

A Generic Pigment Model for Digital Painting

Songhua Xu^{a,b,c,†} Haisheng Tan^c Xiantao Jiao^{a,c} Francis C.M. Lau^c Yunhe Pan^a

^a CAD & CG State Key Lab of China, Zhejiang University, Hangzhou, Zhejiang, P.R. China, 310027

^b Department of Computer Science, Yale University, New Haven, Connecticut, USA, 06520

^c Department of Computer Science, The University of Hong Kong, Hong Kong, P.R. China

Abstract

We propose a generic pigment model suitable for digital painting in a wide range of genres including traditional Chinese painting and water-based painting. The model embodies a simulation of the pigment-water solution and its interaction with the brush and the paper at the level of pigment particles; such a level of detail is needed for achieving highly intricate effects by the artist. The simulation covers pigment diffusion and sorption processes at the paper surface, and aspects of pigment particle deposition on the paper. We follow rules and formulations from quantitative studies of adsorption and diffusion processes in surface chemistry and the textile industry. The result is a pigment model that spans a continuum from the very wet to the very dry brush stroke effects. We also propose a new pigment mixing method based on machine learning techniques to emulate pigment mixing in real life as well as to support the creation of new artificial pigments. To experiment with the proposed model, we embedded the model in a sophisticated digital brush system. The combined system exhibits interactive speed on a modest PC platform. <http://www.cs.hku.hk/~songhua/pigment> provides supplementary materials for this paper.

Categories and Subject Descriptors (according to ACM CCS): I.3.4 [Graphics Utilities]: Paint systems; I.3.5 [Computational Geometry and Object Modeling]: Physically based modeling

1. Introduction

A pigment model is a vitally important component of a digital painting system. Traditional pigment models are mostly built on fluid dynamics or its variations, which assume that pigment behavior is largely due to the autonomous flow of the *pigment-water solution* (PWS). One can simulate such a model based on the self-motion of water and pigment particles. Two representative pieces of work in this category are the watercolor model in [CAS*97] and the ink dispersion model in [CT05]. The former is based on the shallow water model, and the latter on the modified lattice Boltzmann equation. Both consider the PWS's flow being subjected to external forces, resulting in very high-quality "wet brush" effects. For highly viscous pigment solutions, the model proposed in [BWL04], which traces pigment advection through 3D incompressible Stokes equations, can produce highly impressive web brush effects. We find however that these exist-

ent models and other similar ones have only limited ability to emulate the "dry brush stroke" which is extremely important in both Western and Oriental painting. In this paper we propose a new physically based pigment model which tries to capture faithfully the behavior of the PWS over the continuum of a very wet to a very dry state of a brush.

1.1. Main ideas

Our new pigment behavior model can support both dry and wet painting effects or anything in between. One should note that pigment behavior consists of much more than just the spontaneous flow of the solution. Our physically based simulation deals also with the adsorption and desorption processes of the PWS as well as its diffusion process against the leaky brush hair and the paper fibers[†]. This deviates from

[†] Contact him at songhua.xu@yale.edu.

[†] For simplicity, we do not differentiate between adsorption and desorption since one is the reverse process of the other, and refer to either of them as sorption process.

```

proc MainLoop():
  for each time step do:
    1. Transfer PWS between contacting brush and paper
       regions or palette region (Sec. 3)
    2. Diffuse PWS on the paper surface (Sec. 4)
    3. Simulate water evaporation (Sec. 5)
    4. Absorb PWS into paper fibers (Sec. 6)
    5. Render current painting simulation result (Sec. 7)
  end for
end proc

```

Figure 1: Steps of pigment behavior simulation.

traditional flow-based approaches [CAS*97, CT04, CT05] which focus mainly on the PWS's advection process. Our design makes use of relevant results in surface chemistry engineering [AG97, McC01] and the textile and petrochemical industries [McG74, Rut84]. According to surface chemistry, unlike liquids, solid phase solutions cannot easily expand their surface area to reduce free energy on the surface. Therefore, due to the high surface tension, solid phase materials have a strong adsorption or desorption tendency with any contacting external objects. So when the PWS touches the paper surface, adsorption takes place, and for any flow that does happen, the adsorption process is more significant than the flowing/advection process. The variation in adsorption ability across different areas of the paper caused by the non-uniform distribution of the paper fibers is an important contributor to the character of the resultant painting. In a nutshell, our approach is sorption and diffusion based, which offers a closer resemblance to the true physical state of a real brush than traditional flow based models over the continuum of wet to dry brush strokes.

To achieve greater realism, we carefully consider the coupling force and mutual influence between different materials both within the PWS and between the PWS and the paper fibers at various stages of the simulation.

To achieve fast execution, we try to take advantage of known analytic solutions applicable to our formulation. We are thus able to avoid much discrete numerical simulation. Unlike many others who model the paper as a number of layers, e.g., [BWL04], we classify all the particles in the PWS as being in either the mobile state or the immobile state—that is, only two layers (denoted by superscripts *mo* and *im* respectively). Thus, depending on the relative concentrations of pigment particles in the fixed and mobile states respectively, a continuum from very wet to very dry layers can be concisely represented, as opposed to a fixed number of layers, during algorithm design time.

1.2. Pigment model and the brush

Going hand in hand with the pigment behavior simulation is the brush model. Strassmann [Str86] pioneered the re-

search on e-brush. A number of follow-up models were later proposed—e.g., Wong and Ip's model for Chinese calligraphy writing [WI00], Baxter et al.'s model for oil painting [BSLM01], and Xu et al.'s model [XLTP03] and Chu and Tai's model [CT04] for Oriental painting. Despite these efforts, the existent work offers only a loose coupling between the brush model and the pigment model. In both [BSLM01, Bax04] and [CT04]'s pigment models, the brush is used as a tool to deposit pigments onto the paper. Other models are also quite simplistic on this particular issue. We believe that the interaction between the pigment model and the brush is highly intricate, and the intricacies should be carefully considered in order to meet the demand for high expressiveness.

