Volume xx (200y), Number z, pp. 1–10

Real-Time Oil Painting on Mobile Hardware

Tuur Stuyck[1] † Fang Da[2] ‡ Sunil Hadap[3] § Philip Dutré[1] ¶

[1]KU Leuven [2]Columbia University [3]Adobe Research

Figure 1: *Oil painting examples made with our mobile painting system.*

Abstract

This paper presents a realistic digital oil painting system, specifically targeted at the real-time performance on highly resource constrained portable hardware such as tablets and iPads. To effectively use the limited computing power, we develop an efficient adaptation of the Shallow Water Equations that models all the characteristic properties of oil paint. The pigments are stored in a multi-layered structure to model the peculiar nature of pigment mixing in oil paint. The user experience ranges from thick shape-retaining strokes to runny diluted paint that reacts naturally to the gravity set by tablet orientation. Finally, the paint is rendered in real-time using a combination of carefully chosen efficient rendering techniques. The virtual lighting adapts to the tablet orientation, or alternatively, the front-facing camera captures the lighting environment, which leads to a truly immersive user experience. Our proposed features are evaluated via a user study. In our experience, our system enables artists to quickly try out ideas and compositions anywhere when inspiration strikes, in a truly ubiquitous way. They don't need to carry expensive and messy oil paint supplies.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Computer Graphics—Three-Dimensional Graphics and Realism Animation I.3.4 [Computer Utilities]: Paint Systems—

1. Introduction

Oil paint has been a popular medium for artists for several centuries. However, a high fidelity and immersive digital experience with all the nuances and peculiarities of real oil paint remains unattained by digital simulations. Physically accurate simulations are especially challenging on compute constrained mobile hardware such as iPad and Android tablets.

In recent years, tablets have become one of the predominant media consumption devices. Additionally, the platform is gaining popularity as a mainstream ideation tool with the next generation of artists. Our system offers the best of both worlds: a fully expressive digital medium that closely mimics the serendipitous real life oil painting experience and the convenience and ubiquity of mobile digital devices. The users are liberated from the confinement of art studios with expensive and messy oil painting material and can now create artwork in everyday walks of life. We propose our system as a reliable tool to quickly try out ideas and perform preliminary paint studies and to aid artists complimentary to their traditional workflow to obtain faster convergence towards the desired outcome. Artists demonstrated a large interest in such a portable and realistic painting tool, therefore, we target the mobile devices as a primary platform. Even though differences between mobile and desktop hardware are fading, mobile devices will always be more limited by its form factor and restricted battery life. It is therefore imperative to use efficient algorithms in order to minimize excessive battery drain. Additionally, a mobile

---

† tuur.stuyck@cs.kuleuven.be
‡ fang@cs.columbia.edu
§ hadap@adobe.com
¶ philip.dutre@cs.kuleuven.be

implementation allows us to make use of a number of hardware features to provide a more interactive and engaging experience.

Main Contributions

Mobile devices have seen tremendous performance improvements in recent years. Yet the physical dimensions, weight, expected battery life and tight thermal profile still dictate very modest computing power and resources compared to modern desktop machines. Our primary contribution is a very careful selection and adaptation of specific technologies while maintaining a relatively high degree of realism at real-time speed on the target platform. Our contributions improve the digital painting experience as indicated by our user study. In more detail, our contributions are:

- **Fast fluid dynamics for oil paint:** We adapt the Shallow Water Equations to efficiently simulate the overall dynamics of oil paint on mobile hardware. We then augment the fluid dynamics with a simple, scalable, yet effective model to mimic different behaviors ranging from very diluted and runny to thick and seemingly viscoelastic oil paint. We incorporate gravity read out by the device's accelerometer, combined with the effects of surface tension, which enable the user to control the direction of fluid flow.
- **Multi-layer pigment framework:** We propose a multi-layer structure of vertically stacked pigments to effectively model the complex and distinctive nature of folding, marbling and mixing in oil paint (Figure 2). We develop models that capture pigment deposition, advection and mixing effectively.
- **Real-time rendering:** A combination of carefully chosen techniques efficiently render the oil painting in real-time. We present an immersive rendering experience, which makes full use of the tablet's capabilities: a virtual lighting environment is aligned in accordance with the tilt and orientation measured by the sensor on the tablet, whereas the front-facing camera captures the ambient lighting. Together, they cause the details in the paint geometry to intuitively reveal itself leading to a realistic and immersive experience.

2. Related Work

Oil paint simulation: A very good overview of previous work on painting system as well as three new types of painting systems ranging from a fast but inaccurate to a slow but physically correct model is given by Baxter et al. [Bax04]. The first method, named *dAb Paint*, consists of a fast but simple two layer paint model supporting partial drying, translucency, conservation of volume and a bidirectional paint transfer algorithm. A more complex technique, called *Stokes Paint*, is based on the 3D volumetric Stokes equations and enables physically-based interaction with the paint. The final model, called *IMPaSTo* [BWL04], combines the two previous models and exhibits interactive performance on desktop hardware by representing the fluid surface as a height field with pigment concentration and volume stored at every pixel. In addition to the paint simulation, IMPaSTo uses the Kubelka-Munk pigment color model [Kub48] to achieve realistic mixing and compositing of paint colors. Similar to our work, a 2.5D fluid simulation which supports viscous fluid dynamics of paint is presented by Saito in [Sai11].

More recently, Microsoft's commercial package *Fresh Paint* implements convincing oil paint simulation using [CBWG10] and [BG10]. In [OJIN08], the authors present simulated thick oil paint with a two-way transfer model along with a virtual palette knife. A data-driven approach for synthesizing the 6D hand gesture data for users of low quality input devices called *HelpingHand* [LYFD12] and a a data-driven approach for modelling virtual brush strokes named *RealBrush* [LBDF13] is presented by Lu et al. Recently Chen et al. [CKIW15] introduced the Wetbrush model for realistic real-time oil paint simulations on desktop machines with a high-end GPU. In this work, the authors propose a hybrid framework for fluid simulation where the paint is represented by particles near the brush and by a density field otherwise. The system can realistically model sub-pixel paint detail and brush behavior as well as the liquid transfer but it does this at a high computational cost making it unsuitable for current mid- to lower-end desktop or mobile devices.

