The Line of Action: an Intuitive Interface for Expressive Character Posing

Martin Guay* Marie-Paule Cani Rémi Ronfard
Laboratoire Jean Kuntzmann - Université de Grenoble - Inria

Figure 1: *Expressive character poses created in a few seconds each, by sketching intuitive lines of action.*

Abstract

The line of action is a conceptual tool often used by cartoonists and illustrators to help make their figures more consistent and more dramatic. We often see the *expression* of characters—may it be the dynamism of a super hero, or the elegance of a fashion model—well captured and amplified by a single *aesthetic* line. Usually this line is laid down in early stages of the drawing and used to describe the body's *principal* shape. By focusing on this simple abstraction, the person drawing can quickly adjust and refine the overall pose of his or her character from a given viewpoint. In this paper, we propose a mathematical definition of the line of action (LOA), which allows us to automatically align a 3D virtual character to a user-specified LOA by solving an optimization problem. We generalize this framework to other types of lines found in the drawing literature, such as secondary lines used to place arms. Finally, we show a wide range of poses and animations that were rapidly created using our system.

CR Categories: I.3.6 [Methodology and Techniques]: Interaction techniques— [I.3.7]: Computer Graphics—Animation

Keywords: Line of action, sketch-based animation, character animation.

Links: DL PDF

1 Introduction

Because humans have been drawing and sketching for centuries, many researchers—from within and from outside the computer graphics community—have proposed and argued for sketching as an intuitive and natural interface for both modeling and editing 3D virtual objects. By designing these interfaces closer to the creative process involved when sketching, the cognitive workload, *i.e.* the mental steps required to achieve a specific task, can be efficiently reduced. Looking at these stages—*i.e.* the ones involved in the creative sketching process [Goldschmidt 1991; Cherlin et al. 2005]—we usually find laid down in early stages, *principal descriptive primitives*—circles, lines, ovals, etc.—that give the overall perspective, shape and mass of objects and characters. For modeling 3D objects, primitive shapes such as cylinders [Gingold et al. 2009] and abstract forms such as contours [Igarashi et al. 1999; Karpenko and Hughes 2006] have been successfully used to ease the sketching process by making coarse shape design easier.

When it comes to character posing however—a task commonly done by manipulating an articulated skeleton used to parametrize the character's geometry—sketch-based modeling research has mostly relied on stick figures as *the* 2D representation of the character's skeletal parametrization [Davis et al. 2003]. Although drawing 2D stick figures accelerates posing, it requires sketching multiple, intersecting strokes for specifying the individual limbs, and getting expressive poses requires skills.

Inspired by the practice of cartoonists [Lee and Buscema 1978; Blair 1994; Hart 1997; Brooks and Pilcher 2001], we explore the alternative of a single, smooth stroke—the line of action (LOA)—as an abstraction of the character's body. We claim that the LOA is an *intuitive* interface that helps creating more *expressive* poses. Controlling the whole body with a single stroke makes early design easier and less time consuming. And the resulting poses are more expressive as they often exhibit an aesthetic curved shape that conveys the full body expression more clearly in a given *viewpoint*. Examples of expressive poses rapidly created with lines of action are shown in Fig.1. They mimic the poses of a fashion model, of two dancers, a full body swing and a super hero on the run.

Since the line of action is an abstraction of the body and the correspondence between both is not explicitly given, posing a 3D virtual character from an arbitrary LOA is a challenge. In this paper, we give a formal definition of the LOA, enabling us to solve character posing by solving an optimization problem. Our method includes an automatic way of determining the *correspondence* between the line of action and a subset of the character's bones. We then propose extensions to the LOA concept for drawing secondary lines and we address the well known depth ambiguities in a new, but simple way;

---

*martin.guay@inria.fr

by constraining the transformations to the viewing plane. We validate this approach by generating both expressive static poses and quickly drafting keyframe animations.

Related Work

Standard character posing methods: Posing virtual characters by manipulating a skeletal parametrization of the character's geometry is common practice in computer animation with previous works dating back to [Burtnyk and Wein 1976]. Positions on the kinematic chains are non linear and manipulating every joint is cumbersome. Inverse kinematics (IK) [Zhao and Badler 1994; Girard and Maciejewski 1985], allows the user to focus on end-effector position targets, while the system automatically solves for the degrees of freedom in the kinematic chains. However, manipulating 3D widgets can be confusing and time consuming. For this reason, several recent works propose to *sketch* the characters in 2D.

Sketch-based character posing: Unlike sketch-based freeform shape modeling—which holds several high-level shape descriptors such as visible contours [Karpenko and Hughes 2006; Igarashi et al. 1999] or annotated shape primitives [Gingold et al. 2009]—sketch-based character posing has relied mostly on stick figures [Davis et al. 2003; Mao et al. 2005; Lin et al. 2010; Wei and Chai 2011; Choi et al. 2012]. In contrast, our work makes use of a higher-level abstraction—the line of action—enabling us to shape characters using one or two smooth strokes, each drawn in a single hand gesture. This makes the posing interface closer to the early stages of the creative process where coarse strokes are used to describe the principal body shapes [Goldschmidt 1991]. It also increases consistency as the lower and upper bodies typically fit onto the same aesthetic curved stroke.

Re-constructing a 3D pose from a 2D sketch is, due to depth, highly ambiguous and under-constrained; many poses exist for the same sketch. A first solution is to let the user choose from several possible solutions as [Davis et al. 2003] suggests. Another approach is to supplement the problem with prior information on the solutions. For instance, physiological insight on the human anatomy leading to specific joint limits, combined with a constraint to maintain balance along additional information on the environment were used to sketch sitting poses [Lin et al. 2010]. The poses can also be constrained to lie in the space of "natural poses", extracted from a database of human motion [Wei and Chai 2011; Choi et al. 2012].

These assumptions could be used to resolve ambiguities within the line of action framework we provide. However, they would also make the tool less flexible. In this work, we propose the alternative approach of constraining skeletal transformations to lie in the viewing plane. This way, users are free to exaggerate bending angles and making unbalanced poses, adding drama to their figures—as often done by cartoonists. On the other hand, the user has to turn the camera in order to "edit depth"—while previous edits orthogonal to the viewing plane are kept unmodified.