We adopt the brush model in [XLTP03] which is physically based. In that model, the geometry representation for a brush tip bundle is created through the general sweeping operation in CAD by sweeping a variable ellipse along the trajectory of the brush. The brush dynamics are simulated through a two-staged process. In the first stage, the skeleton of the brush tip bundle is deformed according to both the external forces exerted onto it and its collision with the canvas; in the second stage, the brush surface deforms following the deformation of the skeleton. The split of the brush head is supported at the geometry model level and is caused by an estimated internal tension force distribution inside the brush tip bundle. This pressure term is very useful when integrating the brush model with our pigment model as our proposed pigment behavior simulation needs to carefully consider the pressure at the contact point between the brush and the paper. Furthermore, having explicitly the geometry of the split brush bundle (rather than some image-based shortcut) makes physically-based simulation of the pigment behaviors on a split brush head possible and not too tricky to implement.

1.3. Organization of the paper

Sec. 2 discusses related work. Sec. 3 explains the PWS transfer between the brush and the paper surface. Sec. 4 discusses PWS diffusion on the paper surface. Sec. 5 explains how evaporation is handled. Sec. 6 covers how the PWS is deposited on the paper fibers. The rendering process is presented in Sec. 7 which also presents a novel method for pigment mixing. We present the experimental results in Sec. 8. Sec. 9 concludes the paper.

2. Previous Work

Pham [Pha91] was the first to generate brush strokes having different shading, scratchiness and spreading effects; it was done via a variable offset approximation of B-splines. Cockshott et al. [CPE92] suggested a “wet and sticky” model for simulating textured shiny paint using bump mapping and illumination models. Both methods are not physically based—they produce various paintbrush effects without necessarily following the laws of physics. On the other end,

Guo and Kunii [GK91] proposed a diffuse paint behavior model based on analysis of the paper structure which produces singularities in intensity for the diffuse ink painting process. Since then, appearance centric and physically based methods represent two parallel threads in the development of paint brush effects by the computer. Kunii et al. [KNH95, KNV01] suggested a phenomenological model for simulating the “initial zone–black border–gray zone” distribution of intensities in diffuse ink painting. Their simulation is based on highly simplified diffusion equations derived from observations of the real painting process. Such a simplification was needed because of the limited computing power available then. As a result, only blurry images can be generated, which lack any flow pattern. What is interesting is that their method falls between purely physically based and purely appearance centric methods, which is still a useful reference today when designing efficient and quality simulation models.

Benefitting from the abundance of computing power, recent research favored more physically based simulation following first-principle physics laws. On this track, Small [Sma91] pioneered computer simulation of watercolor painting. Curtis et al. [CAS*97] then significantly advanced the watercolor simulation technique using a shallow-water based model; their method relies on solving numerically shallow-water equations, which is too slow for interactive painting even with powerful PC. Beside watercolor, oriental ink based painting simulation also attracted much research endeavors. Lee’s [Lee01] was the first paper where an ink model was used together with a brush model [Lee97, Lee99] to mimic realistic black ink diffusion effects. Their extended algorithm can produce very impressive effects [GK03] and can also be used for calligraphy [Guo95]. Zhang et al. [ZST*99] proposed a simple cellular automaton-based model for capturing black ink painting behavior in Sui-bokuga painting, which they successfully applied to render painter-style 3D trees. Yu et al. [YLLC03] expanded this approach to support more Oriental painting styles using a local equilibrium model. Lin and Shih [LS04] simulated Chinese color ink painting based on phenomenological rules for the ink diffusion process which they obtained by observation. Their painting can be carried out on any simple device such as a tablet PC. Other than watercolor and Oriental ink painting, oil painting simulation is another active field. A representative work is that by Baxter et al. which produced impasto effects for oil or acrylic painting [BWL04]. Other types of painting being were simulated include Dave et al.’s [RMN03] who studied the problem of wax crayon painting and obtained some very interesting results. But despite all that has been done in the field, support for the dry brush stroke effect yet remains very limited.

Along another line, Laerhoven et al. proposed a distributed paper model where the paper is divided into a grid of subpapers for concurrent evaluation through remote parallel processes [LLR04, LR05]. In their model, a procedural

texture creation technique based on the cellular texture basis function suggested in [Wor96] was employed to enhance the realism of painting. They also introduced textured tissues to remove pigments and water as a novel interactive device for the artist [BLR06]. In general, we feel applying parallel computing techniques in first principle physically-based simulation for producing realistic paint effects is a promising area for more future work. On the other aspect, it is also interesting to notice that Xu et al. proposed a single stroke appearance model for capturing stroke texture in Chinese paintings using a parametric approach [XXX*06].

Comparison with Chu and Tai’s work [CT04, CT05]

Chu and Tai proposed an ink dispersion model [CT05] for painting on absorbent paper based on modified Lattice Boltzmann Equations (LBE). The model can produce realistic wet Oriental paintings. LBE however is not well suited for simulating dry brush effects or brush effects arising from a rich variation of pigments and water concentration, due to the theoretical limit in simulation capability of LBE. To support dry brush painting, they had to use an image based method [CT04] which might not produce the most accurate results in response to the user’s inputs. Finally, because there are separate mechanisms for dry brush strokes and wet brush strokes, frequent switching between the two models is necessary during painting, and generation of some in-between effects becomes difficult.

3. Pigment Sorption between the Brush and the Paper Surface

When the brush touches the paper, transfer of the PWS takes place, the direction of which depends on the pigment concentration on either side, i.e., the brush surface in contact and the paper. This is a sorption process between the two media. We choose the Dubinin-Radushkevich isotherm equation [Mis69] to model the process because it considers the influence of pressure:

$$\Delta\rho = \rho e^{-\left(\frac{\kappa_G T}{\beta E_0} \ln \frac{P_{max}}{P}\right)^2}, \quad (1)$$

where ρ is the pigment concentration in the source media; $\Delta\rho$ is the pigment concentration to be sorbed in the process; κ_G is the gas constant, i.e., $8.314 \text{ J}/(\text{mol} \cdot \text{K})$; T is the temperature, which is set to 300K, a typical room temperature; β is the affinity coefficient characterizing the polarizability of the adsorbate; E_0 is the adsorption characteristic energy of the adsorbent, which is mainly affected by the pore density in the adsorbent and assumed to be proportional to the paper fiber density η ; P_{max} is a pressure response coefficient for the pigment; and P is the local contacting pressure between the brush and the paper. $\ln \frac{P_{max}}{P}$ is a term reflecting the impact of brush pressure on the PWS sorption process. Note that the equation describes an equilibrium state. Because such a sorption process usually happens in a flash, for efficiency, we adopt this equation but applied to it a simple scaling (to

be explained at the end of this section), rather than to work out a series of intermediate results over some fine time steps.