Watercolor simulation: A method for the automatic simulation of computer-generated Eastern watercolor paint named *MoXi* [CT05] is presented by Chu et al. Their model makes the distinction between three types of paint flow. In the top shallow water layer, pigments and water flow according to the Navier-Stokes equations using a staggered grid [FM96]. In the middle pigment deposition layer, pigments are absorbed into and lifted from the paper. The bottom layer models the diffusion by capillary actions of fluid absorbed into the paper to simulate the backrun effect using cellular automatons. The pigment layers are composited using the Kubelka-Munk equations to accurately compute colors.

Similar to the work of Curtis et al. [CAS*97], a framework for a physically-based interactive paint system that supports the creation of images with watery paint is proposed by Van Laerhoven et al. [VL06]. Instead of a staggered grid, a semi-Lagrangian method [Sta99] is used for better performance. The work focusses on watercolor simulation but the framework is general enough to simulate related media such as gouache and Oriental ink. A method for the simulation of Eastern brushwork is presented by Chu et al. [CT05]. The author proposes a real-time physically-based method for simulating ink dispersion in absorbent paper using a novel fluid flow model based on the lattice Boltzmann equation for simulating percolation in disordered media. To obtain high quality output, the physics simulation is coupled to simple implicit modelling and image-based methods. An efficient procedural method for generating watercolor-like dynamic paint bahavior aimed towards limited compute hardware has recently been proposed by DiVerdi et al. [DKMI13].

Compared to previous work, our method provides an oil painting framework with the following new properties: a more versatile pigment representation and mixing algorithm that allows multiple wet paint layers, flowing paint under varying gravity and interaction with the environment. Our system is computationally efficient and able to recreate a wide range of oil paint styles and behaviors.

3. Overview of Oil Paint Simulation

Physical Properties of Oil Paint

Oil paint is a complex and versatile medium, which can be worked on in a variety of ways. It is a slow drying paint consisting of color pigments suspended in oil. The dynamics of the paint are often

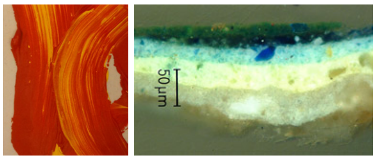

Figure 2: *The distinct nature of viscoelastic paint results in a unique layered pigment structure that expresses itself in marbling and folding of the paint. The left image shows the marbling effect of real paint and the right picture a microscopic cross section showing the layered nature of oil paint pigments (Giovanni Bellini and Titian's 'The Feast of the Gods', 1514 and 1529, National Gallery of Art, Widener Collection © 2006 www.nga.gov).*

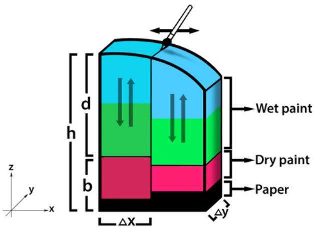

Figure 3: *Example of two cells in which pigments are stored in two stacked layers of active paint. The wet paint sits on top of the dry paint under which the paper texture resides. The fluid height is given by d and the height b is the sum of the paper texture and dried paint height.*

modified by adding thinner fluid. The slow drying property of oil paint allows artists to develop paintings gradually over a prolonged period of time.

The viscoelastic property of oil paint, which is greatly dependent on pigment concentration, is what sets it apart from other paint techniques such as watercolor. The high concentration of suspended pigments in oil results in a viscoelastic fluid with non-Newtonian properties such as shear-thinning. When oil paints are applied the shear created by the brush or roller will allow them to thin and wet out the surface evenly. When the shear is removed, the viscosity returns to normal avoiding drips and runs. The thick oil paint exhibits interesting effects such as preserved brush furrows and overhangs. An artist has the ability to paint over wet paint, without immediately mixing the two strokes completely. This leads to the unique behavior of folding, marbling and interesting mixing of pigments, see Figure 2 and 8.

In comparison, paint mixed with large amounts of thinner has properties similar to watercolor, where the fluid flows freely. Successive thin oil paint strokes blend well, making subtle color variations possible. Flowing paint under gravity occurs naturally and is often used in a technique called underpainting which is said to be one of the most important stages in paintings created by artists like Vermeer [ver]. Additionally, artists often use gravity and runny paint as one of the interesting techniques.

Overview of The Simulation

We represent the oil paint on the canvas as a 2.5D height field as shown in Figure 3. We intentionally sacrifice interesting textural effects such as 3D pointy overhangs and 3D folds, which are sometimes found in impasto style oil paintings, however, this important decision enables us to achieve our primary design goal of real-time simulation on compute constrained mobile hardware. We also prioritize a high lateral (xy-plane) resolution of the simulation to capture intricate brush marks, as opposed to the potentially lower lateral resolution permitted by a full 3D simulation. Various quantities such as fluid height, fluid velocity, and pigments are spatially discretized into cells and are stored in 2D textures. Each cell consists of a fluid volume and several pigment layers. These are the only values that need to be tracked throughout the simulation. The pigment density $\theta_{i,j}$ can be computed as the ratio of the sum of the amounts of pigments over the fluid volume in a cell at position (i,j), this varies for every cell. The density of the paint to be applied can be controlled by the artist by adjusting sliders in the interface. The pigments are suspended in the fluid and are advected by the fluid velocities.

For the fluid dynamics, we use the Shallow Water Equations [LvdP02] as the underlying dynamic formulation. We use these equations because of their efficiency compared to full 3D viscoelastic fluid simulation algorithms. We extend these equations to incorporate the approximate effects of lateral gravity, surface tension, viscoelasticity and viscosity. We use a discrete set of vertically stacked pigment layers in each cell to model the pigment mixing in multiple layers of wet oil paint. Each cell is orthographically rendered using carefully chosen shading models to achieve a characteristic wet and vivid appearance.

4. Brush Model

While any brush model can be used with our system, for simplicity, we use brush stamping. We refer to [LBDF13], [DKH10] and [CBWG10] for some alternatives. Our implementation is based on the paint transfer model detailed in [Bax04] and [BWL04], please refer to these papers for implementation details. In the next paragraphs, we discuss our contributions in order to interact with a multi-layered wet paint structure for pigments.