The line of action is covered by many textbooks and tutorials on drawing such as [Lee and Buscema 1978; Blair 1994; Hart 1997; Brooks and Pilcher 2001; Doucet 2011]. Although they explain the concept quite well, they never give a formal definition that can be translated into mathematical terms. The books recommend drawing a single line, and then populating the body around it. They strongly advise using only C and S-shaped curves—which tend to produce more aesthetically pleasing, and also more physiologically plausible poses. The line has multiple purposes, best summarized as: *"You can think of it as the back bone of a character or just as the imaginary line that dictates how the body will move"* [Doucet 2011]. In other words, it can be used to render explicit the state of mind of a static character, but also to convey upcoming motion as clearly as possible. In this work, we give a formal definition of the LOA and formalize how it relates to the character's body. Following our observations of these textbooks, we use the line to drive both the positions and tangents of a sub-chain of bones within the character body.

To our knowledge, the first use of lines of action in computer animation was for post-processing animations; *i.e.* to automatically exaggerate motions by stretching the character's limbs along the line [Noble and Tang 2006]. Similarly, [Öztireli et al. 2013] consider lines of action for creating new poses and animations, but their focus is on deforming objects based on curving the bones. In contrast, our approach uses LOA to pose standard articulated models, focusing on the issues in making the LOA an effective interface. A popular tool to pose characters with lines is the *IK spline* [Autodesk 2009]. However, in contrast to our goals, IK splines are 3D curves, are controlled by position targets only, and their mapping to the bones has to be set manually. Lastly, a very intuitive interface based on gestural 2D strokes was used to drive character animations [Thorne et al. 2004]. The method focuses on trajectory and timing specifications to select and combine pre-computed motion clips. In contrast, the lines we draw specify poses, and our tool can be used to create new animation clips.

Overview

This work introduces a direct and natural method for character posing inspired by the way humans tend to sketch shapes in early stages of design. The line of action (LOA), an aesthetic curved stroke, allows the user to quickly specify the character's global shape in a single hand gesture. The pose can then be refined from other viewpoints and the remaining body parts can be posed using so-called secondary lines, as suggested in [Abling 2012]. The method is fully automatic; the user does not need to specify any correspondence between the LOA and the character's body, nor to manually pre-select bones.

Although sketching curves in screen space shares similarities with the 2D stick figures approach introduced by [Davis et al. 2003], using a single line of action as an abstraction of the skeletal parametrization introduces several problems of its own. First, an LOA is a smooth curve, restricted to a specific set of shapes, namely S and C shapes. Second, by looking at many examples such as those in Fig.2 or Fig.4, we notice that the LOA dictates both position *and* tangent targets, but for *only* a *specific* set of bones in the character. Based on these insights, we derive an optimization problem in Section 2 to pose 3D characters from 2D sketches.

This model is sufficient to handle perfectly drawn lines of action whose lengths' are the same as the characters' in screen space. In many cases however, the correspondence is ambiguous; lines are drawn too small or too large locally or globally. A good example is the S-shaped curve often used for feminine stances shown in Fig.4. We can see an area near the pelvis which seems to correspond to none of the bones in the body. We propose an automatic method to solve this problem. At each step we solve for an optimal reparametrization of the *correspondence* between the line of action and the bones—which we describe in Section 3.

Also, the "3D pose from 2D sketch" problem is well known for being under constrained; not enough information is provided by the sketch to fully specify the solution. Since one our goals is to allow *expressive*, sometimes *exaggerated* and *non-realistic* poses, we make only minimal assumptions on the character's physiology. To constrain the problem properly while allowing flexibility in the pose, we propose to constrain the character transformations to lie in the viewing plane (Section 5). Consequently, the user has to rotate in order to "edit depth" without ambiguity. This allows the user

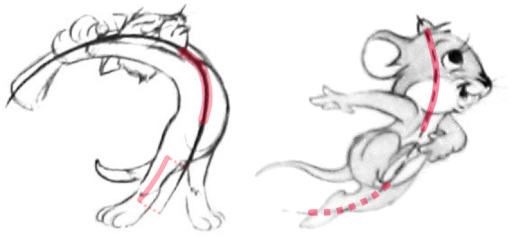

Figure 2: *Our definition of the line of action: The line is a target for* **a subset** *of the kinematic chain from a particular viewpoint. The bones are generally a lower leg, the spine, the head and sometimes the arms. ©The Estate of Preston Blair, [Blair 1994].*

to draw poses not restricted by physiological assumptions such as balance, joint constraints, or pose priors—although such constraints are also compatible with our framework. Finally, we show results of our tool in section 6 and accompanying video.

2 The Line of Action

The line of action's description often differs from one artist to the next. To formalize the concept, we provide a definition which, in our view, spans a wide range of cases. A first aspect is the range of possible shapes an LOA can take, and the second is its relation to the character's pose. Both are discussed below.

LOA-shaped curves: In most textbooks on drawing, the LOA is restricted to the family of *C*- and *S*-shaped 2D curves—which we refer to as LOA-shaped curves. This restriction can be a useful for two reasons: it constrains the poses to aesthetically pleasing shapes, and to physiologically plausible poses. We define C and S-shaped curves as having zero and one inflexion points respectively. To naturally reduce the variability of the line, we fit a cubic Hermite curve, defined by two end points and two tangent vectors. This is done as follows: the end-points of the Hermite curve are positioned at the extremities of the stroke. We then minimize the squared distance between sample points along the stroke and the curve positions, w.r.t. the angles and lengths of the tangent vectors, represented in polar coordinates. This approximation is sufficient for representing the usual C and S-shaped curves. More freedom to the user is given by fitting more curves to the input stroke; in our case two curves are used, each capturing half of the user's stroke.

Using the LOA to drive the pose: We use the LOA to dictate the positions and tangents in screen space of *a subset* of the character's bones—which we call the *body line*. For instance, we often see the line setting the bones going from the head to a foot—the yellow bones in Fig.3. Our definition of the body line is as follows:

The **body line** is defined as a maximal, connected linear chain in a character's skeletal kinematic tree.