Put (1) into the context of our simulation, we have:

$$\begin{cases} \Delta\rho_{x,y}^{mo,t}(i) = \gamma_{x,y}^t(i)\rho_{x,y}^{bru,t}(i)e^{-\left(\frac{\kappa_{GT}}{\beta(i)E_0}\ln\frac{P_{x,y}^{max}}{P_{x,y}^t}\right)^2} \\ \quad \text{if } \rho_{x,y}^{bru,t}(i) > \rho_{x,y}^{mo,t}(i); \\ \Delta\rho_{x,y}^{bru,t}(i) = \gamma_{x,y}^t(i)\rho_{x,y}^{mo,t}(i)e^{-\left(\frac{\kappa_{GT}}{\beta(i)E_0}\ln\frac{P_{x,y}^{max}}{P_{x,y}^t}\right)^2} \\ \quad \text{otherwise.} \end{cases} \quad (2)$$

Here $P_{x,y}^t$ is the contacting pressure between the brush and the paper surface at the location of the paper site (x,y) at time t ; $\rho_{x,y}^{mo,t}(i)$ is the concentration of the i -th pigment which is in mobile state at site (x,y) of the paper at time t and $\rho_{x,y}^{bru,t}(i)$ is the i -th pigment concentration on the brush surface which contacts the paper site (x,y) at time t . $\gamma_{x,y}^t(i)$ is a randomized term which is defined as:

$$\begin{cases} Prob[\gamma_{x,y}^t(i) = 1] \\ \quad = \min(1, \kappa_{bru}|\rho_{x,y}^{bru,t}(i) - \rho_{x,y}^{mo,t}(i)|\eta_{x,y}) \\ Prob[\gamma_{x,y}^t(i) = 0] \\ \quad = 1 - \min(1, \kappa_{bru}|\rho_{x,y}^{bru,t}(i) - \rho_{x,y}^{mo,t}(i)|\eta_{x,y}) \end{cases}, \quad (3)$$

in which $\eta_{x,y}$ is the local paper fiber density and κ_{bru} is a scaling coefficient to produce a suitable match between the magnitude of $|\rho_{x,y}^{bru,t}(i) - \rho_{x,y}^{mo,t}(i)|$ and $\eta_{x,y}$. The reason for introducing a randomization term in the above equation is that such an equation is used to describe the macro behavior of the sorption process; however, here we are doing per pixel location particle behavior simulation in a micro scale; thus adding some random deviation would make the simulation resemble more the real situation.

We also employ the following equation suggested in [Tie94] for the conservation of the pigment mass in the sorption process:

$$|\mathbf{v}_{x,y}^{bru,t}| \Delta\rho_{x,y}^{bru,t}(i) + \eta_{x,y} \Delta\rho_{x,y}^{mo,t}(i) = 0, \quad (4)$$

where $\mathbf{v}_{x,y}^{bru,t}$ denotes the local velocity of the brush tip when in contact with the paper site (x,y) at time t ; this is an input from the hosting e-brush system for the pigment model. Recall $\eta_{x,y}$ is the paper fiber density at the site (x,y) .

To prevent oscillation during the simulated sorption process, we derive the limit on the amount of pigment transferable in the process:

$$|\Delta\rho_{x,y}^{bru,t}(i)| + |\Delta\rho_{x,y}^{mo,t}(i)| = |\rho_{x,y}^{bru,t}(i) - \rho_{x,y}^{mo,t}(i)|. \quad (5)$$

Without loss of generality, we assume $\rho_{x,y}^{bru,t}(i) > \rho_{x,y}^{mo,t}(i)$. Solving the pair of equations (4) and (5) gives the upper bound for $\Delta\rho_{x,y}^{mo,t}(i)$: $\frac{|\rho_{x,y}^{bru,t}(i) - \rho_{x,y}^{mo,t}(i)|}{\eta_{x,y}/|\mathbf{v}_{x,y}^t| + 1}$. Incorporating this bound, we can modify (2) to be

$$\tilde{\Delta\rho}_{x,y}^{mo,t}(i) = \min(\Delta\rho_{x,y}^{mo,t}(i), \frac{|\rho_{x,y}^{bru,t}(i) - \rho_{x,y}^{mo,t}(i)|}{\eta_{x,y}/|\mathbf{v}_{x,y}^t| + 1}). \quad (6)$$

We also consider the mutual influence between multiple pigments in the sorption process, i.e.:

$$\tilde{\Delta\rho}_{x,y}^{mo,t}(i) \kappa_{mutual} \tilde{\Delta\rho}_{x,y}^{mo,t}(i) + (1 - \kappa_{mutual}) \frac{\sum_{j=1, j \neq i}^{\xi} \tilde{\Delta\rho}_{x,y}^{mo,t}(j)}{\xi - 1}. \quad (7)$$

Here κ_{mutual} is a correlation coefficient depicting the mutual influence between the sorptions of different pigments in the PWS, and ξ is the number of pigment species, including water and glue. Once $\tilde{\Delta\rho}_{x,y}^{mo,t}(i)$ is known, we can also update $\rho_{x,y}^{bru,t+\Delta t}(i)$ with the relationship revealed by (4), i.e.: $\rho_{x,y}^{bru,t+\Delta t}(i) = \rho_{x,y}^{bru,t}(i) - \frac{\eta_{x,y}}{|\mathbf{v}_{x,y}^t|} \tilde{\Delta\rho}_{x,y}^{mo,t}(i)$. For the case of $\rho_{x,y}^{bru,t}(i) < \rho_{x,y}^{mo,t}(i)$, a similar treatment is applied. The only difference is that we derive $\tilde{\Delta\rho}_{x,y}^{bru,t}(i)$, $\tilde{\Delta\rho}_{x,y}^{bru,t}(i)$ instead of $\tilde{\Delta\rho}_{x,y}^{mo,t}(i)$, $\tilde{\Delta\rho}_{x,y}^{mo,t}(i)$ through a variant of (6) and (7).