The brush is represented using two texture maps. The first is a user-chosen grayscale texture map which models the brush shape. Different contours are modelled with different textures and the grayscale value of a pixel represents the scaling factor for the amount of paint deposited. A second texture map represents the brush reservoir and stores the pigments present in the brush tip.

Pigment Mixing in the Brush

The pigments are exchanged bidirectionally between brush and wet paint on the canvas. In our multi-layer implementation, soft strokes interact solely with the top layers whereas pressure causes the brush to penetrate all layers inducing them to participate in the mixed color. The new brush pigments are computed by linearly interpolating the current pigments in the brush with the corresponding pigments on the canvas where the amount of interpolation can be controlled by the artist. The total mix of the pigments in a single cell on the canvas is computed as a weighted sum of the pigments in the separate layers where different types of strokes can be modelled using weights proportional to brush pressure and bristle stiffness. Alternatively, when no pressure information is available, layers are mixed in equal amounts. Lastly, to model the effect of paint pigments in the brush reservoir being drained to the brush tip, the current brush color is linearly mixed with the user-selected brush color to gradually make the brush pigments return to its original color.

5. Simulation

Simply using the standard Shallow Water Equations would not truthfully simulate viscoelastic shape-retaining fluids like oil paint. We propose a number of efficient augmentations to enable the equations to approximately capture a variety of characteristic physical properties of oil paint: viscoelasticity, wet paint dripping under gravity along with surface tension and pigment advection.

Shallow Water Equations

The Shallow Water Equations are used for their efficiency, enabling high resolution simulations on compute constrained hardware. The equations allow simple integration of our proposed augmentations which enable it to recreate a wide variety of fluid behavior based on the viscoelastic properties and gravity as read out by the device's accelerometer. This captures the effect of rotating and vigorously shaking the tablet. Additionally, the equations allow to easily and uniformly combine watercolor and oil paint within the same simulation.

The standard Shallow Water Equations assume that the velocity in the xy-plane is constant with depth, the fluid is inviscid and the volume can be represented by a 2D height field. The frame of reference is attached to the tablet, which is typically held in hand with varying orientation, such that the xy-plane coincides with the tablet surface, and the positive z-direction is pointing out from the tablet. These assumptions allow one to integrate the 3D Navier-Stokes equations in the vertical direction to arrive at the following equations which take a convenient 2D form [LvdP02]:

$$\begin{aligned}
&\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + g_z\frac{\partial h}{\partial x} = 0\\
&\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + g_z\frac{\partial h}{\partial y} = 0\\
&\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}[u(h-b)] + \frac{\partial}{\partial y}[v(h-b)] = 0,
\end{aligned} \tag{1}$$

where (u,v) is the fluid velocity in the xy-plane, h is the fluid height from the frame of reference xy-plane and b is the sum of the height of the paper texture and the dried paint height (Figure 3). Finally, g_z is the gravity assumed to be acting along the negative z-direction. These equations have only $\mathcal{O}(N^2)$ complexity with nominal grid resolution of $N \times N$. We represent (u,v), h and b as 2D textures. Let $\mathbf{g} = (g_x, g_y, g_z)$ be the gravity vector expressed in the simulation's frame of reference. We update the gravity vector at each simulation step by reading the onboard accelerometer. Even though the simulation frame of reference is non-inertial, we ignore the effects of Coriolis forces.

Equation (1) is stiff w.r.t. gravity, so one directly integrates for the height h^{n+1} at time step $n+1$ using implicit backward Euler discretization in time and using semi-Lagrangian advection [Sta99]. Please refer to [LvdP02] for the derivation of the following formulation and details on the numerical integration:

$$\begin{aligned}
h^{n+1}_{i,j} &+ g_z\Delta t^2\Bigg(\frac{b_{i+1,j}-b_{i-1,j}}{2\Delta x}\,\frac{h^{n+1}_{i+1,j}-h^{n+1}_{i-1,j}}{2\Delta x}\\
&\qquad + \frac{b_{i,j+1}-b_{i,j-1}}{2\Delta y}\,\frac{h^{n+1}_{i,j+1}-h^{n+1}_{i,j-1}}{2\Delta y}\Bigg)\\
&- g_z\Delta t^2 d^n_{i,j}\Bigg(\frac{h^{n+1}_{i-1,j}-2h^{n+1}_{i,j}+h^{n+1}_{i+1,j}}{\Delta x^2}\\
&\qquad + \frac{h^{n+1}_{i,j-1}-2h^{n+1}_{i,j}+h^{n+1}_{i,j+1}}{\Delta y^2}\Bigg)\\
&= \tilde{h}^n_{i,j} + \Delta t\left(\tilde{u}^n_{i,j}\frac{b_{i+1,j}-b_{i-1,j}}{2\Delta x} + \tilde{v}^n_{i,j}\frac{b_{i,j+1}-b_{i,j-1}}{2\Delta y}\right)\\
&- \Delta t d^n_{i,j}\left(\frac{\tilde{u}^n_{i+1,j}-\tilde{u}^n_{i-1,j}}{2\Delta x} + \frac{\tilde{v}^n_{i,j+1}-\tilde{v}^n_{i,j-1}}{2\Delta y}\right),
\end{aligned} \tag{2}$$

where (i,j) are the indices of the space discretized fluid cell, $b(i,j)$ is the sum of the paper texture and dried paint height that are fixed in time and $d_{i,j} = h_{i,j} - b_{i,j}$ is the fluid depth (Figure 3). The quantities $\tilde{u}^n_{i,j}$, $\tilde{v}^n_{i,j}$ and $\tilde{h}^n_{i,j}$ are the values at time n traced back to the departure points using the semi-Lagrangian advection method. We use the GPU-implementation-friendly Jacobi iterations algorithm to solve these equations and obtain stable results after four Jacobi iterations. The velocity field (u,v) can be computed by back substituting the integrated height h^{n+1} into Equation (1):

$$\begin{aligned}
\frac{u^{n+1}-\tilde{u}^n}{\Delta t} + g_z\frac{\partial h^{n+1}}{\partial x} &= 0\\
\frac{v^{n+1}-\tilde{v}^n}{\Delta t} + g_z\frac{\partial h^{n+1}}{\partial y} &= 0.
\end{aligned} \tag{3}$$

Viscoelasticity and Viscosity

Although the distinct mechanical behavior of the paint is central to the experience of a digital oil painting system, accurately simulating complex non-Newtonian fluid dynamics in real-time on mobile hardware is simply not feasible given the limited computing power. We provide an efficient approximate model that is simple and tailored to our GPU implementation, thus finding a practical trade-off between physical plausibility and computation time for the purpose of digital oil painting as confirmed by the user study.

