

# Highly Stylised Drawn Animation

Fabian Di Fiore<sup>1</sup>, Frank Van Reeth<sup>1</sup>, John Patterson<sup>2</sup>, and Philip Willis<sup>3</sup>

<sup>1</sup> Hasselt University  
Expertise Centre for Digital Media  
transnationale Universiteit Limburg  
Wetenschapspark, 2  
BE-3590 Diepenbeek (Belgium)  
{fabian.difiore, frank.vanreeth}@uhasselt.be  
<http://www.edm.uhasselt.be>

<sup>2</sup> Dept. of Computing Science Glasgow University 17 Lilybank Gardens Glasgow G12 8QQ (Scotland, UK) <a href="mailto:jwp@dcs.gla.ac.uk">jwp@dcs.gla.ac.uk</a> <a href="http://www.dcs.gla.ac.uk">http://www.dcs.gla.ac.uk</a>	<sup>3</sup> Media Technology Research Centre Dept. of Computer Science University of Bath Bath Avon BA2 7AY (England, UK) <a href="mailto:P.J.Willis@bath.ac.uk">P.J.Willis@bath.ac.uk</a> <a href="http://www.bath.ac.uk/media">http://www.bath.ac.uk/media</a>
---	--

**Abstract.** In this paper we argue for our NPAR system as an effective 2D alternative to most of NPR research which is focused on frame coherent stylised rendering of 3D models. Our approach gives a highly stylised look to images without the support of 3D models, and yet they still behave as though animated by drawing, which they are.

First, a stylised brush tool is used to freely draw extreme poses of characters. Each character is built up of 2D drawn brush strokes which are manually grouped into layers. Each layer is assigned its place in a drawing hierarchy called a Hierarchical Display Model (HDM). Next, multiple HDMs are created for the same character, each corresponding to a specific view. A collection of HDMs essentially reintroduces some correspondence information to the 2D drawings needed for in-betweening and, in effect, eliminates the need for a true 3D model.

Once the models are composed the animator starts by defining keyframes from extreme poses in time. Next, brush stroke trajectories defined by the keyframe HDMs are in-betweened automatically across intermediate frames. Finally, each HDM of each generated in-between frame is traversed and all elements are drawn one on another from back to front. Our techniques support highly rendered styles which are particularly difficult to animate by traditional means including the ‘airbrushed’, scraperboard, watercolour, Gouache, ‘ink-wash’, and the ‘crayon’ styles.

We believe our system offers a new fresh perspective on computer aided animation production and associated tools.

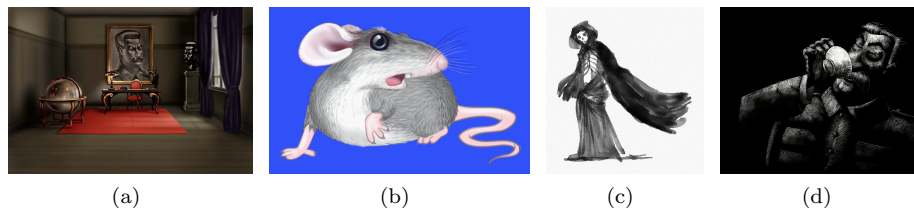
*Keywords:* artist driven, stylised modelling, stylised animation, computer animation, computer-assisted animation, NPR, NPAR

## 1 Introduction

**Motivation.** Although 3D animation is a popular form, because of the indirect nature of the interaction model, many details are extremely hard to construct and animate, while it is much simpler to design very convincing lookalikes in 2D. For example, ask a modeler to make an animation of a walking dinosaur and watch another artist draw a much more fancy 2D version during the time needed to start up the designer’s favourite 3D software.

The difference between 2D and 3D modelling is even more apparent when subtle animation effects (artistic expressions, caricatures, ...) are involved. The stylistic possibilities afforded by 2D animation mean that 2D animations can be rich in a way which is seldom achieved by 3D animations (even with significantly more effort).

It is our contention that this overhead is unnecessary. Figure 1(a) shows a 2D image rendered without the use of a 3D model; it is inexpensively animatable, open to visual modification to suit the animator’s individual style, yet still ‘highly rendered’ in appearance.



**Fig. 1.** Snapshots of highly stylised animations. a) Highly rendered 2D image © Beb Deum/CALON. b) Mouse raising its head in an *airbrushed* style. c) Death in the guise of the ‘Grim Reaper’ in an *Oriental black ink* style. d) Stalin character drinking tea in a *scraperboard* style.

**Contribution.** We present here a system which is an effective 2D alternative to most of NPR research that is focused on frame coherent stylised rendering of 3D models. It supports 2D drawing instead of stylising strokes on 3D geometries. Furthermore, it relies on the simulation of natural materials and processes involved in making brush-strokes.

Our approach gives a highly stylised look to drawn animation but also allows an artist to give a 3D-like look to images yet they still behave as though animated by drawing, which they are. We believe our system offers a new fresh perspective on computer aided animation production and associated tools.

A series of examples are shown in Figure 1. We emphasise that at no time any 3D modelling is performed and in particular no 3D models are used to support the rendering of the images, scenes and animations we show.

**Approach.** Technically the challenge is to achieve stable rendering across frames. Artists find it difficult to achieve temporally coherent lighting and shading effects, potentially resulting in jittery, unpleasant imagery. To achieve such consistency the simulation is associated with the trajectory of the brush stroke itself and these brush trajectories are in-betweened across frames so the artist needs only to paint up a single frame. The brush trajectories are thus in-betweened and the simulation reproduced, possibly requiring some ‘touching up’ in more distant frames. While the method of managing consistency is essentially the same across all simulations, individual simulations may require different controls to manage different evolutions of effects like texture behaviour.