Finally, we apply a simplifying treatment to scale the amount of sorbed concentration $\tilde{\Delta\rho}$ to account for the sorption time. We model the sorption speed, $v_{sorption}(t)$, as an exponentially decreasing variable, which is true of most natural sorption processes; i.e., $v_{sorption}(t) = \kappa_v e^{-t}$ where κ_v is a sorption speed parameter. We also denote the period of time needed to reach the equilibrium state as χ . That is, it is assumed $\int_{t=0}^{\chi} v_{sorption}(t) dt = \tilde{\Delta\rho}$, from which we can get the relationship $\tilde{\Delta\rho}(t) = \min(\frac{1-e^{-t}}{1-e^{-\chi}}, 1) \tilde{\Delta\rho}$. Since our simulation time step is Δt , substituting the corresponding variables into the equation, we have $\rho_{x,y}^{mo,t+\Delta t} = \rho_{x,y}^{mo,t} + \tilde{\Delta\rho}_{x,y}^{mo,t}(i)$ where $\tilde{\Delta\rho}_{x,y}^{mo,t}(i) = \min(\frac{1-e^{-\Delta t}}{1-e^{-\chi}}, 1) \tilde{\Delta\rho}_{x,y}^{mo,t}(i)$.

4. Pigment Diffusion on the Paper Surface

To simulate pigment behavior on the surface of a porous paper, we assume a diffusion process. We choose the following advection diffusion equation [SW04] since it considers pigment advection in the midst of diffusion, and is thus capable of supporting both dry and wet brushing.

$$\frac{\partial \rho}{\partial t} = \kappa_d \nabla^2 \rho - \mathbf{v} \cdot \nabla \rho, \quad (8)$$

where ρ is the pigment concentration in the adsorbate solution, κ_d is the diffusion coefficient, and \mathbf{v} is the external advection field in which the diffusion takes place.

We also notice in the diffusion process, multiple pigments compete with one another, which is the so-called ‘‘cross diffusion phenomenon’’ in surface chemistry. The Lotka-Volterra competition model [YLP05] is a standard mathematical treatment for the cross diffusion process with two participating diffusing substances in a 1D domain. In our problem, however, the paper is modeled as a 2D plane. We extend their model to the 2D domain with multiple compet-

ing pigments:

$$\frac{\partial \rho(i)}{\partial t} = \kappa_{s,i} \nabla^2 \rho(i) + \sum_{j \neq i}^{\xi} \kappa_{c,i,j} \nabla^2 \rho(j) \quad (i = 0, \dots, n), \quad (9)$$

where $\rho(i)$ is the concentration of the i -th pigment, $\kappa_{s,i} \geq 0$ is the self-diffusivity of the i -th pigment, $\kappa_{c,i,j}$ is the cross-diffusivities between the i -th and the j -th pigment species, and n is the number of pigment species participating in the process. In our design, the 0-th pigment is water and the 1-st pigment is glue.

In our context, ρ actually refers to $\rho_{x,y}^{mo,t}(i)$. Combining the advection diffusion equation (8) with the above extended cross-diffusion equation (9) gives the final form of the equation to be used in our simulation:

$$\frac{\partial \rho_{x,y}^{mo,t}(i)}{\partial t} = \kappa_{d,i,x,y} \nabla^2 \rho_{x,y}^{mo,t}(i) + \sum_{j \neq i}^{\xi} \kappa_{c,i,j} \nabla^2 \rho_{x,y}^{mo,t}(j) - \mathbf{v}_{x,y}^t \cdot \nabla \rho_{x,y}^{mo,t}(i) \quad (i = 0, \dots, n) \quad (10)$$

where $\kappa_{d,i,x,y}$ is the diffusion coefficients at paper position (x,y) for the i -th pigment, $\mathbf{v}_{x,y}^t$ denotes the velocity field imposed by the external conditions, which in here is estimated to be half of the velocity of brush movement, as suggested by [BWL04]. Notice that, since $\kappa_{s,i}$ and $\kappa_{d,i,x,y}$ are acting on the term $\nabla^2 \rho$ in (8) and (9) respectively, for simplicity, we absorb $\kappa_{s,i}$ into $\kappa_{d,i,x,y}$ during the above combined operation.

Once $\frac{\partial \rho_{x,y}^{mo,t}(i)}{\partial t}$ is known, $\rho_{x,y}^{mo,t+\Delta t}(i)$ can be trivially updated as $\rho_{x,y}^{mo,t+\Delta t}(i) = \rho_{x,y}^{mo,t}(i) + \frac{\partial \rho_{x,y}^{mo,t}(i)}{\partial t} \Delta t$.

5. Evaporation

We consider evaporation at the brush tip and on the paper surface. For simplicity and efficiency concerns, we only deal with the water (the 0-th pigment) evaporation process.

5.1. At the brush tip bundle

We adopt the diffusion based evaporation equation suggested in [Cra75] which considers both evaporation and the propagation of the evaporation results to the surroundings:

$$-\kappa_e \frac{\partial \rho_{r,\theta,z}^{bru,t}(0)}{\partial r} = \epsilon_{bru} (\rho_{equi}(0) - \rho_{r,\theta,z}^{bru,t}(0)), \quad (11)$$

where $\rho_{equi}(0)$ is the current water concentration in the atmosphere, i.e., the equilibrium vapor pressure; and $\rho_{r,\theta,z}^{bru,t}(0)$ is the actual water concentration; κ_e is an evaporation coefficient; ϵ_{bru} is a user adjustable scaling constant. Fortunately, [New31] gave the analytic solution for (11), which is:

$$\frac{\rho_{r,\theta,z}^{bru,t+\Delta t}(0) - \rho_{r,\theta,z}^{bru,t}(0)}{\rho_{equi}(0) - \rho_{r,\theta,z}^{bru,t}(0)} = 1 - \sum_{n=1}^{\infty} \frac{2LJ_0(r\beta_n/\epsilon_{bru})}{(\beta_n^2 + L^2)J_0(\beta_n)} e^{-\beta_n^2 \kappa_e t / \epsilon_{bru}^2}, \quad (12)$$

where β_n 's are the roots of $\beta J_1(\beta) - LJ_0(\beta) = 0$ and $L = \epsilon_{bru} / \kappa_e$. Again, for speed, we store the pre-computed β_n 's for a range of equations in the above form in a table.

5.2. On the paper surface

We employ a slightly different diffusion based evaporation equation (13) to handle evaporation on the paper surface since pigment concentrations at different depths of the paper differ insignificantly; we ignore these differences in our simulation.

$$-\kappa_e \frac{\partial \rho_{x,y}^{mo,t}(0)}{\partial t} = \epsilon_{paper} (\rho_{equi}(0) - \rho_{x,y}^{mo,t}(0)), \quad (13)$$

where ϵ_{paper} is a user adjustable scaling constant. Note that as mentioned earlier, unlike in [BWL04] where the paper is modelled as a number of layers, we only classify all the particles in the PWS as either in a mobile state or an immobile state. According to this equation, at the end of each time step, we update $\rho_{x,y}^{mo,t}(0)$ numerically.