Our empirical model is based on observations made on the behavior of oil paint depending on the dilution where we noticed that thicker paint is better able to withstand gravity forces and is more viscous. This simple yet fundamental observation allows us to construct an efficient and visually pleasing and believable (see Section 9)

Figure 4: *Paint runs down caused by tilting the device resulting in dripping. The dripping is not limited to be within stroke boundaries.*

simulation approach for mimicking these distinctive properties of oil paint but it is likely not suitable for other types of viscoelastic fluid simulations.

The actual behavior varies greatly with the pigment density. When few pigments are suspended in a lot of medium, the paint will behave like a standard Newtonian fluid which flows freely. The opposite case occurs when high concentrations of pigments make the paint withstand the gravity forces and act viscous. We choose to model this behavior by smoothly modulating the gravity term and varying the viscosity based on the local pigment density $\theta_{i,j}$ at cell (i,j). We arrive at a new velocity update rule in the same form as before (3), only with a modulated gravity $\bar{g}_z = (1-\Phi(\theta))\,g_z$ which ranges between zero and g_z. Viscosity in the xy-plane is modelled by incorporating $\nu(\theta)\nabla^2 u$ and $\nu(\theta)\nabla^2 v$ in the right-hand side of the x- and y-velocity component computation respectively [CAS*97]. The kinematic viscosity $\nu(\theta)$ is linearly interpolated between ν_{min} and ν_{max} given the pigment density $\alpha \leq \theta \leq \beta$.

We found the following empirical formula for modulating the gravity term which works well (see Section 9):

$$\Phi(\theta) = \begin{cases} 0 & \text{if } \theta < \alpha \\ H[(\theta-\alpha)/(\beta-\alpha)] & \text{if } \alpha \leq \theta \leq \beta \\ 1 & \text{if } \beta < \theta, \end{cases}$$

with $H(t) = -2t^3 + 3t^2$, $0 \leq H(t) \leq 1$ the smooth Hermite interpolation of t between 0 and 1. The user-chosen values α and β (Table 1) determine the viscoelastic behavior of the fluid. For low pigments densities ($\theta < \alpha$) the paint acts as a Newtonian fluid and produces watercolor like behavior with low viscosity. In contrast, when the density is large ($\theta > \beta$) the paint behaves like a shape-retaining material where the effects of gravity are withstood and viscosity is high. For $\alpha \leq \theta \leq \beta$, the gravity term has a limited effect on the simulation.

Dripping Paint Under Gravity

To account for the non-zero components of the gravity vector in the xy-plane, we augment Equation (3) as follows:

$$\begin{aligned} \frac{u^{n+1}-\tilde{u}^n}{\Delta t} + \left(1-\Phi(\theta_{i,j})\right)\left[g_z\frac{\partial h^{n+1}}{\partial x} + g_x\right] &= \nu(\theta_{i,j})\nabla^2 u \\ \frac{v^{n+1}-\tilde{v}^n}{\Delta t} + \left(1-\Phi(\theta_{i,j})\right)\left[g_z\frac{\partial h^{n+1}}{\partial y} + g_y\right] &= \nu(\theta_{i,j})\nabla^2 v. \end{aligned} \tag{4}$$

The viscoelastic Shallow Water Equations with gravity in all three axes are thus described by the combination of Equations (2), (4) and using the modulated gravity term in the height computation. Figure 4 depicts the typical simulated effect of diluted oil paint dripping under gravity which might occur when using the underpainting technique.

Surface Tension

The surface tension of oil paint and the cohesion between the oil paint and the canvas does not allow the fluid to flow freely in the lateral direction. This is true even when the pigment concentration is small and oil paint is runny. To simulate this, we clamp the magnitude of the velocity to zero when it falls bellow a certain threshold. Additionally, we dismiss the gravity force for cells with little amount of fluid (e.g. 1% of the maximum allowable amount).

Pigment Advection

The brush velocity is set as the Dirichlet velocity boundary condition at the fluid surface and the pigments are advected along with the fluid velocities u and v parallel to the canvas using semi-Lagrangian advection. The top of the fluid moves with velocity u and v computed with Equation (4). The bottom of the fluid in contact with the paper remains stationary. The velocities at each pigment layer are computed by linear interpolation between the fluid surface velocity and zero based on fluid depth d and pigment layer n. The pigments are assumed to be stored at the middle of the respective pigment layer. A fixed minimum amount of pigment residue permanently sticks to the paper due to absorption. To model this property, we threshold the minimum amount of pigment in a cell.

Paint Drying

To simulate paint drying, all velocities are set to zero and the paint is moved from the active canvas to the static dried layer (Figure 3). Drying occurs on user request, outside of the main simulation loop.

6. Multi-Layer Pigment Structure

Instead of a detailed, computationally expensive viscoelastic fluid model that computes forces which lead to mixing such as used in [CBP05] and [GBO04], we use a simple method for modelling the characteristic layered structure of oil paint. We store pigments applied with different brush strokes in separate vertically stacked layers. This enables the system to retain information about pigments that are covered by other paint and moving through the paint will stir the bottom pigments upwards causing the paint to mix into a new color.

Pigments are stored in a fixed number of layers where more layers means better physical accuracy but also increased memory and computing power requirements. Figure 3 shows two layers of wet paint being mixed by a brush on top of a layer of dry paint above the paper texture. A practical example is when a color is painted over an existing color, covering the lower pigments. Moving a brush through the paint will cause the pigments to mix with vertically neighboring pigment layers. The multi-layer approach thus makes it possible for previously covered pigments to reemerge.