**Paper Organisation.** This paper is structured as follows. Section 2 surveys work we consider related to ours. Section 3 describes the data path to be followed to create highly stylised animations. In Section 4 we show some examples. Finally Section 5 is our conclusions section in which we also discuss our results and set the contexts for future work.

## 2 Related Work

In this section we elaborate on work we consider related to ours.

### 2.1 The Traditional Approach to Drawn Animation

Broadly speaking, traditional animation is defined as a technique in which the illusion of movement is created by depicting a series of individual drawings on successive frames. Unlike live action, where the camera is running continuously, each frame of an animation film is shot one by one. Moreover, as characters are separated into several layers, each single frame might consist of numerous layers stacked up together.

What we give here is an account as it might be in a large industrial studio. The overall work can be thought of as falling into two phases, that of *pre-production or design* (15% of the total effort) and *production* (about 85%). The stages are story development, Leica reel test, scene staging, exposure sheet preparation (which completes *preproduction*); then drawing, line test, ink-and-paint, rostrum camera stage (otherwise known as composition), and sound-track synchronisation or ‘synch’ (which completes *production*).

The drawing process itself is done in three phases: (i) lead animators draw the most significant images, which are referred to as extreme frames or poses, containing the major features of the action; (ii) assistant animators produce key frames between the extreme frames, hence detailing the desired animation action; while (iii) less experienced animators (in-betweeners) are responsible for creating all the remaining in-between frames of the animation, resulting in a smooth sequence of drawings. Drawing these frames is known as ‘in-betweening’ or ‘tweening’.

We refer to Patterson & Willis [1] and Preston Blair [2] for readers interested in an in-depth explanation.

## 2.2 The Use of Computers in Drawn Animation

By far the most common use of computers in drawn animation is in the stage which in a feature film would be referred to as postproduction, namely the inking, painting and compositing of the artwork, as discussed in the previous section. Generally speaking they are not used earlier in the data path although there is a small proportion of studios whose style is well-suited to automatic in-betweening and use it. Software to support artists to produce in-between drawings are becoming more widely used but these just provide on-line support for making the drawings, checking and line testing on the assumption that a sequence of drawings is going to be produced by hand. On the whole, animators do not like automatic in-betweening precisely because they want to break the rules in some way on just about every frame.

**In-betweening.** In-betweening is effected on the basis of one of two models: *shape-based* (e.g., Reeves [3], Sederberg & Greenwood [4], Ranjian [5] and Kort [6]) and *skeleton-based* methods (e.g. Burtnyk & Wein [7], Shapira & Rappoport [8] and Jinhui Yu [9]), which are often used in combination. The skeleton-based methods use an articulated skeletal structure which is normally layered to reflect the order in which the skeleton elements are encountered but may sometimes exceptionally be retained as a 3D entity, then layered on the basis of view-dependency. Each skeletal element in the view-dependent form will have a shape associated with it and this will follow the movements of the skeleton which is defined in terms of the movements of the skeletal joints. There is the issue of the control of the skeleton which can be determined by motion capture or manual posing.

The shapes associated with the skeletal elements can themselves be in-betweened using such techniques as Moving Reference Points (MRPs) [3]. A MRP reflects common animator's practice in selecting a point on the drawing and describing its trajectory in 2D with associated timing. A set of MRPs subdivides a line into segments which in its original formulation implicitly define a 3D patch of which the evolution of the segment is a 2D projection. The segment is initially oriented by the movement of the skeleton, then re-shaped according to the MRP trajectory as mapped into the 2D space defined by the skeleton element itself.

More recently, Kort presented a rule-based method for computer aided in-betweening [6]. The content of each key drawing is analysed and classified into strokes, chains of strokes and relations that hold among them. Rules decide what parts of different drawings may be matched. Finally, generated animation paths between corresponding strokes determine the resulting in-betweens.

**Rotoscoping.** When artwork is animated from film or video footage this is referred to as rotoscoping [10] and typically rotoscoped artwork differs significantly in timing and behaviour from drawn animation. Rotoscoped artwork, however, is 'trapped' by being too realistic: since the underlying outlines are rendered too accurately, a very realistic silhouette is generated which we especially want to avoid.

**Non-photorealistic Animation and Rendering.** It is fair to say that our work intrudes into the area generally known as Non-photorealistic Animation and Rendering (NPAR). We do not build 3D models so we do not render in the conventional sense, and the work of Hays and Essa [11] comes closest to what we do. However, Hays and Essa use photographic images as source material, analyse them down into brush-strokes decorated with parameters which may be interpreted in different ways to achieve different appearances, then use optical flow analysis to determine how to interpolate the brush-strokes. In our approach the artist applies the strokes to obtain the visual effect wanted — which requires a real-time simulation — then an in-between interpolates the stroke trajectories.

While the idea of simulating the physical effects of drawing tools has been presented before [12,13] it is also the case that the stability of the renderings in animation has been identified as a continuing research problem. Existing methods of toon rendering [14,15] and painterly rendering [16,17,18] have to build 3D models or infer them [19]. Such approaches swiftly run into being too ‘3D-ish’: the enforcement of the geometric, illumination and shading rules of the model often give a salient sense of the 3D aspects of a scene inappropriate to the intended staging the animator seeks. Of course a view-dependent layered model [20] could be extracted from an underlying 3D model but that still doesn’t help us when we want the idea of the 3D in an scene to be unobtrusive or even exaggerated. In the end, rendering directly from a drawing means that shading, texturing and lighting (e.g. highlights) are all mixed up together and so are wholly a product of the artist’s vision with an appearance finally determined by the last brush stroke. In 3D they are all separate and for the most part independent, making reworking a technical guessing game. How, for example can one manage the behaviour of the textures if you want them to be part of the animation? With direct control this is possible but models and interpretations of models make the problem far more difficult.