Maximal: Most body lines start and end at extremes of the full kinematic tree. Restricting our definition to maximal lines leads to a set of 10 possible kinematic chains. Different ways of choosing between these *body lines* are discussed in Section 4. For the sake of clarity, we will continue our description supposing a classic *body line*, *i.e.* a chain between the head and one of the feet.

Connected: The bones are children or parents of one another in the full kinematic tree. For instance, if the LOA goes from one foot to the other, it also includes the two lower and upper legs.

Linear: Each node between two bones, that are not the extremities, are of degree 2. The extremity nodes are degree 1 nodes; they have only one parent or one child.

There are two important aspects to note about the body line in regard to the line of action. One is the importance of *shape*: it implies that tangents should prevail over positions. It seems more important for the character to match the *shape* of the LOA than having the character's body line be *close* to the line of action. This will be reflected by a tangent matching constraint in our model.

The second is that some bones seem to be more important, and to *align* more with the curve. For instance, in Fig.4, the lower leg, the spine and the head seem to be matching the shape of the curve, while the upper leg, the pelvis and the neck account for less. Although we describe a general model, we find important to emphasize that these *principal bones* will weigh more in practice. Both the sketched LOA (2D) and the character's *body line* (3D), can be seen as one dimensional parametric curves. For the sake of clarity, we use the same parameter s for both curves. In Section 3, we will make our solution more general by finding an optimal warping function between bone coordinates and curve coordinates.

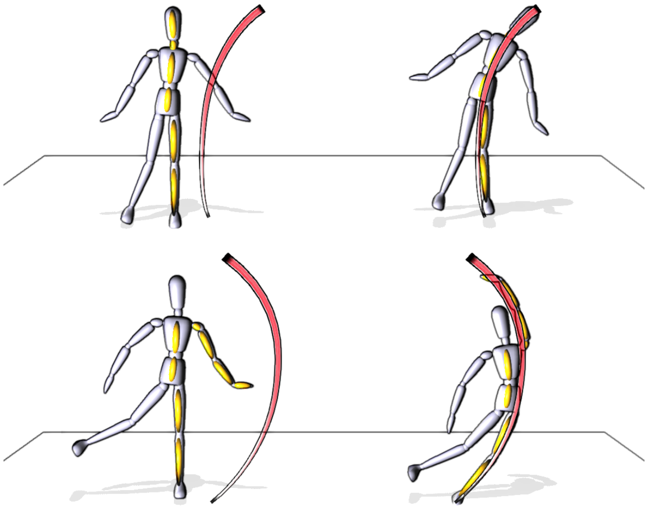

Figure 3: *In yellow, on the left, are the bones forming the body line that will be deformed according to the sketched line of action. On the right is the result after solving the LOA problem (1). In the top image, the body line goes from the head to a foot. And in the second, it goes from a hand to a foot.*

Let the position in screen space of the line of action be $\mathbf{x}_{loa}(s)$, the bones' positions $\mathbf{x}_b(s)$ and their position in screen space $\mathbf{P}_{\text{vp}}\mathbf{x}_b(s)$, where $\mathbf{P}_{\text{vp}}$ is a view and *perspective* projection transformation in homogenous coordinates.

Following our definition, the best pose $\mathbf{x}_b(s)$ of the character minimizes the shape difference between the two curves. Importantly, shape is defined here both in terms of positions and tangents. We thus solve for:

$$\min_{\mathbf{x}_b(s)} \int_s E_{\mathbf{x}}(s) + E_{\widehat{\mathbf{T}}}(s)\, ds, \tag{1}$$

$$E_{\mathbf{x}}(s) = \lambda_{\mathbf{x}}(s) \left\| \mathbf{P}_{\text{vp}}\mathbf{x}_b(s) - \mathbf{x}_{loa}(s) \right\|^2,$$

$$E_{\widehat{\mathbf{T}}}(s) = \lambda_{\widehat{\mathbf{T}}_{\mathbf{loa}}}(s) \left\| \widehat{\mathbf{T}}_b(s) - \widehat{\mathbf{T}}_{loa}(s) \right\|^2,$$

where $\mathbf{T}_b(s) = \frac{\partial \mathbf{P}_{\text{vp}}\mathbf{x}_b}{\partial s}(s)$, $\mathbf{T}_{loa}(s) = \frac{\partial \mathbf{x}_{loa}}{\partial s}(s)$, and $\widehat{\mathbf{T}}$ denotes *unit* tangents. The terms $\lambda_{\mathbf{x}}(s)$ and $\lambda_{\widehat{\mathbf{T}}}(s)$ are used to emphasize tangents over positions, and to emphasize the importance of specific bones over others, *e.g.* the *principal bones*. For instance, for the classic head-to-foot body line, the lower leg, the spine and the head usually weigh more than the upper leg, pelvis and neck. We minimize w.r.t. to the full body line $\mathbf{x}_b(s)$, using a discrete version of the problem given in the appendix.

3 Automatic Correspondence

Driving the character's body line from a single stroke, requires identifying which part of the body corresponds to which part of the stroke. In Section (2), we supposed a known mapping, expressed through a common coordinate s, but this assumption does not hold in practice: the user-sketched stroke is often ambiguous and gives no information on which part of the LOA the bones should correspond to. Moreover, two simple ideas for setting this mapping just do not work:

- A first idea would be to match each projected bone to the nearest point on the LOA. Although this works well for a character that is already "near" the curve or already well aligned with the curve, it falls short for ambiguous curves. Additionally, the closest point can lead to unexpected character configurations as shown in Fig.4; we see the upper body sliding to the middle of the line of action when we would expect it to remain in the upper part of the LOA.
- A second idea is to consider the length of the bones. However, the sketched line is often drawn imperfect with local segments having a different length than the projected body line's length. We even found cases where some parts of the LOA correspond to nothing in the body, as with the S-shaped curve shown in Fig.4—with a jump visible near the pelvis.