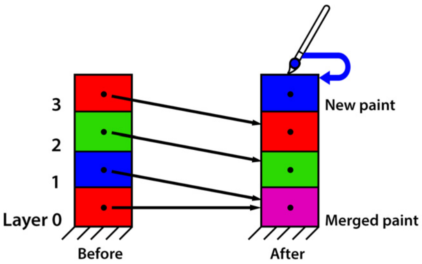

Figure 5: *Example of a single cell in which pigments are stored in four stacked layers. When all the layers are full and a new brush stroke is presented, the lowest two are merged and the rest are moved down, freeing up the top layer in which the brush deposits new pigments.*

Paint Deposition

Distinct brush strokes are stored in separate layers. In order to resolve disparate brush strokes, the cell stores the identification number of the last stroke to update the cell. In our implementation, a brush stroke is defined as one continuous stroke on the screen which starts when a user touches the screen and ends when the touch is lifted. Every time the user touches the screen a new brush stroke is initialized.

When a new stroke is presented to the canvas, the pigments are stored in the lowest free layer. When all available pigment layers in a cell are occupied and a new brush stroke is presented, the two bottom layers are merged together (Figure 5). The contents of layer 1 are added to layer 0 and the rest of the pigments move down one layer to free up the top layer in which the new pigments are stored.

Layer Mixing

Whenever a brush moves through a cell, the pigments in the individual layers linearly mix with the neighboring layers causing the pigments to propagate through the layers. Short excitations will cause the layers to mix partially and repeatedly moving the brush through the paint increases mixing until eventually all layers contain the same pigment mixture.

7. Rendering

We propose a combination of efficient algorithms to render oil paint in real-time. Additionally, we propose a new technique to allow interaction with the physical environment in which the painting on the tablet is viewed. We render these components using the following methods which we each discuss in more detail later in this section:

- **Color** is obtained by compositing the pigment layers using the Kubelka-Munk pigment model [Kub48];
- **Diffuse shading** is efficiently computed using image-based lighting [RH01];
- **Specular shading** is modelled by the Torrance-Sparrow BRDF fitted to measured oil paint reflectance data [SSR*07];
- **Glossy reflections** are added from a video feed encoding the real incident lighting on the tablet directly captured from the front-facing camera.

Kubelka-Munk Pigment Model: Following Curtis et al. [CAS*97], we allow the user to select a color interactively and compute the absorption and scattering values using the inverted Kubelka-Munk (KM) equations. This allows us to use a large variety of different colors without being restricted to physical pigments. Using the KM equations, the different pigment layers can be combined and converted to a RGB color. The KM equations require the depth of the pigment layer. Since we do not store the depth of separate layers, every pigment layer is assumed to have a depth equal to the total fluid depth d divided by the number of layers occupied in that particular cell.

Varying light positions: We allow the user to effortlessly change the virtual light position relative to the canvas based on the tablet's orientation. The artist can move the tablet to reveal details caused by the changing lighting that would otherwise remain unnoticed.

Diffuse: Diffuse lighting is computed using an image-based lighting technique based on a decomposition of the irradiance map in spherical harmonic components [RH01], reducing the environment map to a set of just 9 coefficients.

Specular: The Kubelka-Munk equations assume no interface between the pigments and the surrounding environment and therefore do not model shading effects caused by light rays interacting with the interface. In reality, the dielectric medium causes a fraction of the incident light to be reflected and the remaining light is refracted into the paint. The refracted fraction is modelled by the Kubelka-Munk equations and the reflected light is modelled using the specular Torrance-Sparrow BRDF [SSR*07] for rendering oil paint.

Glossy reflections: To achieve the glossy look of wet paint we use reflection mapping based on Schlick's approximation. A video stream captured by the front-facing camera on the device is used for the reflection map. We assume the view direction is perpendicular to the canvas plane. This produces an immersive experience where different orientations, shapes and colors of physical lights become apparent in the appearance of the digital artwork (Figure 6). The limited field-of-view of the front-facing camera poses no real problems. Most of the paint normals point upwards so most of the normals sample the top of the hemisphere which is adequately captured by the video stream.

8. Implementation and Results

Apart from standard functionalities such as color selection, changing brush size, drying, zooming and loading files, we provide two sliders for selecting the medium and pigment amount in the brush. This decision proved to provide the most control and to be the most understandable for both experienced and novice painters as indicated by our SUS score results (Section 9). To enable artists to print their artwork in high quality on large formats, all user actions are automatically recorded enabling the system to perform the identical sequence of actions offline at any desired resolution.

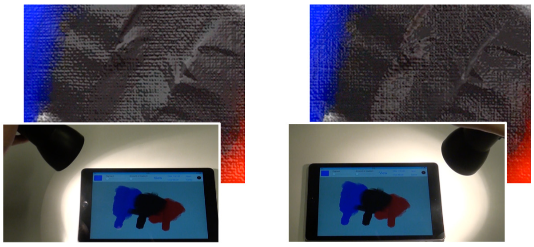

Figure 6: *The reflection map captures the ambient lighting of the physical environment. The top right shows a detail of a valley of paint lit in two different situations. The bottom left inset shows the physical light position relative to the painting.*

8.1. Implementation Details

We have implemented a paint system based on our oil paint simulation model using OpenGL ES 3.0 and the glsl shading language on iOS. Our implementation makes use of multiple target rendering. Apart from user interactions, all operations are implemented in high precision in fragment programs running on the GPU. The simulation quantities are stored in textures which are updated with the results of the fragment operations using ping-pong buffering. Currently, we limit our implementation to three pigments in order to store them and the brush stroke identification value in one RGBA texture for a single pigment layer. The video stream is implemented using a separate OpenGL context to allow writes on textures asynchronous to the main simulation loop.

During the brush stamp shader call, the brush is replenished with the user selected color by 5% and the remaining fraction is computed as an equal linear mix scaled by the brush factor encoded in the stamp brush texture. For shown examples, all pigment layers contributed equally in color mixing. To model the brush drying out, the brush fluid density is drained with a constant amount equal to a small fraction of the maximum brush fluid density when stroking the dry canvas. For the density based viscoelasticity, we selected the values shown in Table 1.

