cycle we can return to a description of synthesising Duchamp. Motion blur effects are recreated by "welding" polygons around a limb. As a limb moves through the 1/4 cycle we record the location of its polygon and "weld" these polygons by finding their convex hull. The depth of this amalgamated polygon is fixed at the depth of the contributing limb. When all polygons for all limbs have been amalgamated in this way over the whole time period of the video we acquire a set of depth ordered amalgamated polygons. These are rendered in back-to-front order. By making the polygons partly transparent the visual effect is to entwine the limbs, yet the near polygons appear brighter so that visual sense can be discerned from the picture.

Ghosting and streak-lines are produced using techniques described elsewhere [11]. Ghosting marks are painted on the near limbs only — rendered on top of the welded polygons. Streak-lines are traces of the rear polygons, clipped against the welded polygons of the rear limbs but painted as the top-most layer. Variations in rendering parameters such as the colour of streak-lines, opacity of polygons, and colour of ghosting lead to variations in the final image, as shown in Figure 9.

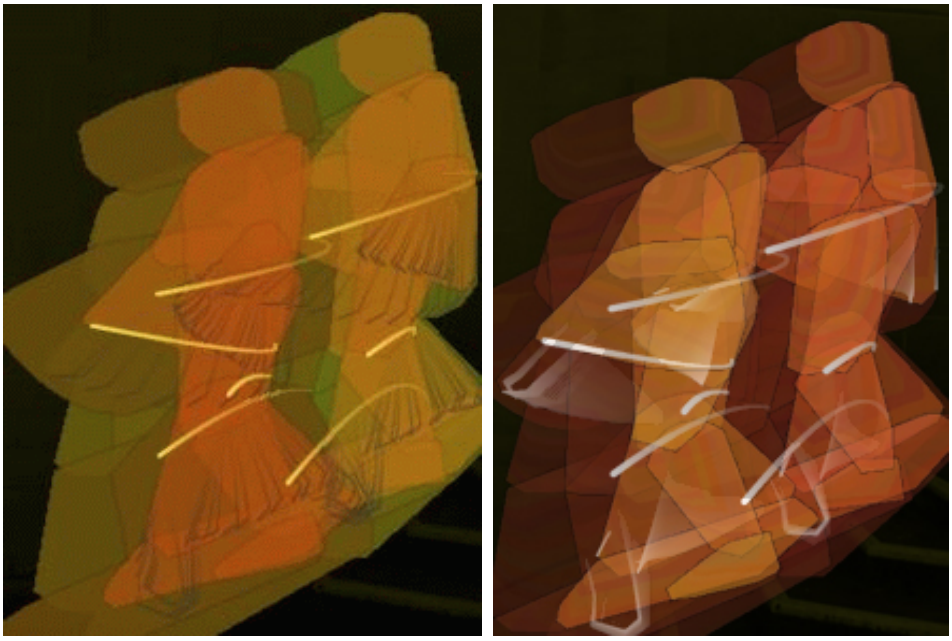

Fig. 9. “Figure descending the steps, version I and II”

4 Concluding remarks

This paper described our initial steps towards automatically synthesising dynamic cues from video, focusing on anticipation and motion exaggeration. Whether the principles we have introduced in addressing these cases generalise easily is unknown. It is likely that inverse kinematics of some kind will play a major role in automating anticipation, although whether pose analysis will ever be of sufficient power to produce the necessary key-frames is an open problem.

As presented, our framework for dynamic cues is premised on the automatic recovery of articulated structures. However initial experiments operated at a lower level of abstraction, requiring no such model and allowing limbs to move without the constraints of pivots. Although more generally applicable, the aesthetics of the resulting motion were disappointing. It is likely that the conceptually higher level model of the articulated structure confers more believable movement because it more closely matches our mental model of the way in which our subjects move. By substituting our hierarchical model with, say, a facial muscle model, we may be able to create anticipation in alternative classes of subject commonly used by animators. Future work might address a methodology for the selection and substitution of models by the computer animator. We might also seek to extend our analysis to three-dimensional affine motion, rather than motion in a plane, which currently restricts the classes of motion that our system is capable of processing.