### 2.3 The Limits of Traditional 2D Animation

Our work has been focused on topics which are very hard to impossible for animators to do by hand and are the sorts of things animators would like to have automated. These include highly-rendered strip cartoon (*bande dessinée*) styles with *faux* lighting and shading effects, ‘difficult’ materials which are visually appealing, and some aspects which are currently impossible to handle in animation. These include: avoiding ‘boiling’ or instabilities on highly rendered images, (e.g. smooth airbrush style, *Carte à Gratter Noire*, childrens’ book illustrations, watercolours etc.), and topics like very slow movements and varying perspectives on complex mechanisms which we are not tackling here. In fact these last are best handled using explicit modelling whether or not wholly 3D. Childrens’ books in particular often contain beautiful, highly rendered images which give the story much of their charm but they are often made with materials which are impossible to manage in animation by manual means or are impracticably expensive to animate. The methods we describe here are aimed at making this potential business a practical proposition.

While abandoning 3D models seems a retrograde step [21] in view of the elaborate machinery which 3D graphics and animation provide, our argument is that it is that very machinery which gets in the way when an artist turns to achieve a given look or staging. So far we have focused on giving artists direct access to intuitive, even familiar, methods for deriving the finished look although the extended structures they have built need controls of their own to keep the creative process manageable and this is an on-going task. The examples we show here are all studies carried out for the express purpose of understanding how to provide convenient and productive interface to these techniques and the evaluations of the experiences of creating these examples are themselves data we are currently interpreting. For example, many of the rendering techniques impose secondary patterns in the form of textures and these, too, need to be the subject of animation management processes. For example the *Carte à Gratter Noire* style imposes a texture in the form of the prevailing directions of the scraper tool as it cuts the surface. On a static background they take up orientations which vary only as the notional camera position moves but on a foreground object they will reorient not only to match the virtual camera but also the movements of the foreground character as it is affected by plays of light and shadow. The problem here is not what the textures do in the foreground or the background but what happens where the textures join. One imagines an intermediate area in which the behaviour of the texture is the dominant character and it has to somehow integrate foreground and background. While an artist can decide what to do here no obvious 3D model, as is often invoked to deal with issues of varying lighting in NPAR, comes to mind.

### 3 Data Path of Highly Stylised Drawn Animation

In this section we describe the data path to be followed to create highly stylised animations.

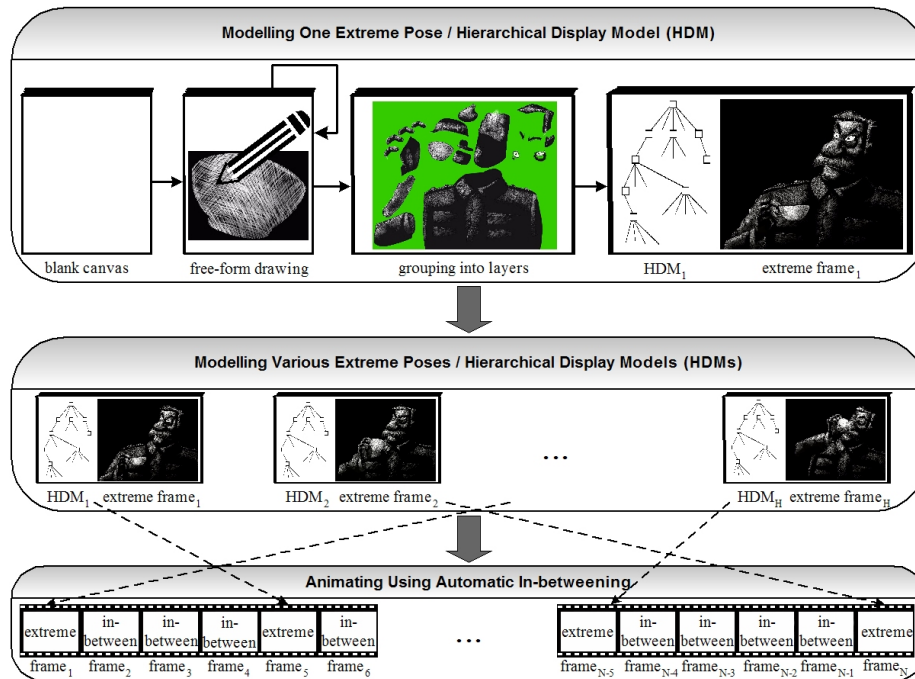
#### 3.1 Overview

Figure 2 depicts a schematic overview of the main parts of the data path.

Starting from a blank canvas, a stylised brush tool (simulating a particular style) is first used to freely draw extreme poses of characters. Each drawing of an extreme pose is built up of a collection of 2D drawn strokes which are manually grouped into layers. Each layer is assigned a place in the Hierarchical Display Model (HDM).

Next, multiple HDMs are created for the same character, each instance corresponding to a specific view (another extreme pose) of the character. A collection of HDMs essentially reintroduces some correspondence information to the 2D drawings needed later for in-betweening and, hence, substitutes for a true 3D model.

Once the extreme poses are properly composed, the animation phase can start. The animator first defines keyframes by specifying extreme poses/HDMs



**Fig. 2.** Schematic overview of the main parts of the data path.

in time. Next, corresponding brush trajectories defined by the HDMs of the keyframes are then in-betweened automatically across intermediate frames. Finally, each HDM of each generated in-between frame is traversed and all elements are drawn one on another from back to front.