Parameter	**Value**	**Adjustability**
ν	$\in [\nu_{min}, \nu_{max}]$	function of α, β and θ
ν_{min}, ν_{max}	$10^{-6},\ 10^{-2}\ m^2 \cdot s^{-1}$	$\in \mathbb{R}^+$, $\nu_{min} < \nu_{max}$
α	$\gamma \times \frac{\text{maxPigment}}{\text{maxMedium}}$	
β	$\zeta \times \frac{\text{maxPigment}}{\text{maxMedium}}$	
γ, ζ	$0.2,\ 0.8$	$\in [0,1],\ \gamma < \zeta$
maxPigment	1	$\in \mathbb{R}^+$
maxMedium	$0.05\ m$	$\in \mathbb{R}^+$
$(\Delta x, \Delta y)$	$(1/1024,\ 1/768)$	
Δt	$\frac{1}{45}\ s$	$\frac{1}{\text{framerate}}$
$\mathbf{g} = (g_x, g_y, g_z)$	Accelerometer data	$\in [-9.81,\ 9.81\ ]\, m \cdot s^{-2}$

Table 1: *Typical implementation values and ranges.*

Simulation Resolution	**Shader GPU Time**	**Total GPU Time**	**FPS**
2048×1536	33.4 ms	43.8 ms	23
1024×768	8.8 ms	20.8 ms	46
512×384	2.9 ms	16.6 ms	60

Table 2: *Performance measurements for various simulation resolutions measured on iPad Air 2. Render resolution is fixed at* 2048×1536*. The data clearly shows the* $\mathcal{O}(N^2)$ *complexity for the shader time. Doubling the resolution results in roughly four times the amount of work. The total GPU time contains a relatively constant overhead time.*

8.2. Performance

Using two pigment layers in combination with a render resolution of 2048×1536 and with a simulation resolution of 1024×768, we achieve a frame rate of 45 frames per second on iPad Air 2. More performance measurements are given in Table 2. On a first generation iPad Air our implementation with a single pigment layer has a frame rate of 30 frames per second. Note that the CPU is not critical for interactive response. The implementation makes extensive use of multiple target rendering in order to reduce redundant computations and uses the 16-bit GL_HALF_FLOAT data type. This results in just under 19MB of memory usage for storing the entire dynamic simulation state with two pigment layers. Every extra pigment layer requires 6.3MB of additional storage at the current resolution. Figure 7 shows a breakdown of the typical cost of a simulation iteration.

The algorithm is easily parallelizable and computation time scales well with resolution. Compared to full 3D simulation $\mathcal{O}(N^3)$, our system obtains three-dimensional results at the cost of a 2D simulation because of the 2.5D approach which has $\mathcal{O}(N^2 k)$ complexity with k the number of Jacobi iterations. In our simulations four iterations were sufficient, therefore leading to an overall complexity of $\mathcal{O}(N^2)$ (Table 2).

The viscoelastic effect is effectively computed by a cheap function evaluation based on the local pigment density for every fluid cell. With ever increasing computational power our system scales well by providing larger resolutions and frame rates. A desktop implementation would allow for even higher frame rates, more pigment layers, larger painting resolutions and more allocatable computation time for the brush model.

8.3. Paint Behavior and Pigment Mixing

Without a multi-layer structure, pigments painted on top of each other would mix or overwrite, omitting information about the actual layered pigment structure, see Figure 8. Complete pigment mixing results from agitations through brush movement. This is especially true for thick undiluted paints. Our system is the only one that retains the structure of pigments that are covered making mixing at a later time possible, recreating the slow drying property of oil paint when painting wet on wet. In the next section, our user-study shows that artists found our system to produce very believable paint behavior despite the crude viscoelastic approximations made.

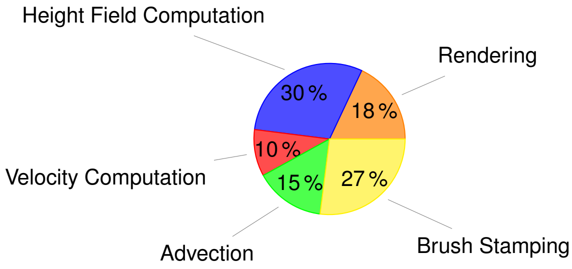

Figure 7: *Typical breakdown of computational cost. This example runs at 46FPS for two pigment layers and four Jacobi iterations for the height field computation.*

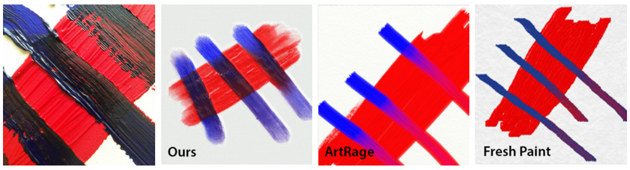

Figure 8: *The left image shows an example of real layered wet paint. Little to no mixing occurs when layering differently colored strokes. Blue pigments are painted over wet red paint diagonally left to right. The second figure shows our multi-layer implementation with two layers. The third and fourth image shows a scene recreated in ArtRage and FreshPaint. Note that our system recreates the physical experiment the most truthfully.*

9. User Study and Comparison with Related Work

To show the importance of our contributions and their effect on the digital painting experience, we performed a user study where we surveyed 13 artists with varying experience. Additionally, we give an objective feature comparison of the current most advanced available software suited for mobile hardware: Fresh Paint [Fre], RealBrush [LBDF13], Procreate [Pro] and ArtRage for iPad [Art] and NVIDIA Dabbler for the NVIDIA Shield tablet [NVi].

User Study Design

During the user study, we offered the artists a version of our applications with the option to disable and enable the proposed features one by one to isolate the effects of individual contributions. We asked all 13 artists to paint a portrait and after 60 minutes of use, they were asked to rate the painting experience using a questionnaire. The questionnaire contains a total of 33 questions to allow evaluation of the application in general, the different features and the perceived usability using the System Usability Scale (SUS) [Bro96]. The questions are scored using an integer Likert scale [Lik32] ranging from 1 (strongly disagree) to 5 (strongly agree). In the next subsection we highlight some of the interesting results.