We address these problems by finding an optimal spatial warping between the *body line* coordinate s and the line of action coordinate $w(s)$. We formulate our approach on the closest point $E_{\mathbf{x}}(s, w(s))$ objective, but add regularization on the warping function $w(s)$ based on the assumptions of rigidity and connectivity of the skeleton's tree structure. Note that rigidity here is used to find the mapping, but the bones can be made to bend in a final step simply by breaking them into smaller ones.

In short, assuming the bones are rigid implies they should avoid covering areas on the LOA with *high curvature* which translates into constraint $E_\kappa(w(s))$. To allow *jumps* in the solution (see Fig.4), the warping function is defined piecewise, each corresponding to a rigid bone interval on the body line denoted A_i. Assuming a skeletal tree structure, the associated warping functions should preserve the parent-child relations established by the skeleton; bones should not overlap or go too far from one another—making a connectivity constraint $E_C(w(s))$ relevant. The final optimization problem is as follows:

$$\min_{w} \int_s E_{\mathbf{x}}(s, w(s)) + E_\kappa(w(s)) + E_C(w(s)) \;\; ds \;, \qquad (2)$$

$$E_\kappa(w(s)) = \lambda_\kappa \left\| \frac{\partial^2 \mathbf{x}_{loa}(w(s))}{\partial^2 s} \right\|^2 ,$$

$$E_C(w(s)) = \lambda_C \left\| \chi_{i+1}(s) w(s) - \chi_i(s) w(s) \right\|^2 ,$$

where the term $E_{\mathbf{x}}(s, w(s))$ is the same as in problem (1), but here the line of action coordinate is warped $w(s)$. The term $\chi_i(s)$ is a step function worth 1 at end points of the rigid bone interval A_i and 0 elsewhere. Note that we optimize for the warping function on the fly; *i.e.* at each step when solving problem (1), we solve problem (2) for an optimal warping function. Between successive steps, convergence for problem (2) is quite fast when initialized with the previous step solution. More details on our discretization are given in the appendix.

4 Selecting the Body Line

The line of action can be used to modify different sets of bones, or body lines. Sometimes the *body line* includes an arm, sometimes both legs, while in other cases it includes only the arms—as with *secondary lines* [Abling 2012].

Our definition of the body line was designed to include all of these cases. Due to the maximality criterion, we only have 10 body lines for a humanoid: *head to left hand, head to left foot, head to right foot, head to right hand, left hand to left foot, left hand to right foot, left hand to right hand, left foot to right foot, right hand to left foot and right hand to right foot*. Selecting the right body line in the right viewpoint can be viewed as an artistic choice. Because we wanted to offer a simple interface, we propose an automatic way of selecting the "most appropriate" body line for a given line of action and viewpoint.

We propose two interfaces from which the user can choose in order to select a body line. The first is manual. The user either draws a first line close to the bones he wishes to include, or he selects the body line from a tab. Finally he draws the line of action. With the second interface, the user draws the target line of action directly and the system automatically chooses the body line with the smallest energy (1) from the 10 possible candidates. When two body lines have similar energies, we take the closest one in view space.

5 Resolving Depth Ambiguities

The line of action problem (1), which seeks a 3D pose from a 2D sketch given a viewpoint, is under constrained; many solutions (3D poses) exist for the same 2D line of action. For instance, we can simply imagine a line drawn shorter than the character in screen space. It could mean the character has translated away from the camera, but it could also mean that he has crouched. Since we don't make any assumptions other than the piecewise-rigidity of the skeletal structure—in order to allow non-realistic poses—we need additional constraints to resolve depth ambiguities.

Using the 2D LOA to drive the character in all three dimensions can be confusing for the user due to the unexpected character configurations; several 3D poses can be a minimum to problem (1). To "remove" depth, we constrain the character's transformations to lie in the viewing plane—which also reduces ambiguities when solving problem (1). We parametrize each bone rotation to a single axis-angle component θ_i as illustrated in Fig.5. The corresponding axis is the camera's viewing direction projected onto the floor plane. The translations are also parametrized to lie in the viewing plane, *i.e.* along the 2-components u, v of the viewing plane. Finally, we solve the LOA problem (1) with respect to θ_i and u, v.

With our approach, the user can separately sketch and rotate the camera to "edit in depth". As an example, if he wants to bend the character's knees, he has to look at the character sideways and draw a curve indicating a bent leg as shown in Fig.5. Working only in the viewing plane removes a lot of the ambiguities—both for solving the problem *and* for the user as well. Without this constraint, we would necessarily have to supplement the problem with other

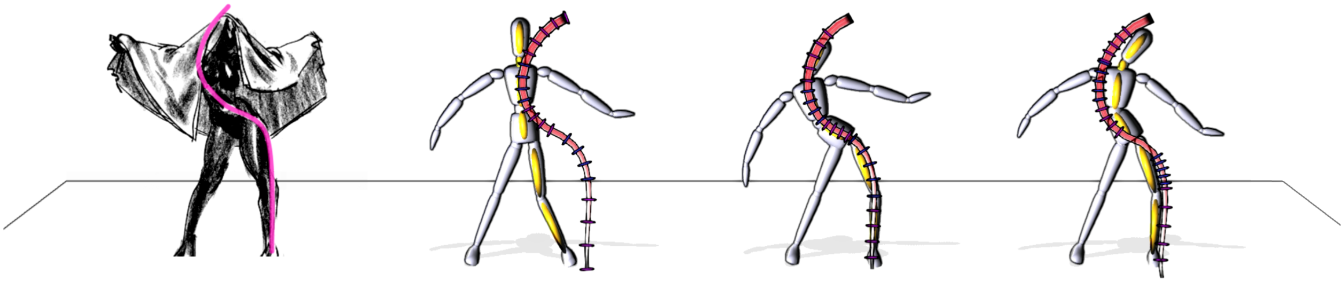

Figure 4: **Left:** *A classic S-shaped curve used for feminine stances. It is difficult to know which part of the line, the bones should map to—not to mention parts near the pelvis that correspond to none of the bones. The three last images are: the initial pose, the result parameterization using the closest point only and finally, our result with curvature-based regularization. Left figure: ©Ben Jelter.*