The following subsections describe the data path in detail. We start with a description of how to simulate highly rendered styles which are particularly difficult (if not impossible) to animate by traditional means (being traditional 2D handcrafted techniques or traditional 3D animation techniques). Next, the different stages of the data path are elucidated: modelling, manipulating and animating stylised drawings.

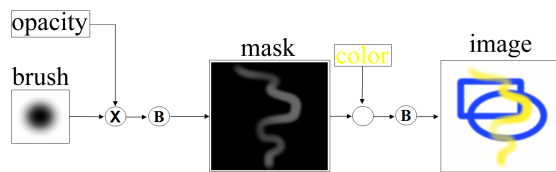
### 3.2 Simulation of Highly Rendered Styles

As we are interested in creating stylised animations, all simulated styles will be based on drawing using a free-form sketching tool [22] which is similar to what is available in any professional-standard vector drawing program.

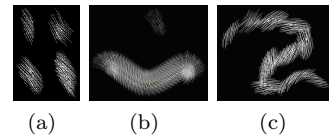
In our system the creation of a stroke is done interactively by sampling a stylus along the trail of the stroke. Our freeform curve model is that of a Bézier chain. However, instead of approximating the Bézier chain by polylines,

purpose-made drawing primitives are employed including paint and air brushes, crayon textures, cross hatched strokes, and pigment particles. All these drawing primitives fully utilise graphics hardware including multi texturing and anti-aliased rendering. This way the artist gets visual feedback immediately. The following subsections give an overview of the realised highly rendered styles.

**Airbrush.** Figure 3 depicts the pipeline of the airbrush tool [23]. The tool creates three objects: (i) a colour layer with (ii) a mask, and (iii) a temporary bitmap layer. Depending on the user input from the canvas, the tool renders the brush into the mask. Circular brushes have the following parameters: radius, opacity, and softness. The mask is a greyscale image used to mask painting. Using this mask, the colour layer is rendered into the working layer. Rendering the colour just copies the colour into the destination layer. Brushes can overlap each other and will create more opaque areas. An airbrush not only releases paint on movement, but at regular intervals while the stylus is down.



**Fig. 3.** Pipeline of the airbrush tool.



**Fig. 4.** a) Pressure sensitive scraperboard paintbrush. b–c) Cross hatched strokes.

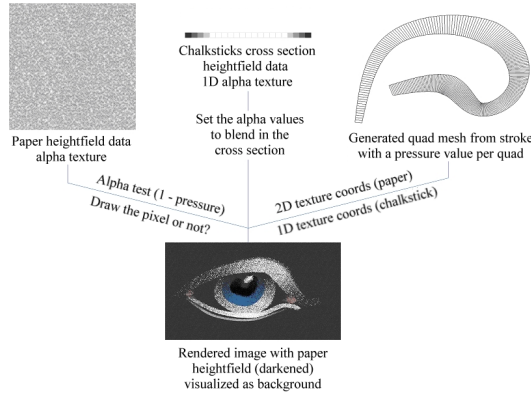
**Scraperboard (*Carte à Gratter Noire*).** For the *Carte à Gratter Noire* style the airbrush tool is adapted. Instead of the circular brush that is procedurally created using the softness and radius parameters, the user is allowed to import a sequence of textured bitmaps which are used as one pressure sensitive *Carte à Gratter Noire* paintbrush (Figure 4(a)). Figure 4(b) illustrates the effect of a spline made out of cross hatched strokes while Figure 4(c) shows how the pressure controls which brush from the image is used. Furthermore, because the brushes are not circular anymore, an option is available, which links the orientation of the brush to the direction of the stroke.

**Crayon.** For the crayon style, a paper height map model [18] is used to represent the rough texture of the paper which is frequently used when drawing with chalk sticks. The chalk sticks themselves are simulated using an 1D alpha texture.

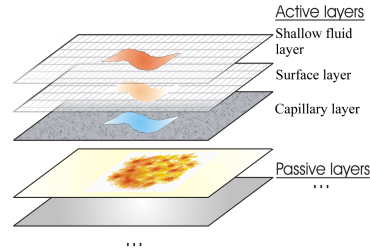
Using the points of the strokes, along with width and pressure values, a mesh is created representing the geometry of the strokes [24]. Using the paper height map and the stroke’s pressure values, the mesh is coloured using the colour of



the chalk stick. Furthermore, an additional 1D alpha texture is used representing the cross section of the chalk stick in order to simulate the soft edges some chalk sticks have (see Figure 5).



**Fig. 5.** Schematic overview of the crayon and chalk stick rendering algorithm.



**Fig. 6.** The canvas model with a layered design.

**Watercolour/Gouache/Ink.** The watercolour painting system employed [25] targets the real-time interactive creation of watercolour images. This is in contrast to most existing work on watercolour applications, which is mostly focused on automatic generation of painterly-style watercolour images from input images. The simulation adopts stable fluid dynamics algorithms to transfer pigment densities and water quantities on top of the canvas. Heuristic rules handle the deposition of pigment within the irregularities of the canvas surface, as well as the evaporation, absorption and capillary diffusion of water inside the canvas structure.

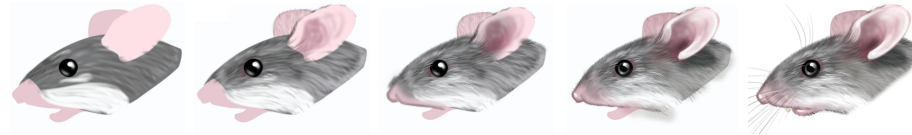
The canvas model has a layered design, consisting of three active layers (shallow, surface and capillary layer) and an unlimited number of passive layers, which are considered to contain previously drawn strokes that have dried and no longer participate in the simulation, except in the final step when the canvas is rendered (Figure 6).