Study Results

Artists strongly agreed that paint studies are useful and that our proposed system can satisfy this need, see Table 3 (top) for score data of a selection of the questions asked. Both in open-ended questions and in the questionnaire, artists indicated that they found the virtual oil paint believable despite the simplified viscoelastic model. They very much showed their interest in using this application as a tool to perform preliminary paint studies or to paint on the go. Question six shows the high variability between different types of users, not everyone would use our system to finish complete paintings. This is however not unexpected since we propose our system as a tool to try out ideas. The high variability is due to the difference between digital and traditional artist preferences. The ease of use to perform paint studies as well as the portability of the system proved to be one of the main advantages. The biggest downside artists noted is the lack of simulated brush behavior which can be addressed in future work as our model poses no restrictions on the brush model used.

The scores for the different proposed contributions are listed in Table 3 (bottom). The features are ordered in decreasing order of importance according to experienced oil paint artists. From our user study we can conclude that all our proposed features contribute to the digital painting experience. The varying lighting and environment interaction is perceived as a positive contribution to the painting experience but it is somewhat less important then the paint dynamics itself. The use of multiple layers for storing pigments is valued most by artists, even with only two layers as tested in the study. This contribution is closely followed by simulated paint dynamics and varying gravity. Even though we use an approximate viscoelastic model results are convincing as indicated by the responses to question five in Table 3.

	Score	Median
1. I found the experience enjoyable	4.6±0.5	5
2. Testing ideas is helpful	4.1±0.6	4
3. I would use this system for testing ideas	4.2±1.0	4
4. This system would allow me to paint more efficiently	3.8±0.6	4
5. I found the digital oil painting experience believable	4.2±0.5	4
6. I would use this to make complete virtual paintings	3.2 ±1.5	3
Multi-Layer Pigments	4.5±0.5	5
Dynamic Simulation	4.5±0.5	4
Varying Gravity	4.2±1.0	5
Environment Interaction	3.7±0.8	4
Varying Lights	3.6±0.8	3

Table 3: ***Top:*** *General application evaluation average scores with standard deviation and median score. Scores can range between 1 and 5, higher is better.* ***Bottom:*** *Feature scores in descending order of perceived importance.*

Our system obtained a System Usability Score of 85.4/100 with a standard deviation of $\sigma = 7.4$. This score is significantly above the industry average value of 68/100 [BKM09], indicating the high perceived usability of our contributions integrated into painting software.

Feature Comparison with Related Work

We give an objective comparison of the different features offered by state-of-the-art implementations compared to our work.

Fresh Paint Fresh paint has very good brush behavior, color mixing and 2.5D rendering but no simulation based on dilution is provided nor is varying lighting or influence of the environment modeled. The user experience is limited to scumbling paint styles.

RealBrush RealBrush produces very realistic brush strokes. However, there is no physical simulation of the paint since it is a 2D procedurally generated oil paint appearance. The complete stroke shape needs to be known when applying RealBrush making it less suited for a responsive interactive painting program.

Procreate Procreate is a mobile application that mimics the appearance of oil paint by offering a large variety of brush stamp shapes. There is no simulation and the painting is flat 2D without rendering light interactions. Color mixing occurs in an unphysical manner by setting the opacity of brush color.

ArtRage for iPad ArtRage produces nice brush strokes but no interaction with the canvas. The brush color quickly mixes when painting over other wet paint making it difficult to change the color. The application provides static painting and does not offer dilution of the paint nor provides a physical simulation of the medium. It provides nice looking 2.5D rendering but no varying lights or features involving the environment. Only scumbling paint style is supported.

NVIDIA Dabbler This application produces impressive results and provides simulated paint on a tablet device. However, it does so by using a high-end GPU not commonly found on other mobile devices. It does not include multiple pigment layers which has been shown to be essential by our user study and no environment interaction.

While the results of all methods are pleasingly artistic and impressive, apart from our system, none of the above methods provide multiple wet layers or incorporate varying gravity and interaction with the environment and varying lighting based on device orientation. The applications only support a certain paint thickness and this is not user-controllable. Our method supports glazing, wet on/in wet, wet on dry, watercolor effects, scumbling and impasto and can handle the same resolutions as other applications at real-time frame rates. Our study clearly indicates that dynamic simulation, varying gravity and multi-layer pigments are very important to the painting experience. Varying lighting is less crucial in creating better paintings but is still appreciated by most users. A schematic comparison of the discussed applications is given in Table 4.

	Brush	**Simulation**	**Pigments**	**Rendering**
Our System	+*	++	++	++
Fresh Paint	++	+	++	++
RealBrush	++	-	++	++
Procreate	+	-	-	-
ArtRage	+	-	+	+
Dabbler	+	++	+	++

Table 4: *A schematic comparison of the different components of the reviewed mobile applications. * We opted for a simple brush model but our systems poses no restrictions on using more accurate models.*

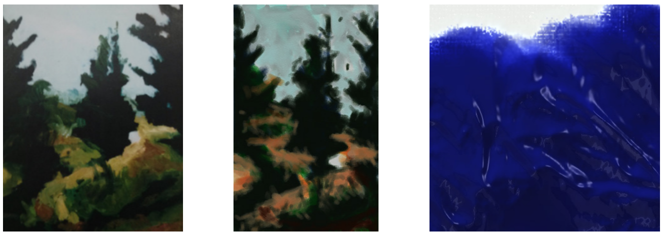
(a) Real painting (b) Digital paintings

Figure 9: *A visual comparison between a real (left) and a digitally created painting (middle) made by professional artist 1. The rightmost image shows an impasto style close-up.*

10. Limitations and Future Work

The relatively small screen size, touch resolution and lack of a digital paint brush equivalent make it hard to create paintings in great detail. In contrast, the ubiquity and mobility make it an ideal ideation platform for experimentation in a way that supports the traditional painting workflow. The equations do not model 3D overhanging paint features but this limitation seems justified when querying artists about the believability of the virtual paint.

The final look of a painting is largely dependent on the natural appearance of the brush strokes. A simulated deforming brush based on applied pressure and brush orientation would be a valuable addition. Volumetric computation of the Kubelka-Munk pigments using spectral lighting [ARC05] to model effects like hue shifts in the paint would lead to more realistic renderings. In addition, finished paintings can be rendered offline at high resolution using a high quality suspended particle ray tracing technique [Jar08]. The front-facing camera feed is used as a reflection map but could also be used to light the painting by decomposing it in its spherical harmonic coefficients and using it as an environment map. Similarly, face tracking can be used to estimate the viewing direction to obtain varying highlights.