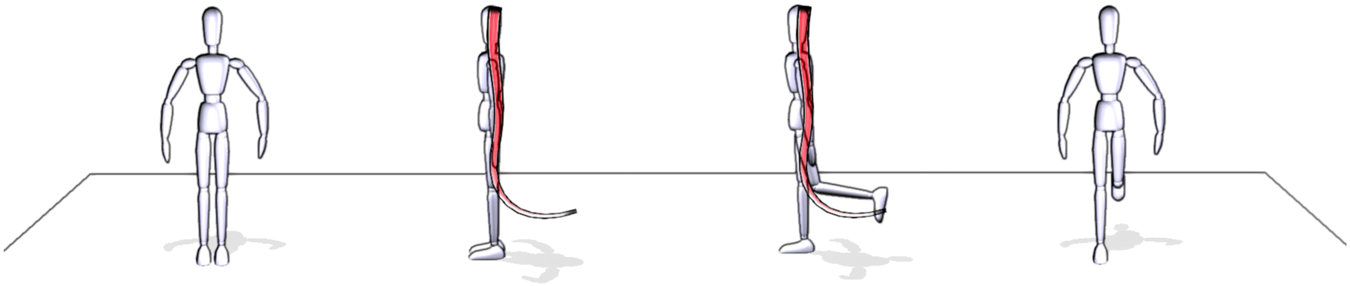

Figure 5: *Bone rotations are constrained to the viewing plane. They are parametrized as a single axis-angle component. The axis is the camera's viewing direction projected onto the floor plane.* **From left to right** *is the initial pose, a side view with the line of action stroke, the result after optimization, and a frontal view of the final pose.*

constraints such as ensuring poses lie in the space of prior "natural poses"[Wei and Chai 2011]. Although it would be very useful to do so, especially for beginners, it would also restrict the space of poses that can be created. The system used in this paper gives entire freedom to create non-realistic and exaggerated poses with arbitrary sequences of sketching and rotation gestures.

6 Results & Discussion

As a proof of concept, we evaluated our method with the task of reproducing static poses from photographs, cartoon images, and finally, example motions from video frames. The poses were then exported to Maya for keyframe interpolation. The implementation is a stand-alone software that offers all the necessary tools for drawing multiple lines of action in multiple viewpoints, and computing the corresponding poses at interactive rates. In our implementation, we solve the discrete version of problem (1)—discussed in the appendix—using gradient-based local optimization. At the start of each step, we solve the discrete version of problem (2) for an optimal reparametrization. In both cases we optimize until the gradient's length is smaller than a small criterion. The optimization is interactive and can be visualized on-line.

All the poses and animations were made by the authors of this paper—who are not animation experts—using lines of action only. For the evaluation, we measure the total number of user gestures (strokes and rotations), as well as the actual time used to generate a satisfactory result. We start from a neutral pose, then draw all the strokes necessary to complete the pose or sequence of poses. Then we asked a professional animator (with five years experience) to reproduce the same poses using IK widgets in Maya. We measured the time required to finish the same task and summarized our evaluations in a table. Our preliminary evaluation demonstrates that all the example poses could be generated within reasonable times.

Figure	Num. Strokes & Rotations	Time LOA	Time Maya
Walk (Fig.10)	4 st. 1 r.	20 s.	90 s.
S shape (Fig.4)	3 st. 0 r.	20 s.	120 s.
Hero Punched (Fig.11)	6 st. 2 r.	90 s.	150 s.
Hero Punch (Fig.11)	7 st. 4 r.	2 m.	3 m. 30 s.
Animation			
Dancers (Fig.8)	6 st. 1 r.	30 s.	2 m. 45 s.
Cartoon swing (Fig.7)	7 st. 1 r.	45 s.	2 m. 30 s.
Muybridge (Fig.9)	34 st. 1 r.	6 m.	22 m.

Figure 6: *Number of strokes, rotations and time taken using lines of action v.s. using 3D IK widgets in Maya.*

While we have not performed a complete comparison, we extrapolate that our tool can be used to produce rough poses and keyframes in a fraction of the time it would take with existing tools. The benefits seem especially spectacular when considering longer animation sequences like the *Muybridge* sequence—shown in Fig.9 and the accompanying video. It took a total of 6 min. to make using lines of action while it took 22 minutes to a professional artist using traditional 3D IK widgets in Maya.

One of the reasons for this speedup is the reparametrization described in Section 3. Thanks to this adjustment, we can now sketch bent knees and elbows more freely without having to produce an accurate stroke each time. This is beneficial in reducing the number of strokes required to produce the poses—not to mention poses like the feminine stance (Fig.1) that are nearly impossible to reproduce without a reparametrization. But the foremost benefit of our method is the fact that beginners *could* actually create expressive poses using a line of action, while there is no way in Maya for beginners to easily create poses that exhibit an aesthetic curved shape.

The line of action is particularly useful to create dramatic poses that convey full body expressions more clearly.

In practice we found that most keyframes were made by sketching 3 strokes: one to specify each half of the full body, then a third to adjust the arms using a *secondary line*. Moreover, it is worth noting that once a pose is created, variations of the same pose can be easily tried-out by drawing slight variations of the initial line. This explains why making a basic walking pose (Fig.10) takes only 20 seconds while making a dramatic punch destined for comics like the one in Fig.11 takes minutes; we tend to redraw the same stroke over previous ones to perfect the pose and make it more dramatic.

The poses and animations shown in this paper and video were made using lines of action only. However, there are many poses that just cannot be *fully* made using lines of action only—insofar the line is restricted to S- and C-shaped curves (see Section 2). Sharp corners such as completely bent knees cannot be reproduced using a limited number of smooth parametric curves. On the other hand, this option is useful to constraint the user to make more consistent poses at the coarse level—a feature useful for beginners. Of course, LOA constraints can be combined with other techniques better suited for small scale detail editing.

A limitation of our viewing plane constraint is that bones can no longer twist when orthogonal to the viewing direction. We explored the possibility of letting the bones rotate around their own direction axis, but have not yet established an intuitive way of controlling this feature. Up to now we used the secondary lines for the arms or legs in order to twist the upper and lower bodies. For instance, this was used to produce Fig.11. Future work could investigate considering the length of each arm or leg in order to estimate the amount of twist required.