First, some sort of brush puts a mixture of water and pigment onto a paper canvas. At this instant, the paint fluid acts like a flow of water, carrying pigment particles. This ‘fluid body’ is modelled using the *Navier-Stokes* equations. At some point, depending on the paper fabric, the water will be absorbed into the paper and spread throughout the paper structure. As the pigment particles are too large to be absorbed, they will be deposited on the surface and possibly picked up by the paint fluid later on.

Although the brushes and techniques used in Oriental paintings are very different from those in Western painting, the mechanics of pigment and water are quite similar. For Oriental paintings such as shown in Figure 1(c), the canvas is generally more textured and more absorbent, and the dense black carbon particles are smaller and able to diffuse into the paper. The former property is easily expressed in a watercolour simulation by generating a rougher canvas texture and using a higher absorption constant. Despite the fact that the canvas model does not simulate pigment particles inside the canvas structure, ink diffusion can still be handled by the top layer and produce the typical feathery pattern. The palette consists of very dark pigment with high density.

### 3.3 Modelling Stylised Drawings

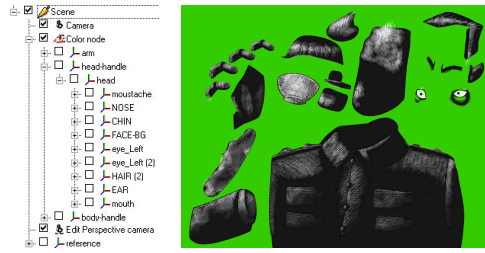
Starting from a blank canvas, the artist starts by using one brush tool (corresponding to one of the styles described in Section 3.2) and freely drawing extreme poses of characters. This corresponds to the drawing of extreme frames by lead animators in traditional animation (see Section 2.1). Each character is built up



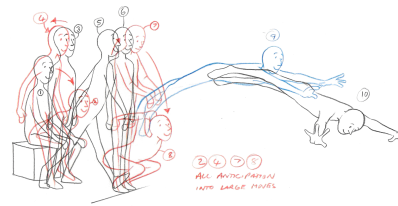
**Fig. 7.** Overview of the drawing process.

of a collection of 2D drawn strokes which are imposed one on another. As is illustrated in Figure 7 this process of drawing characters and objects very much resembles traditional painting: first, a basic (rough) image is drawn, and this is followed by adding detail stroke-by-stroke.

While each stroke could be placed on a single layer it is far more efficient to group these together into a few layers which would tend to be in-betweened together. In fact layers can be grouped together in the usual way in a drawing hierarchy known as an Hierarchical Display Model (HDM). Figure 8 shows a representation for an HDM for the ‘Stalin’ figure in *Carte à Gratter Noire*. If each drawing component is thought of as a leaf element the composition steps are shown on the left. Each node of the HDM is the potential recipient of a channel stream which redefines any parameter that might be there, for example an orientation transformation like rotate about an arbitrary axis or a parameterised warp. Typically an animator will make a series of reference drawings to define the action. An example is shown in Figure 9 below. Note that these are all breakdown poses, at least 4 frames apart so the action covers only 2 seconds. In some cases, particularly for figures and heads, it is desirable to make a series of view-dependent studies, for a standing human figure at  $45^\circ$  angles all round and horizontally,  $\pm 45^\circ$  vertically (and optionally views from above or below).



**Fig. 8.** Stalin drinking a cup of tea. a) Hierarchy Display Model (HDM). b) All separate elements.



**Fig. 9.** Sequence showing a dive off the end of a pool from a sitting position © Les Orton 2005.

Hierarchies for these are all constructed and these give the rendering order for the elements of the character.

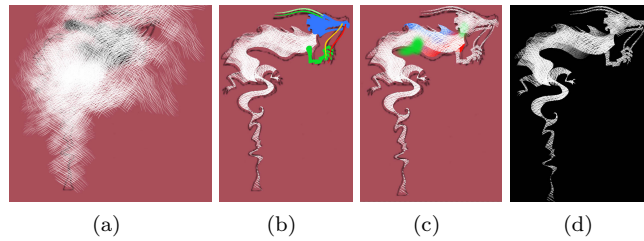
In our system, multiple HDMs for the same character can easily be created by using techniques described in [26] and [6] or simply by altering a duplicate of the original HDM. Each HDM corresponds to a specific view of the character. Consequently, a collection of HDMs for the same character forms the main set of entries of an electronic model sheet for that character and, in effect, substitutes for a true 3D model. Using model sheets like this is usually referred to as 2++D modelling — not 2D but not really 3D either. A collection of HDMs essentially reintroduces some 3D information to the 2D drawings (see Figure 10). Besides



**Fig. 10.** Three HDMs each depicting a specific view of the same character.

brush strokes, other elements can be part of a HDM having its own place in the hierarchy. For example, if lighting and texturing needs to be altered systematically this can be done by adding another layer and managing that layer to provide the desired effect. This will be exemplified by a step by step overview of building up a *Carte à Gratter Noire* version making use of masks and lighting effects. To start the drawing, an area is filled with white strokes (Figure 11(a)). Next, invisibility masks (b) are added to ensure the correctness of the silhouettes. These masks are built up from vectors, without any texture association. Each of these masks will also have its own time line and its independent place in the hierarchy tree. Similar to traditional animation the animator does need to

care herself/himself about lighting and shadowing effects by applying the correct densities to create the desired dark and light regions. In this case the animator added some extra layers (c) to obtain a hazy effect (d). These ‘hazy layers’ separate the different parts and give them a feeling of depth.