6. Pigment Deposition on the Paper Fibers

During the process of pigment diffusion, pigment particles whose velocity is relatively low tend to be captured by and thus fixed to those paper fibers having a high surface free energy [Mas96]. In chemical engineering, this is the phenomenon of a typical adsorption process [Rut84, Suz90]. Similar to the process described by the Dubinin-Radushkevich isotherm equation (1), the intermediate adsorption process progresses towards the equilibrium state rather rapidly [Tie94]. Thus, we model the above rapid adsorption process via the Langmuir isotherm equation [Lan08], an equation to describe the equilibrium state of a system:

$$\Delta \rho(i) = \frac{\kappa_{l,i} \rho(i)}{1 + \sum_{j=1}^{\xi} \kappa_{l,j} \rho(j)}, \quad (14)$$

where ξ is the number of participating species of adsorbates; $\rho(i)$ and $\Delta \rho(i)$ are the concentration of the i -th adsorbate in the solution phase and that of the adsorbed adsorbent phase respectively; $\kappa_{l,i}$ is the Langmuir constant for the i -th adsorbate. As pointed out in [Yan03], $\kappa_{l,i}$ can be estimated

as: $\kappa_{l,i} \triangleq \frac{\kappa_{behavior,i} \kappa_{behavior}^{paper}}{2\pi \rho(i) T \sqrt{\eta_{x,y}}} e^{\kappa_i / T}$ where κ_i is a thermodynamics coefficient, $\eta_{x,y}$ is the local density of the paper fiber, and $\kappa_{behavior}^{paper}$, $\kappa_{behavior,i}$ are the adsorption behavior coefficients of the paper and the i -th pigment respectively. $\kappa_{behavior}^{paper}$ is determined by the smoothness of the paper surface, the size and distribution of the micro-scale pores formed by neighboring paper fibers; $\kappa_{behavior,i}$ is affected by the adsorption ability of the i -th pigment, e.g., its material polarity. Since the factor $\frac{1}{2\pi T} e^{\kappa_i / T}$ remains constant, we absorb the term into $\kappa_{behavior}^{paper}$, giving the simplified form of $\kappa_{l,i} \triangleq \frac{\kappa_{behavior,i} \kappa_{behavior}^{paper}}{\rho(i) \sqrt{\eta_{x,y}}}$.

Because the deposition process is caused by the random walk of diffusing substance particles [Cra75, Rut84], we further modulate $\kappa_{l,i}$ with a random walk coefficient κ_{ranwk} , which is a random number from a user adjustable random distribution.

Similar to the treatment where we employ the Dubinin-Radushkevich equation in Sec. 3, we perform the same simplifying scaling conversion to derive the intermediate results of the sorption process. Substituting the corresponding variables in our context into (14), we have the final form of the equation for our simulation:

$$\Delta \rho_{x,y}^{im,t}(i) = \min\left(\frac{1-e^{-\Delta t}}{1-e^{-\chi}}, 1\right) \frac{\frac{\kappa_{ranwk} \kappa_{behavior,i} \kappa_{paper}^{paper}}{\rho_{x,y}^{im,t}(i) \sqrt{\eta_{x,y}}} \rho_{x,y}^{mo,t}(i)}{1 + \sum_{j=1}^{\xi} \left(\frac{\kappa_{ranwk} \kappa_{behavior,j} \kappa_{paper}^{paper}}{\rho_{x,y}^{im,t}(j) \sqrt{\eta_{x,y}}} \rho_{x,y}^{mo,t}(j) \right)}, \quad (15)$$

where $\rho_{x,y}^{im,t}(i)$ is the concentration of the portion of the i -th pigment in an immobile state at paper site (x,y) at time t ; and χ is the time span to arrive at the equilibrium state.

To ensure the conservation of pigment mass during the deposition process, we additionally employ a mass transfer conservation formula [Tie94]:

$$|\mathbf{v}_{x,y}^t| \Delta \rho_{x,y}^{mo,t}(i) + \eta_{x,y} \Delta \rho_{x,y}^{im,t}(i) = 0, \quad (16)$$

according to which we can update $\rho_{x,y}^{mo,t+\Delta t}(i)$ as $\rho_{x,y}^{mo,t+\Delta t}(i) = \rho_{x,y}^{mo,t}(i) - \frac{\eta_{x,y}}{|\mathbf{v}_{x,y}^t|} \Delta \rho_{x,y}^{im,t}(i)$.

7. Rendering the Simulation Results

The result of the simulation is a collection of pigment concentrations: $\rho_{x,y}^{im,t}(i)$ and $\rho_{x,y}^{mo,t}(i)$ ($i = 1, \dots, \xi$) for each paper site (x,y) . To render the painting, we need to solve two problems: 1) To derive the overall appearance of the mobile layer and that of the fixture (immobile) layer, each containing a number of pigment constituents; 2) to superimpose these two layers on top of the background paper layer. To address the first problem and to achieve high fidelity, we propose a new machine learning approach for pigment mixing, which will be detailed shortly; the second problem is solved via the Kubelka-Munk model (KM model) [Kub48] with inspiration from [DH96, CAS*97].

Among the popular models for pigment mixing, such as average mixing, additive or subtractive mixing, and the KM model, the KM model is the most favored one as suggested by Haase and Meyer [HM92]. However, even with the KM model, there are well known limitations. Although there are better models such as the three channel model [PE73], simulation of pigment mixing behaviors still fails to agree completely with the real behaviors. Baxter et al. reported in [BWL04] that using 101 samples, each corresponding to a wavelength, pigment mixing through the KM model could still markedly deviate from reality. We propose a novel kernel method assisted neural network approach for pigment mixing.

7.1. Acquisition of training samples

In the preprocessing stage, we collect a number of pigment mixing samples for training our pigment mixing pre-

diction network. We rely on both physically-based measurement and interactive user input to collect training samples. In the physical measurement method, we use the desktop reflective spectrophotometer X-RiteColor® Digital Swatchbook® [XRi06] to measure the reflectance of pigments. We measure the reflectance of both pure pigments before mixing and that of the resultant pigment mixture for the training samples. For each measurement, we put the sample against a background material whose reflectance is known. We measure the overall reflectance of the two superimposing layers. For robustness, we carry out five measurements for five different background materials and use a least squared approach to determine the optimal R and T . According to the KM model, once R is known, we can compute $\frac{K}{S}$ as $\frac{(1-R)^2}{2R}$ [Kub48]. Inspired by [DH96], we then assume a value for S , namely 1, so that K is also known.