11. Conclusion

We present a fully expressive digital paint system, liberating artists from the confinement of their art studios. The proposed system offers natural physical paint behavior to improve the virtual painting experience and has proven to be successful in aiding artists to create new work whether it be digital or traditional. We incorporate gravity based on the device's orientation, along with the effects of surface tension. The proposed fluid simulation equations effectively capture details of oil paint behavior ranging from scumbling to impasto, yet can be solved in real-time on mobile hardware. We propose a new multi-layer pigment representation system to model the layered nature of oil paint. We present an immersive rendering experience which uses the tablet's orientation to reposition virtual lights and the front-facing camera to capture ambient physical lighting to allow interaction with the environment.

The user study shows the added value of our contributions to the digital painting experience. Multiple pigment layers and simulated paint dynamics were evaluated as very important in order to produce better paintings. Adaptive virtual and physical lighting improves the painting experience. Our method is the first to provide multiple separately stored wet layers with varying gravity and interaction with the environment.

We would like to thank all the artists that provided us with excellent feedback and the reviewers for their insightful comments. Tuur Stuyck is funded by the Agency for Innovation by Science and Technology in Flanders (IWT).

References

[ARC05] Abdul-Rahman A., Chen M.: Spectral volume rendering based on the kubelka-munk theory. In *Computer Graphics Forum* (2005), vol. 24, Wiley Online Library, pp. 413–422. 9

[Art] ArtRage: Ambient design, artrage for ipad. `http://www.artrage.com/artrage-ipad/`. 8

[Bax04] Baxter W.: *Physically-based modeling techniques for interactive digital painting*. PhD thesis, Citeseer, 2004. 2, 3

[BG10] Baxter W., Govindaraju N.: Simple data-driven modeling of brushes. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (2010), ACM, pp. 135–142. 2

[BKM09] Bangor A., Kortum P., Miller J.: Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies 4*, 3 (2009), 114–123. 9

[Bro96] Brooke J.: Sus-a quick and dirty usability scale. *Usability evaluation in industry 189*, 194 (1996), 4–7. 8

[BWL04] Baxter W., Wendt J., Lin M. C.: Impasto: a realistic, interactive model for paint. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering* (2004), ACM, pp. 45–148. 2, 3

[CAS*97] Curtis C. J., Anderson S. E., Seims J. E., Fleischer K. W., Salesin D. H.: Computer-generated watercolor. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 421–430. 2, 5, 6

[CBP05] Clavet S., Beaudoin P., Poulin P.: Particle-based viscoelastic fluid simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), ACM, pp. 219–228. 5

[CBWG10] Chu N., Baxter W., Wei L.-Y., Govindaraju N.: Detail-preserving paint modeling for 3d brushes. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (2010), ACM, pp. 27–34. 2, 3

[CKIW15] Chen Z., Kim B., Ito D., Wang H.: Wetbrush: Gpu-based 3d painting simulation at the bristle level. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 200. 2

[CT05] Chu N. S.-H., Tai C.-L.: Moxi: real-time ink dispersion in absorbent paper. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 504–511. 2

[DKH10] DiVerdi S., Krishnaswamy A., Hadap S.: Industrial-strength painting with a virtual bristle brush. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology* (2010), ACM, pp. 119–126. 3

[DKMI13] DiVerdi S., Krishnaswamy A., Mech R., Ito D.: Painting with polygons: A procedural watercolor engine. *Visualization and Computer Graphics, IEEE Transactions on 19*, 5 (2013), 723–735. 2

[FM96] Foster N., Metaxas D.: Realistic animation of liquids. *Graphical models and image processing 58*, 5 (1996), 471–483. 2

[Fre] FreshPaint: Microsoft corporation, fresh paint. `http://www.microsoft.com/en-us/freshpaint/default.html`. 8

[GBO04] Goktekin T. G., Bargteil A. W., O'Brien J. F.: A method for animating viscoelastic fluids. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 463–468. 5

[Jar08] Jarosz W.: *Efficient Monte Carlo methods for light transport in scattering media*. ProQuest, 2008. 9

[Kub48] Kubelka P.: New contributions to the optics of intensely light-scattering materials. part i. *JOSA 38*, 5 (1948), 448–448. 2, 6

[LBDF13] Lu J., Barnes C., DiVerdi S., Finkelstein A.: Realbrush: painting with examples of physical media. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 117. 2, 3, 8

[Lik32] Likert R.: A technique for the measurement of attitudes. *Archives of psychology* (1932). 8

[LvdP02] Layton A. T., van de Panne M.: A numerically efficient and stable algorithm for animating water waves. *The Visual Computer 18*, 1 (2002), 41–53. 3, 4

[LYFD12] Lu J., Yu F., Finkelstein A., DiVerdi S.: HelpingHand: Example-based stroke stylization. In *ACM Transactions on Graphics (Proc. SIGGRAPH)* (Aug. 2012), vol. 31, pp. 46:1–46:10. 2

[NVi] NVidia: Nvidia dabbler. `https://shield.nvidia.com/`. 8

[OJIN08] Okaichi N., Johan H., Imagire T., Nishita T.: A virtual painting knife. *The Visual Computer 24*, 7-9 (2008), 753–763. 2

[Pro] Procreate: Savage interactive., procreate. `http://procreate.si/`. 8

[RH01] Ramamoorthi R., Hanrahan P.: An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 497–500. 6

[Sai11] Saito S.: Paint model having sense of material on computer. *ITE-HI 35*, 16 (2011), 87–91. 2

[SSR*07] Sun B., Sunkavalli K., Ramamoorthi R., Belhumeur P. N., Nayar S. K.: Time-varying brdfs. *Visualization and Computer Graphics, IEEE Transactions on 13*, 3 (2007), 595–609. 6

[Sta99] Stam J.: Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 121–128. 2, 4

[ver] Vermeer: "dead coloring" or underpainting. `http://www.essentialvermeer.com/technique/technique_underpainting.html`. 3

[VL06] Van Laerhoven T.: An extensible simulation framework supporting physically-based interactive painting. 2