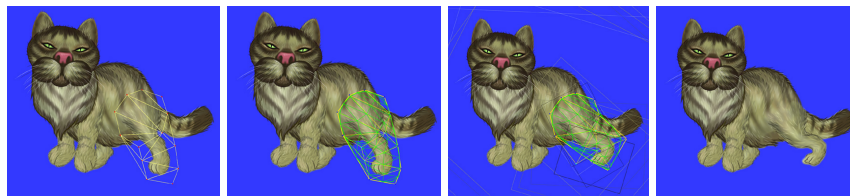


**Fig. 11.** Example of applying masks and defining lighting effects.

### 3.4 Manipulating Stylised Drawings

In this section, we will introduce some tools that enable the animator to manipulate the drawings on a higher level than altering single brush strokes.

We successively implemented a grouping tool, transformation tools, and deformation tools. Each of these tools can operate on both the whole drawing and on a user selected part of the drawing. Existing applications manipulate drawings on a per-pixel basis which results in artifacts because the manipulated parts are cut out and then pasted at a new position. In contrast, due to the use of curve primitives, manipulations can also be performed on parts of the drawn character. In our case the manipulation tools (translate, rotate, scale, deform, . . . ) only affect the control points selected by the grouping tool and so the animator has local control over the drawing while preserving the continuity and connectivity of strokes. As most transformations are rather straightforward to implement,



**Fig. 12.** Subsequent images of a deformation process.

we only highlight the use of locally interpolating subdivision surfaces on model elements in order to realise deformations on the characters to be animated. The

locally interpolating nature allows precise control over the shape of the surfaces: the surfaces are attached to specific parts of the hierarchical models, so the deformation realised only affects the control points of the brush strokes targeted. Figure 12 shows the deformation process in action on an airbrushed cat: (a) positioning of top level subdivision surface grid, (b) an additional subdivision step, (c) deformation of the surface and hence the underlying control points of the targeted brushes (in this case: those of the left hind leg), and (d) the final deformed result.

### 3.5 Animating Stylised Drawings

In the previous sections (Sections 3.3 and 3.4) we focused on the modelling and manipulation of highly stylised drawings. Once the models are properly composed, the animation phase can start:

- (i) the animator has to define keyframes by specifying extreme poses in time;
- (ii) all brush trajectories defined by the keyframes are then in-betweened automatically across intermediate frames, creating intermediate HDMs;
- (iii) all HDMs (extreme as well as intermediate) are traversed and all strokes are drawn one on another from back to front.

Regarding the second step two issues need to be addressed: (i) the correspondence problem should be solved, and (ii) a suitable in-betweening algorithm should be employed. This will be explained in the following subsections.

**Correspondence Problem.** The use of Hierarchical Display Models (HDMs) to represent various instances (i.e. extreme poses) of a particular drawn image also lends itself to easily find a correspondence between them.

Drawing a collection of HDMs, each depicting a different view of the same character, is currently implemented by using similar techniques as described in [26] and [6] or simply by altering a duplicate of the original HDM (see Section 3.3). As a result, two subsequent HDMs share a large number of subparts and attributed elements (e.g., brush strokes), yet have a different drawing order, and so there is a 1-to-1 mapping between these elements. For the strokes which have no correspondences in the next or previous predefined view, fade-in and fade-out is used in a similar matter to Hays [11].

Furthermore, as our freeform curve model to represent brush strokes is that of a Bézier chain, the curve control points are what are interpolated when in-betweening different instances of the same brush stroke.

**In-betweening Algorithm.** As we already have a correspondence between HDMs and a mapping of the shared accompanying elements (e.g., brush stroke parameters), any in-betweening algorithm found in the literature (Section 2.2) could be employed.

In the current system, we use a wholly 2D based ‘Moving Reference Point’-inbetweener [26] (based on Reeves’ MRP-inbetweener [3]) which, however, lacks

the vices that Reeves reports. The Coons patch algorithm used by Reeves sometimes contorts when trying to obey cross-boundary constraints in the absence of twist vectors, producing ‘clicks’ in the behaviour. As a result, interpolated curves sometimes fold over in the interior of the patch. This contortion can only be controlled by specifying an additional straightforward keyframe constraint. However, Reeves pointed out that it was often necessary to specify many more static keyframes than were wanted when using their version of MRP.

In our implementation, the animator can control the interpolation between more than two keyframes with as many moving points as necessary — by default, each control point is treated as a MRP. This means we have a set of 2D patches, each defining the trajectory and timing between two extreme frames. At run-time, interpolated versions of the patches are created so the MRPs do not necessarily follow the exact paths defined by the patches but instead can follow an intermediate path. As a result, our MRPs do not necessarily pass through all defined keyframes and hence ‘clicks’ are eliminated in the animated sequence.

## 4 Examples

In the following subsections several ways to animate are discussed: pose-to-pose animation, step-ahead animation, and performance-driven animation.

### 4.1 Pose-to-pose Animation

Drawing or setting up key poses followed by drawing or creating in-between images is referred to as pose-to-pose animation. This is the basic computer keyframe approach to animation and is excellent for fine-tuning, timing, and planning out the animation ahead of time. First, the animator starts with developing/planning the extreme poses of the characters in the modelling phase. Next, once the animator has created the extreme frames, s/he only has to specify keyframes. Finally, the automatic in-betweening method comes into play and generates the desired animation.

The pictures in Figure 13 display an airbrushed cat starting to run. For the animation of the cat the different extreme frames (about 30) were created using the subdivision free-form deformation tool. Afterwards, pose-to-pose animation was used to in-between these keyframes. The background is a 3D background. For the airbrushed mice flying on a paper airplane (Figure 14) about 15 extreme frames were used. 20 extreme frames were involved to create the Gouache simulation of waving flags as shown in Figure 16.