7.2. Kernel functions

Kernel methods are popular in machine learning because they can increase the learnability of many models [CS02]. To enhance the learning ability of our pigment mixing prediction network, we employ three kernel functions: ψ_1 , which is based on the KM model, and ψ_2 and ψ_3 , which are based on average mixing.

With the KM model and the K and S coefficients known for each pigment constituent, the overall pigment mixture's coefficients, K_{mix} and S_{mix} , are computed as the weighted average of each pigment component's K and S coefficients using the pigment's relative concentration as the weight. With K_{mix} and S_{mix} , the overall reflectance R_{mix} and transmittance T_{mix} of the layer can be computed [Kub48, DH96, CAS*97]. R_{mix} and T_{mix} will be used as the input to our neural network and also for superimposing the layers.

For the kernel function of average mixing, the overall appearance of a pigment mixture can be simply derived through weighted average. We also conduct interpolation in the HSV color space since it captures some non-linear relationship between colors and could give us a very different prediction on pigment mixing than operations in a linear space; the prediction could serve as a good clue for our neural network.

In this work, we always use a particular yellow-white light source, which is widely used in Oriental art exhibitions and many museums. Under this condition, we perform the above pigment mixing in the color space, and given the predicted color after pigment mixing, we can inversely compute the reflectance R_{mix} of the overall pigment mixture. Unfortunately, we do not have a handy way to estimate T_{mix} for these two average mixing methods. As can be seen in the subpart of Fig. 2 that is dedicated to illustrate the working of kernel functions, i.e. ψ_1, ψ_2, ψ_3 , only the first kernel function gives an intermediate estimated T value to help the learning process of the neural network.

7.3. Neural network for pigment mixing prediction

Our neural network takes as input the concentrations of the two pigment components to be mixed, ρ_1, ρ_2 , their absorption and scattering coefficients, K_1, K_2, S_1, S_2 , as well as their reflectances and transmittances R_1, R_2, T_1, T_2 , whose estimations have been discussed in Sec. 7.2. Strictly speaking, these quantities are all wavelength dependent. For simplicity, we only consider three typical wavelengths—those of pure red, green and blue light. Thus, each of $K_1, K_2, S_1, S_2, R_1, R_2, T_1, T_2$ is a vector carrying three scalar components. For each pair of pigments to be mixed, we designate a dedicated neural network for pigment mixing prediction. This is because some pigments made from different chemical compounds may carry the same color yet behave differently during pigment mixing. This also enables the user to freely customize a pigment which may not be present in the physical world. We feed the outputs of the three kernel functions, ψ_1, ψ_2, ψ_3 , as the additional inputs to the neural network. The output of the neural network is the reflectance of the pigment mixture R_o^{mix}, T_o^{mix} . In the prediction network, we introduce two hidden layers with full connections to all the intermediate, input and output nodes.

The schema of our neural network approach for pigment mixing prediction is shown in Fig. 2. We use the resilient back propagation technique [Hay01] to train our multi-layer neural network at the preprocessing stage. Each neuron in the network takes a linear function as its associated transfer function. The performance function used when training the network is the mean square error measure. The total sample space contains 1377 two-component pigment mixture instances over every pair of the 18 kinds of pure pigments, and each pair has three different concentrations. We then make the neural network go through 350000 iterations for training, which takes about 4.5 hours to complete on a desktop computer (Pentium IV with 3.0 GHz CPU). During each run, the neural network converges very rapidly in the first 50000 iterations. We use a fixed learning rate throughout the learning process. We expect when employing a new mechanism in the future that supports an adaptive learning rate, the total number of learning iterations could be significantly reduced. Finishing the whole 10 runs of the early stopping training process takes about two days.

Our neural network based approach can capture the possible chemical reactions, the mutual influences between pigments and other factors not considered in the KM model or the average mixing method. This approach also supports users to create novel pigments. The assistance from kernel functions largely improves the learning capability of our pigment mixing prediction network and at the same time ensures the robustness of our learning method—in the worst case where the training data are very sparse or ill posed, our approach can still perform at least as well as the KM model or the averaging mixing.

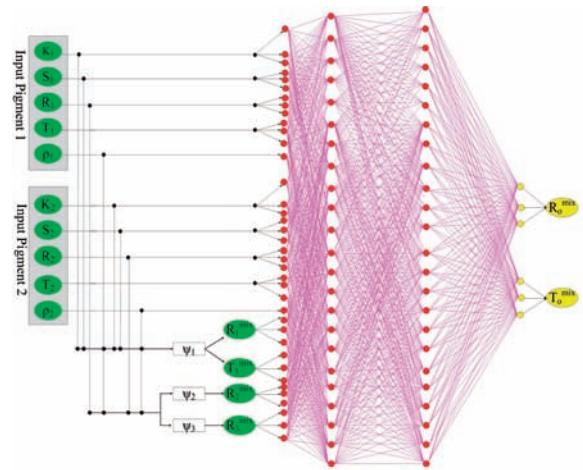


Figure 2: Pigment mixing prediction neural network.

8. Experiment Results

We implemented our method using Microsoft Visual C++ 6.0 and with the support of Microsoft Direct3D V9.0 on a PC with a Pentium 4 3.0 GHz processor, 1 GB main memory and an NVidia GeForce 6600 GT graphics card. The paintbrush system came from our previous e-brush research project [XTLP02, XLTP03, XTLP04]. With our current prototype system implementation, to paint a stroke covering 10000 pixels, the overall system responds at an average rate of 8 frames per second. We also employ GPU hardware to perform general purpose computations to accelerate the equation solving process [HBSL03, KW03], particularly when evaluating the equation (10). This is done in a way similar to that suggested in [PF05] (Part IV: General-purpose computation on GPUs: A Primer, Sec. 31.5, pp. 505–508).

Fig. 3 gives some pigment mixing results using our proposed boosted neural network approach. Fig. 4–Fig. 10 show some stroke effects produced by our new pigment model. More results are available in our project website <http://www.cs.hku.hk/~songhua/pigment>.