7 Conclusion & Future Work

We introduced an intuitive interface for expressive character posing based on the line of action metaphor. We gave a formal definition of the line and provided solutions to several ambiguities related to this abstraction of the body; namely correspondence and depth ambiguities. Our method enable the user to rapidly create expressive poses ranging from elegant fashion stances to exaggerated cartoon actions. We briefly compared our method with traditional authoring software and found it was significantly faster for coarse pose design—especially for *expressive* poses that are challenging to create using traditional animation techniques. This new tool for character posing seems promising and opens several questions for future investigation.

One of the advantages of the optimization problem (1) is that we can easily add additional constraints. For instance, when the line intersects the floor, we can handle a foot or hand placement IK target. This simply corresponds to an additional constraint in problem (1). We extended the line of action concept to several kinds of lines, but we restricted our definition to *maximal* chains (Section 2). It would be convenient to allow individual body parts like a leg or an arm to be sketched individually to refine poses without having to sketch a maximal line each time. On the other hand, our definition of the line is quite general and we could apply it directly to other morphologies like quadrupeds. An LOA could be formed from the head going through the spine and finishing at a foot. Secondary lines would go from one leg to the other. Also, we could think of refining small scale details like hands following the same logic. A line could be used to set one of the fingers to the thumb, or one finger to the elbow.

Our goal was to make the line of action as simple as possible for the user. For this reason we proposed fully automatic solutions to the problems we faced; namely the *correspondence* issue (Section 3) and the *body line* choice (Section 4). However there are cases where the user might want a different correspondence, or a different body line to modify. A semi-automatic approach could be envisioned here were the user would scribble ([Noris et al. 2013]) or sketch small primitives to help both resolve the correspondence *and* specify the choice of internal body line. These insights are left for future work, but seem compatible with our framework; they could be inserted as additional constraints in problems (2) and (1) respectively.

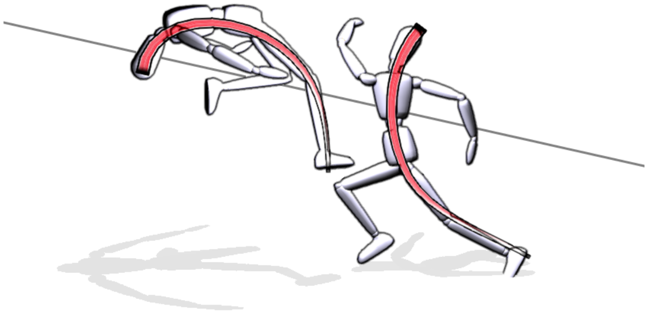

Figure 11: *Inspired by comics, these characters are posed fighting. In each of these poses, we can see the overall body expression controlled by a line of action.*

Acknowledgements

We thank Laura Paiardini for help with Maya scripts, rendering and animation comparisons. We also thank the anonymous reviewers and Mike Gleicher for their useful comments and suggestions. This work was partially funded by the ERC advanced grant EXPRESSIVE.

References

ABLING, B. 2012. *Fashion Sketchbook*. Fairchild Publications.

AUTODESK. 2009. Maya. `http://www.autodesk.com/maya`.

BLAIR, P. 1994. *Cartoon Animation*. Walter Foster.

BROOKS, B., AND PILCHER, T. 2001. *The Complete Cartooning Course*. David & Charles Publishers.

BURTNYK, N., AND WEIN, M. 1976. Interactive skeleton techniques for enhancing motion dynamics in key frame animation. *ACM Transactions on Graphics (TOG) 19*, 10, 564–569.

CHERLIN, J. J., SAMAVATI, F., SOUSA, M. C., AND JORGE, J. A. 2005. Sketch-based modeling with few strokes. *Proceedings of the 21st spring conference on Computer graphics*, 137–145.

CHOI, M., K., Y., IGARASHI, T., MITANI, J., AND J., L. 2012. Retrieval and visualization of human motion data via stick figures. *Pacific Graphics*.

DAVIS, J., IGARASHI, M., CHUANG, E., POPOVIC', Z., AND SALESIN, D. 2003. A sketching interface for articulated figure animation. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 320–328.

DOUCET, R. 2011. Line of action-mickey's christmas carol. `http://floobynooby.blogspot.fr/2011/01/`

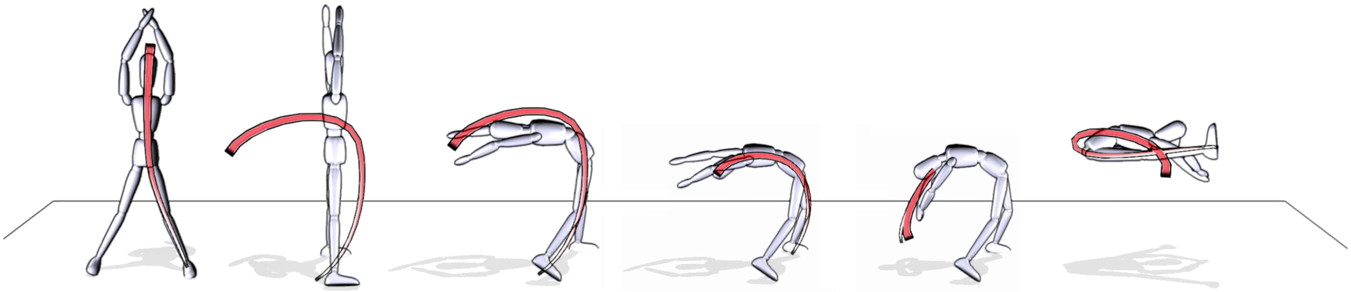

Figure 7: *The main poses of a cartoon character swinging a bat take only 8 strokes and 1 rotation. The overall process took less than a minute to finish compared to several minutes with traditional 3D IK widgets.*