Open questions notwithstanding, we have introduced a number of useful analysis techniques: automatic inference of articulated structure under planar motion; constrained scaling of pose in eigenspace; a robust signal filter to locate turning points. The dynamic cues synthesised by this framework have been integrated into a larger "Video Paintbox" system (see [10]), capable of both alternative motion emphasis styles (through augmentation and deformation cues) and also flicker-free visual stylisation of content (for example, cartoon shading and painting). In addition, our initial experiments in the emulation of Futurist artwork point toward interesting possibilities for study in NPR, with respect to generating both static depictions of motion and abstract artistic styles. We had not anticipated that a simple analytic explanation might lie behind Duchamp's artwork, and this has certainly added to our appreciation of it.

References

[1] P. Litwinowicz, Processing images and video for an impressionist effect, in: Proc. 24$^{\text{th}}$ Intl. Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH), Los Angeles, USA, 1997, pp. 407–414.

[2] A. Hertzmann, K. Perlin, Painterly rendering for video and interaction, in: Proc. 1$^{\text{st}}$ ACM Symposium on Non-photorealistic Animation and Rendering, 2000, pp. 7–12.

[3] J. P. Collomosse, D. Rowntree, P. M. Hall, Stroke surfaces: Temporally coherent artistic animations from video, IEEE Trans. Visualization and Computer Graphics 11 (5) (2005) 540–549.

[4] J. Wang, Y. Xu, H.-Y. Shum, M. Cohen, Video tooning, in: Proc. ACM SIGGRAPH, 2004, pp. 574–583.

[5] S. Chenney, M. Pingel, R. Iverson, M. Szymanski, Simulating cartoon style animation, in: Proc. 2$^{\text{nd}}$ ACM Symposium on Non-photorealistic Animation and Rendering, 2002, pp. 133–138.

[6] M. Brand, A. Hertzmann, Style machines, in: ACM SIGGRAPH, 2000, pp. 183–192.

[7] C. Bregler, L. Loeb, E. Chuang, H. Deshpande, Turning to the masters: motion capturing cartoons, in: Proc. 29$^{\text{th}}$ Intl. Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH), Jul., 2002, pp. 399–407.

[8] T. Strothotte, B. Preim, A. Raab, J. Schumann, D. R. Forsey, How to render frames and influence people, in: Proc. Computer Graphics Forum (Eurographics), Vol. 13, Oslo, Norway, 1994, pp. C455–C466.

[9] A. Agarwala, A. Hertzmann, D. Salesin, S. Seitz, Keyframe-based tracking for rotoscoping and animation, in: Proc. ACM SIGGRAPH, 2004, pp. 584–591.

[10] J. P. Collomosse, Higher level techniques for the artistic rendering of images and video, Ph.D. thesis, University of Bath, U.K. (May 2004).

[11] J. P. Collomosse, D. Rowntree, P. M. Hall, Video analysis for cartoon-like special effects, in: Proc. British Machine Vision Conference (BMVC), Vol. 2, Norwich, U.K., 2003, pp. 749–758.

[12] J. K. Aggarwal, Q. Cai, Human motion analysis: A review, Computer Vision and Image Understanding: CVIU 73 (3) (1999) 428–440.

[13] J. P. Collomosse, D. Rowntree, P. M. Hall, Cartoon-style rendering of motion from video, in: Proc. Intl. Conference of Video, Vision and Graphics (VVG), 2003, pp. 117–124.

[14] R. E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME – Journal of Basic Engineering 82 (1960) 35–45.

[15] M. Kass, A. Witkin, D. Terzopoulos, Active contour models, Intl. Journal of Computer Vision (IJCV) 1 (4) (1987) 321–331.

[16] R. Williams, The Animator's Survival Kit, Faber, 2001, iSBN: 0-571-21268-9.

[17] L. Liang, H. Chen, Y.-Q. Xu, H.-Y. Shum, Example-based caricature generation with exaggeration, in: Proc. Pacific Conf. on Graphics and Applications, 2002, pp. 386–393.

[18] A. Chen, K. Knudtzon, J. L. Stumpfel, J. K. Hodgins, Artistic rendering of dynamic motion, in: Proc. Computer Graphics (ACM SIGGRAPH Sketches), 2000, p. 100.

[19] J. P. Collomosse, P. M. Hall, Cubist style rendering from photographs, IEEE Transactions on Visualization and Computer Graphics 9 (4) (2003) 443–453.

[20] P. Cabanne, Entretien avec Marcel Duchamp (1967).

[21] A. Bangham, K. Moravec, R. Harvey, M. Fisher, Scale-space trees and applications as filters..., in: 10^{th} British Machine Vision Conference, 1999.

[22] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: Proc. British Machine Vision Conf. (BMVC), Vol. 2, Cardiff, 2002, pp. 384–393.