9. Conclusion

In this paper, we introduce a novel pigment model based on careful treatment of many minute physical factors, which can achieve the best attainable effects for Oriental and water-based painting. Experiment results have verified the success of the design and its implementation. Future work includes further acceleration of our current algorithm, providing a more friendly graphical user interface, and extending our algorithm to other forms of painting such as oil painting and wax painting.



Figure 3: Pigment mixing using our neural network. The first and second rows are the two source pigments before mixing. The third row is the pigment mixing results sampled from real world experiment. The fourth row is the prediction of the pigment mixing behavior using our proposed neural network, but without employing the kernel functions. The fifth row is the prediction using our proposed neural network with the kernel functions.

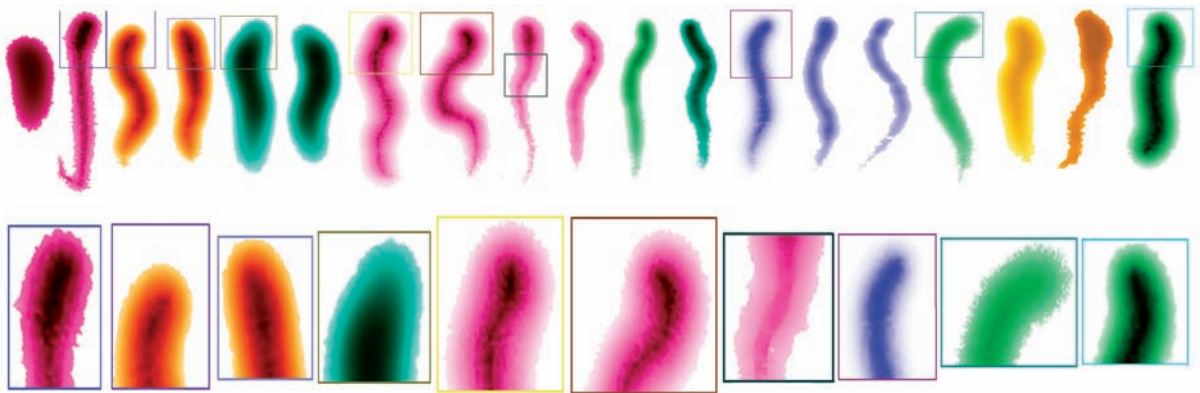


Figure 4: Different wet stroke in colors.

10. Acknowledgement

This project is supported in part by a Hong Kong RGC CERG grant (No. 7145/05E).

References

- [AG97] ADAMSON A. W., GAST A. P.: *Physical Chemistry of Surfaces*. Wiley-Interscience, August 1997.
- [Bax04] BAXTER W.: *Physically-based modeling techniques for interactive digital painting*. PhD thesis, University of North Carolina at Chapel Hill, 2004.
- [BLR06] BEETS K., LAERHOVEN T. V., REETH F. V.: Introducing artistic tools in an interactive paint system.
- [BSLM01] BAXTER B., SCHEIB V., LIN M. C., MANOCHA D.: Dab: interactive haptic painting with 3D virtual brushes. In *Proc. of SIGGRAPH* (New York, NY, USA, 2001), ACM Press, pp. 461–468.
- [BWL04] BAXTER W. V., WENDT J., LIN M. C.: IMPaSTo: A realistic model for paint. In *Proc. of the 3rd International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, 2004.
- [CAS*97] CURTIS C. J., ANDERSON S. E., SEIMS J. E., FLEISCHER K. W., SALESIN D. H.: Computer-generated watercolor. In *Proc. of SIGGRAPH* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 421–430.
- [CPE92] COCKSHOTT T., PATTERSON J., ENGLAND D.: Modelling the texture of paint. *Computer Graphics Forum* 11, 3 (1992), 217–226.
- [Cra75] CRANK J.: *The Mathematics of Diffusion*. Clarendon Press, Oxford, 1975.
- [CS02] CRISTIANINI N., SCHOLKOPF B.: Support vector machines and kernel methods: the new generation of learning machines. *AI Magazine* 23, 3 (2002), 31–41.
- [CT04] CHU N. S., TAI C.-L.: Real-time painting with an express virtual Chinese brush. *IEEE Computer Graphics and Applications* 24 (2004), 76–85.
- [CT05] CHU N. S.-H., TAI C.-L.: Moxi: real-time ink dispersion in absorbent paper. *ACM Trans. Graph.* 24, 3 (2005), 504–511.
- [DH96] DORSEY J., HANRAHAN P.: Modeling and rendering of metallic patinas. In *Proc. of SIGGRAPH* (New York, NY, USA, 1996), ACM Press, pp. 387–396.
- [GK91] GUO Q., KUNII T. L.: Modeling the diffuse painting of Sumie. *IFIP modeling in computer graphics*, Ed. T.L.Kunii (1991), 329–338.
- [GK03] GUO Q., KUNII T. L.: “Nijimi” rendering algo-

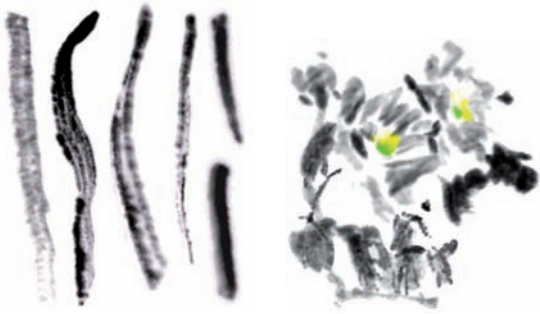


Figure 5: Some wet strokes (left), and a quick sketch of some flowers using mostly wet strokes (right).



Figure 6: Dry strokes.



Figure 7: Some strokes mixed wet and dry effects.



Figure 8: Feathering effects: strong feathering (left) and weak feathering (right). To show the interplay between the feathering effects and the background paper texture, we superimpose our results on top of a section of a scanned real paper. A close-up view of a stroke with strong feathering is displayed at the left bottom.



Figure 9: From dry strokes to semi-dry, semi-wet strokes.



Figure 10: Modest feathering effects with wet color strokes.

rithm for creating quality black ink paintings. In *Proc. of Computer Graphics International (CGI)* (2003), pp. 152–159.

[Guo95] GUO Q.: Generating realistic calligraphy words. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E78A*, 11 (1995), 1556–1558.