Note that the complexity of the animation involved determines how many extreme poses have to be provided by the animator and thus how much of the in-betweening is left to the system.

### 4.2 Step-ahead Animation

In straight-ahead animation the animator draws or sets up objects one frame at a time in sequential order until the sequence is complete. In this way there is one

drawing or image per frame that the animator has setup. This approach tends to yield a more creative and fresh look but can be difficult to time correctly, and tweak. Our system supports straight-ahead animation by making use of the subdivision freeform deformation tool (see Section 3.4) which permits the simultaneous control of many MRPs.

Figure 15 shows that it is possible to control the movement of scraperboard rendering both within a shape and around it in a coherent way. For this *Carte à Gratter Noire* animation only step-ahead animation was employed as complex movements are involved: rising smoke transforming itself into an evanescent creature.

### 4.3 Performance-driven Animation

The example shown in Figure 17 depicts an animation of a human face which was integrally driven by externally gathered facial motion data. After processing the facial motion data and generating a HDM for each frame, the animation was rendered in a crayon style. Note that for this animation we deliberately introduced a ‘shower-door’ effect by superimposing a paper-like structure on the canvas. We believe this ‘shower-door’ effect would be the case as well when creating a real physical chalk animation. In addition, the paper texture could be animated as well thereby mitigating the shower-door effect to some extent.

## 5 Conclusions

In this paper we introduced techniques and tools to draw, manipulate and animate new forms of stylised animation in computer assisted animation production. We focused on realising highly rendered styles which are particularly difficult (if not impossible) to animate by traditional means including the ‘airbrushed’ style, the scraperboard (‘scratched card’ or ‘*Carte à Gratter Noire*’) style, the ‘watercolour / gouache’ style, and the ‘crayon’ (chalk) style.

The introduction of the physical simulation of materials without reference to 3D models has many implications for animation. It is apparent that the production values of 3D animation can be approached if desired or that styles utterly unlike 3D, yet relying and retaining complex textures and structures, can be handled quite stably. While this has a direct commercial value, making possible many projects which were not possible before, it has a wider value in not only allowing styles which artists have wanted to use in animation for a long time but also to allow effects deemed impossible before, acetates which ‘take’ watercolour or ink washes, etc. Many strip cartoon styles, which often have quite high-quality artwork in them, imaginatively if unrealistically staged, have been thought unanimatable precisely because of the problem of frame-to-frame stability. No longer.

**Discussion.** Traditional brush strokes are pixel based and so can be applied immediately which is a cheap operation in terms of processing power. Highly

rendered strokes, on the other hand, are based upon curves which are fitted to user input in real-time. As a reference to the geometry of the strokes is stored, they can be animated easily and the brush properties can be changed after they are applied. One issue with highly rendered strokes, however, is the high fill rate which makes the real-time requirement for the drawing process difficult to achieve without graphics acceleration.

The tools and techniques described also create different ways of looking, at the digital tools (created by engineers, but destined to be used by animators, artists and illustrators), at the rigidity of configuration of certain digital ‘traditions’ inherited from 3D (the timeline, for example), as well as at the relevance of certain fundamental working practices that require a rather laborious apprenticeship. Moreover, these techniques also point to abundant and fruitful exchanges between software engineers and artists in an attempt to find solutions to these various problems.

**Future Work.** We have limited ourselves in this paper to the issue of character drawing and backgrounds only where it has been a matter of rendering style and supporting it. Composition is an equally important component of the model and we will be returning to this topic to do it proper justice in due course. The reader should be thinking about simulation of the acetate stack, the separation of colour and illumination, and all the many effects which a Rostrum camera is capable of with real physical materials.

Also to be fair to 3D NPR techniques, there are aspects of lighting and proportion that 3D models can give which are difficult to imagine for non-skilled artists. In the future we want to explore the suitability of a system that combines the 2D capabilities of our system with 3D shading results shown as a visual reference only.

## Acknowledgements

We gratefully express our gratitude to the European Fund for Regional Development (ERDF), the Flemish Government and the Flemish Interdisciplinary institute for Broadband Technology (IBBT), which are kindly funding part of the research at the Expertise Centre for Digital Media. Part of the work is also funded by the European research project IST-2001-37116 ‘CUSTODIEV’. Many thanks go also to Xemi Morales for his artistic contribution.

## References

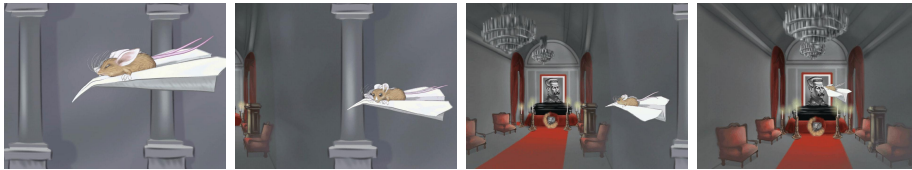
1. Patterson, J.W., Willis, P.J.: Computer assisted animation: 2D or not 2D? *The Computer Journal* **37**(10) (1994) 829–839
2. Blair, P.: *Cartoon Animation*. Walter Foster Publishing Inc., ISBN: 1-56010-084-2 (1994)
3. Reeves, W.: Inbetweening for computer animation utilizing moving point constraints. *Computer Graphics* **15**(3) (1981) 263–269