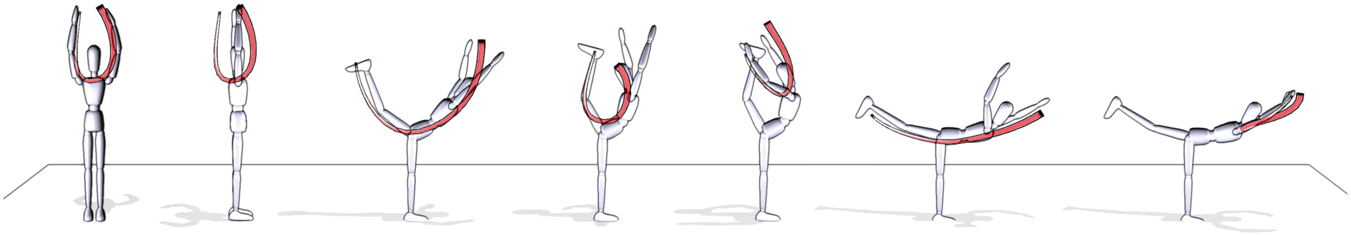

Figure 8: *Dance poses are often LOA-friendly with the body exhibiting an aesthetically curved shape.*

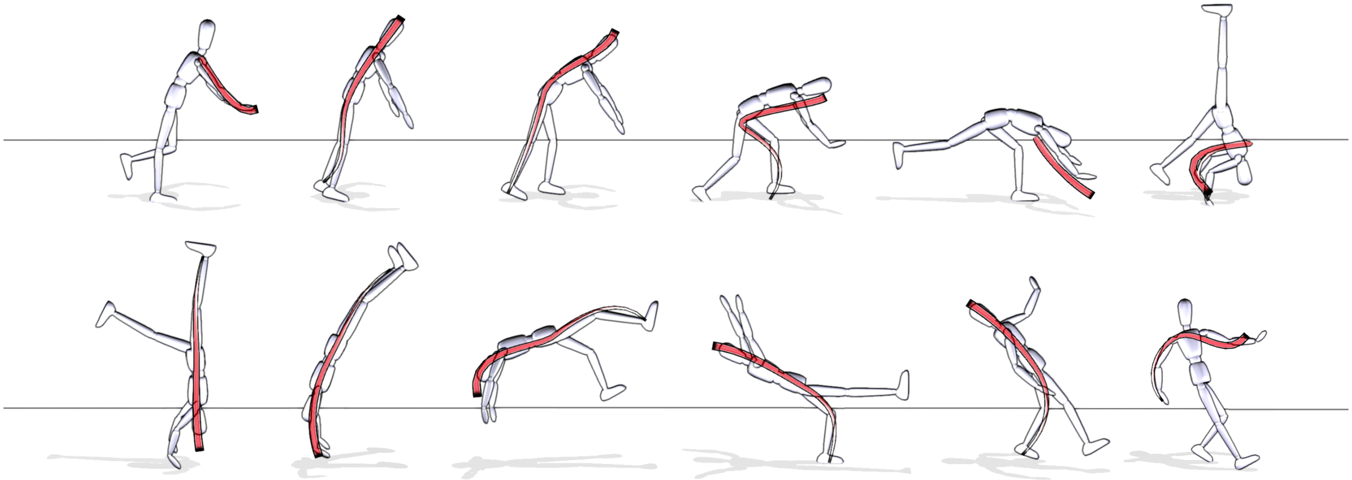

Figure 9: *A complete Muybridge sequence can be reproduced in 6 minutes by drawing lines of action in a single view. We show, for each frame, the last line of action that was drawn.*

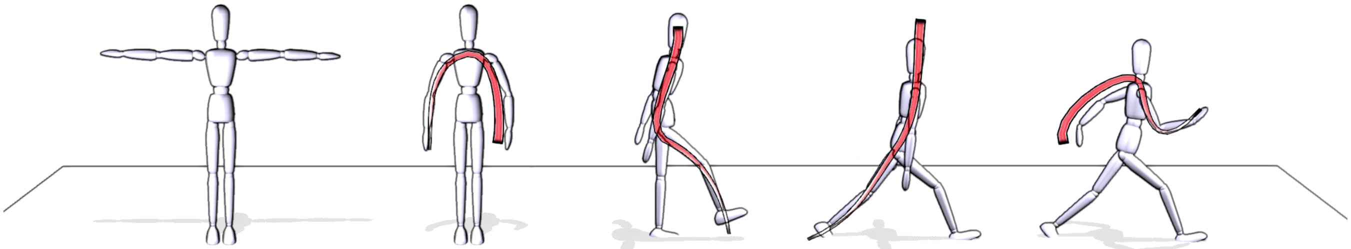

Figure 10: *Starting from a neutral pose, only 4 strokes are required to get started with a walking animation. Note the so-called secondary lines are used here to place the arms.*

`lines-of-action-mickeys-christmas-carol.html`.

GINGOLD, Y., IGARASHI, T., AND ZORIN, D. 2009. Structured annotations for 2D-to-3D modeling. *ACM Transactions on Graphics (TOG) 28*, 5, 148.

GIRARD, M., AND MACIEJEWSKI, A. A. 1985. Computational modeling for the computer animation of legged figures. *ACM Transactions on Graphics (TOG) 19*, 3, 263–270.

GOLDSCHMIDT, G. 1991. The dialectics of sketching. *Creativity Research Journal 4*, 2, 123–143.

HART, C. 1997. *How to draw animation*. Watson-Guptill.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a sketching interface for 3d freeform design. *ACM Transactions on Graphics (TOG)*, 409–416.

KARPENKO, O. A., AND HUGHES, J. F. 2006. Smoothsketch: 3d free-form shapes from complex sketches. *ACM Transactions on Graphics (TOG) 25*, 589–598.

LEE, S., AND BUSCEMA, J. 1978. *How To Draw Comics The Marvel Way*. Simon & Schuster Inc.

LIN, J., IGARASHI, T., MITANI, J., AND SAUL, G. 2010. A sketching interface for sitting-pose design. *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, 111–118.

MAO, C., QIN, S. F., AND WRIGHT, D. 2005. A sketch-based gesture interface for rough 3d stick figure animation. *Proceedings of the 2005 Eurographics workshop on Sketch-Based Interfaces and Modeling*.

NOBLE, P., AND TANG, W. 2006. Automatic expressive deformations for stylizing motion. *GRAPHITE '06 Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, 57–63.

NORIS, G., SÝKORA, D., SHAMIR, A., COROS, S., WHITED, B., SIMMONS, M., HORNUNG, A., GROSS, M., AND SUMNER, R. 2013. Smart scribbles for sketch segmentation. *Computer Graphics Forum 31*, 8, 2516–2527.

ÖZTIRELI, A. C., BARAN, I., POPA, T., DALSTEIN, B., SUMNER, R. W., AND GROSS, M. 2013. Differential blending for expressive sketch-based posing. In *Proceedings of the 2013 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

THORNE, M., BURKE, D., AND VAN DE PANNE, M. 2004. Motion doodles: an interface for sketching character motion. *ACM Transactions on Graphics (TOG) 23*, 424–431.

WEI, X. K., AND CHAI, J. 2011. Intuitive interactive human character posing with millions of example poses. *IEEE Computer Graphics and Applications 31*, 78–88.

ZHAO, J., AND BADLER, N. 1994. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics 13*, 313–336.

Appendix

We parametrize the bone position function $\mathbf{x}_b(s)$ as a chain of piecewise-rigid bones: a single root position at the pelvis $\mathbf{x}_r$ with a quaternion rotation $\mathbf{Q} = \{\mathbf{q}_1, ..., \mathbf{q}_m\}$ for each bone. The chain holds $m+1$ nodes and we denote $\mathbf{x}_i^b$ the *bone position* at node i on the *body line*. The body line coordinate s is discretized into intervals—one for each bone—starting at s_i and covering $\Delta s_i = s_{i+1} - s_i$ space. We denote $s_{i,j} = s_i + \frac{j}{n}$ a coordinate between two nodes and build a piecewise continuous bone position function by linearly interpolating between the nodes: $\mathbf{x}^b(s_{i,j}) = \left(1 - \frac{j}{n}\right)\mathbf{x}_i^b + \left(\frac{j}{n}\right)\mathbf{x}_{i+1}^b$, for a subinterval of $n+1$ samples ($n = 4$).

The LOA curve position is parametrized as a cubic Hermite curve (see Section 2) and can be sampled between nodes as: $\mathbf{x}^{loa}(w_{i,j})$, where each node has a warped coordinate $s_i \to w_i$, and $w_{i,j} = w_i + \frac{j}{n}$ are the coordinates between the nodes. By discretizing the partial derivatives in problem (1) in finite difference, the discrete problem can be written as:

$$\min_{\mathbf{Q},\mathbf{x}_r} \sum_{i=1}^{m+1} \; E_i^x + E_i^{\widehat{T}}$$

$$E_i^x = \frac{\lambda_i^{\mathbf{x}}}{n+1} \sum_{j=0}^{n} \left\| \mathbf{P}_{\text{vp}} \mathbf{x}^b(s_{i,j}) - \mathbf{x}^{loa}(w_{i,j}) \right\|^2$$

$$E_i^{\widehat{T}} = \frac{\lambda_i^{\widehat{\mathbf{T}}}}{n+1} \sum_{j=0}^{n} \left\| \widehat{\mathbf{T}}^b(s_{i,j}) - \widehat{\mathbf{T}}^{loa}(w_{i,j}) \right\|^2$$

$$\mathbf{T}^b(s_{i,j}) = \frac{\mathbf{P}_{\text{vp}}\mathbf{x}^b(s_{i,j+1}) - \mathbf{P}_{\text{vp}}\mathbf{x}^b(s_{i,j})}{\Delta s_j}$$

$$\mathbf{T}^{loa}(w_{i,j}) = \frac{\mathbf{x}^{loa}(w_{i,j+1}) - \mathbf{x}^{loa}(w_{i,j})}{\Delta s_j}$$

where $\Delta s_j = \frac{1}{n}\Delta s_i$ are subintervals' lengths. The matrix $\mathbf{P}_{\text{vp}}$ is the view and perspective projection transformation in homogenous coordinates and $\widehat{T} = \frac{T}{\|T\|}$ denotes unit tangents. The *principal bones* can be weighted more through $\lambda_i^{\mathbf{x}}$ and $\lambda_i^{\widehat{\mathbf{T}}}$ with the later being larger than the former to emphasize tangent matching over position matching. Note that in our implementation, the rotations are parametrized to a single axis-angle component θ_i constraining the transformations to the viewing plane and reducing the number of degrees of freedom to solve for (see Section 5). Finally, we minimize w.r.t. $\Theta = \{\theta_1, ..., \theta_m\}$ and $\mathbf{x}_r$.

The warping function is piecewise-defined—one segment for each bone in the body line—accounting for potential jumps in the solution (see Section 3). We discretize these segments into two components $\mathbf{W} = \{[w_1^0, w_1^1], ..., [w_m^0, w_m^1]\}$, one for each extremity of the bone. This means nodes with two edges also have two coordinates; we average both to sample the curve at a node i as $w_i = 0.5(w_i^0 + w_{i-1}^1)$. By discretizing the partial derivatives in problem 2, a discrete version of the problem is:

$$\min_{\mathbf{W}} \sum_{i=1}^{m+1} E_i^x + E_i^\kappa + E_i^C$$

$$E_i^x = \frac{\lambda^{\mathbf{x}}}{n+1} \sum_{j=0}^{n} \left\| \mathbf{P}_{\text{vp}} \mathbf{x}^b(s_{i,j}) - \mathbf{x}^{loa}(w_{i,j}) \right\|^2$$

$$E_i^\kappa = \frac{\lambda^\kappa}{n+1} \sum_{j=0}^{n} \left\| \frac{\mathbf{x}^{loa}(w_{i,j+1}) - 2\mathbf{x}^{loa}(w_{i,j}) + \mathbf{x}^{loa}(w_{i,j-1})}{{\Delta s_j}^2} \right\|^2$$

$$E_i^C = \lambda^C \left\| w_{i+1}^0 - w_i^1 \right\|^2$$

where $\lambda^{\mathbf{x}}$ is to chose the closest point on the line while the terms λ^κ and λ^C are used to emphasize rigidity of the bones and their connectivity. Finally we minimize w.r.t. the warping function $\mathbf{W}$ on the fly, between steps for problem (1).