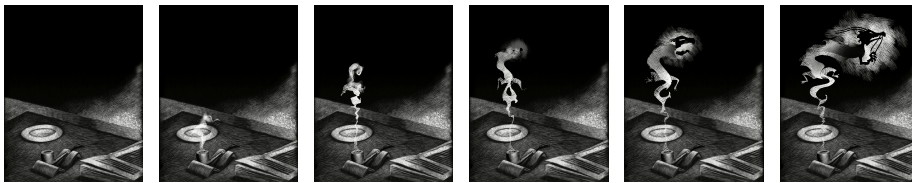
4. Sederberg, T.W., Greenwood, E.: A physically based approach to 2D shape blending. *Computer Graphics* **26** (1992) 25–34
5. Ranjian, V., Fournier, A.: Matching and interpolation of shapes using unions of circles. *Computer Graphics Forum* **15** (1996) 129–142
6. Kort, A.: Computer aided inbetweening. NPAR2002: Symposium on Non-Photorealistic Animation and Rendering (2002) 125–132
7. Burtnyk, N., Wein, M.: Computer-generated key-frame animation. *Journal of the Society Motion Picture and Television Engineers* **8**(3) (1971) 149–153
8. Shapira, M., Rappoport, A.: Shape blending using a star-skeleton representation. *IEEE Computer Graphics and Applications* **15** (1995) 44–51
9. Yu, J., Patterson, J.W.: Object deformation using quaternions. In: *Proceedings of Eurographics UK Chapter 14th Annual Conference*. (1996) 75–88
10. Agarwala, A., Hertzmann, A., Salesin, D., Seitz, S.: Keyframe-based tracking for rotoscoping and animation. In: *Proceedings of SIGGRAPH, ACM* (2004) 584–591
11. Hays, J., Essa, I.: Image and video based painterly animation. NPAR2004: Symposium on Non-Photorealistic Animation and Rendering (2004) 113–120
12. Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes. In: *Proceedings of SIGGRAPH, ACM* (1998) 453–460
13. Hertzmann, A., Perlin, K.: Painterly rendering for video and interaction. NPAR2000: Symposium on Non-Photorealistic Animation and Rendering (2000) 7–12
14. Decaudin, P.: *Modélisation par Fusion de Formes 3D pour la Synthèse d’Images - Rendu de Scènes 3D Imitant le Style “Dessin Animé”*. PhD thesis, Université de Technologie de Compiègne (France) (1996)
15. Lake, A., Marshall, C., Harris, M., Blackstein, M.: Stylized rendering techniques for scalable real-time 3D animation. NPAR2000: Symposium on Non-Photorealistic Animation and Rendering (2000) 13–20
16. Meier, B.J.: Painterly rendering for animation. In: *Proceedings of SIGGRAPH. Volume 25(4)*., ACM (1996) 477–484
17. Daniels, E.: Deep canvas in Disney’s Tarzan. In: *Conference abstracts and applications of SIGGRAPH, New York, NY, USA, ACM Press* (1999) 200
18. Kalnins, R.D., Markosian, L., Meier, B.J., Kowalski, M.A., Lee, J.C., Davidson, P.L., Webb, M., Hughes, J.F., Finkelstein, A.: WYSIWYG NPR: drawing strokes directly on 3D models. In: *Proceedings of SIGGRAPH, ACM* (2002) 755–762
19. Cohen, J.M., Hughes, J.F., Zeleznik, R.C.: Harold: A world made of drawings. NPAR2000: Symposium on Non-Photorealistic Animation and Rendering (2000) 83–90
20. Rademacher, P.: View-dependent geometry. In Rockwood, A., ed.: *Proceedings of SIGGRAPH, Los Angeles, ACM, Addison Wesley Longman* (1999) 439–446
21. Catmull, E.: The problems of computer-assisted animation. In: *Proceedings of SIGGRAPH. Volume 12(3)*., ACM (1978) 348–353
22. Vansichem, G., Wauters, E., Van Reeth, F.: Real-time modeled drawing and manipulation of stylized cartoon characters. In: *Proceedings of the IASTED International Conference on Computer Graphics and Imaging, Honolulu, HI, USA, IASTED* (2001) 44–49
23. Di Fiore, F., Van Reeth, F.: Modelling in 2D enabling fluid stylised animation. In: *Proceedings of GraphiCon, International Conference on Computer Graphics & Vision*. (2003) 124–130
24. Northrup, J.D., Markosian, L.: Artistic silhouettes: A hybrid approach. NPAR2000: Symposium on Non-Photorealistic Animation and Rendering (2000) 31–37
25. Van Laerhoven, T., Van Reeth, F.: Real-time simulation of watery paint. *Computer Animation and Virtual Worlds* (2005) 429–439
26. Flerackers, C.: New algorithms and techniques in 2.5d animation. Master’s thesis, Transnationale Universiteit Limburg (2002)



**Fig. 13.** Airbrush - animation of an airbrushed cat starting to run.



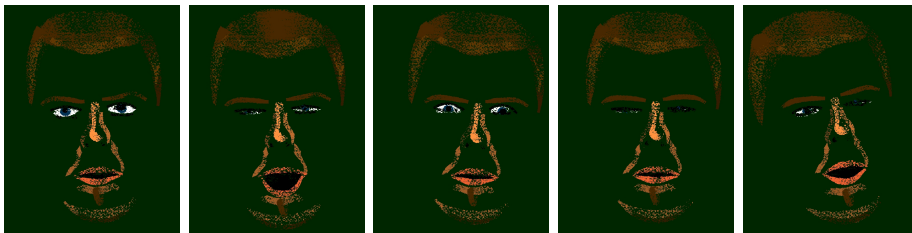
**Fig. 14.** Airbrush - excerpt of a stylised 2D movie.



**Fig. 15.** Scaperboard - rising smoke transforming into an evanescent dragon.



**Fig. 16.** Gouache - excerpt of a sequence depicting waving flags.



**Fig. 17.** Crayon - some snapshots of an animated sequence of a human face.