[Hay01] HAYKIN S.: *Neural Networks: A Comprehensive Foundation*, second ed. Prentice Hall, 2001.

[HBSL03] HARRIS M. J., BAXTER W. V., SCHEUER-MANN T., LASTRA A.: Simulation of cloud dynamics on graphics hardware. In *HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on graphics hardware* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 92–101.

[HM92] HAASE C. S., MEYER G. W.: Modeling pigmented materials for realistic image synthesis. *ACM Trans. Graph.* 11, 4 (1992), 305–335.

[KNH95] KUNII T., NOSOVSKIY G., HAYASHI T.: A diffusion model for computer animation of diffuse ink painting. In *Proc. of Computer Animation* (Geneva, Switzerland, 1995), pp. 98–102.

[KNV01] KUNII T., NOSOVSKIY G., VECHERLININ V.: Two-dimensional diffusion model for diffuse ink painting. *Int. J. Shape Model.* 7, 1 (2001), 45–58.

[Kub48] KUBELKA P.: New contributions to the optics of intensely light-scattering material, part i. *J. Opt. Soc. Am.* 38 (1948), 448.

- [KW03] KRUGER J., WESTERMANN R.: Linear algebra operators for GPU implementation of numerical algorithms. *ACM Trans. Graph.* 22, 3 (2003), 908–916.
- [Lan08] LANGMUIR I.: The adsorption of gases on plane surfaces of glass, mica and platinum. *J. Amer. Chem. Soc.* 40 (1908), 361–1403.
- [Lee97] LEE J.: Physically-based modeling of brush painting. *Comput. Networks ISDN Syst.* 29, 14 (1997), 1571–1576.
- [Lee99] LEE J.: Simulating oriental black-ink painting. *IEEE Computer Graphics and Applications* 19, 3 (1999), 74–81.
- [Lee01] LEE J.: Diffusion rendering of black ink paintings using new paper and ink models. *Computers and Graphics* 25, 2 (2001), 295–308.
- [LLR04] LAERHOVEN T. V., LIESENBORG J., REETH F. V.: Real-time watercolor painting on a distributed paper model. In *Proc. of Computer Graphics International* (2004), pp. 640–643.
- [LR05] LAERHOVEN T. V., REETH F. V.: Real-time simulation of thin paint media. *SIGGRAPH2005 sketch* (2005).
- [LS04] LIN W.-J., SHIH Z.-C.: Computer-generated Chinese painting with physically-based ink and color diffusion. In *In Proc. of CGW* (2004).
- [Mas96] MASEL R. I.: *Principles of Adsorption and Reaction on Solid Surfaces*. Wiley-Interscience, March 1996.
- [McC01] MCCASH E. M.: *Surface Chemistry*. Oxford University Press, July 2001.
- [McG74] MCGREGOR R.: *Diffusion and sorption in fibers and films (Volume 1 An introduction with particular reference to dyes)*. Academic Press, 1974.
- [Mis69] MISRA D. N.: *Adsorption on Heterogeneous Surfaces: A Dubinin-Radushkevich Equation*. Surface Science, 1969.
- [New31] NEWMAN A. B.: *Trans. Am. Inst. Chem. Engrs.* 27 (1931), 203.
- [PE73] PAULI H., EITL D.: Comparison of different theoretical models of multiple scattering for pigmented media. *Colour* 73 (1973), 423–426.
- [PF05] PHARR M., FERNANDO R.: *GPU gems 2: Programming techniques for high-performance graphics and general-purpose computation*. nVIDIA, 2005.
- [Pha91] PHAM B.: Expressive brush strokes. *CVGIP: Graph. Models Image Process.* 53, 1 (1991), 1–6.
- [RMN03] RUDOLF D., MOULD D., NEUFELD E.: Simulating wax crayons. In *Proc. of Pacific Graphics* (2003), IEEE Computer Society, pp. 164–173.
- [Rut84] RUTHVEN D. M.: *Principles of adsorption and adsorption processes*. John Wiley & Sons Inc., 1984.
- [Sma91] SMALL D.: Modeling watercolor by simulating diffusion, pigment, and paper fibers. *Proc. of SPIE* (1991).
- [Str86] STRASSMANN S.: Hairy brushes. In *Proc. of SIGGRAPH* (New York, NY, USA, 1986), vol. 20, ACM Press, pp. 225–232.
- [Suz90] SUZUKI M.: *Adsorption Engineering*. Elsevier, 1990.
- [SW04] SANTHANAGOPALAN S., WHITE R. E.: Series solution to the transient convective diffusion equation for a rotating disk electrode. *The Electrochemical* (2004).
- [Tie94] TIEN C.: *Adsorption Calculations and Modeling*. Butterworth-Heinemann, November 1994.
- [WI00] WONG H., IP H.: Virtual brush: a model-based synthesis of Chinese calligraphy. *Comput. & Graphics* 24, 1 (2000), 99–113.
- [Wor96] WORLEY S.: A cellular texture basis function. In *Proc. of SIGGRAPH* (1996).
- [XLTP03] XU S., LAU F., TANG F., PAN Y.: Advanced design for a realistic virtual brush. *Computer Graphics Forum (Proceedings of Eurographics)* 22, 3 (2003), 533–542.
- [XRi06] X-Rite Incorporated, <http://www.xrite.com/>, 2006.
- [XTLP02] XU S., TANG M., LAU F., PAN Y.: A solid model based virtual hairy brush. *Computer Graphics Forum (Proceedings of Eurographics)* 21, 3 (2002), 299–308 & 625.
- [XTLP04] XU S., TANG M., LAU F., PAN Y.: Virtual hairy brush for painterly rendering. *Graphical Models* 66, 5 (2004), 263–302.
- [XXK*06] XU S., XU Y., KANG S. B., SALESIN D. H., PAN Y., SHUM H.-Y.: Animating Chinese paintings through stroke-based decomposition. *ACM Trans. Graph.* 25, 2 (2006), 239–267.
- [Yan03] YANG R. T.: *Adsorbents: Fundamentals and Applications*. Wiley-Interscience, March 2003.
- [YLLC03] YU Y. J., LEE D. H., LEE Y. B., CHO H. G.: Interactive rendering technique for realistic oriental painting. *Journal of WSCG* 11 (2003), 538–545.
- [YLP05] Y. LOU S. M., POLACIK P.: Loops and branches of coexistence state in a Lotka-Volterra competition model. *AMS* (2005).
- [ZST*99] ZHANG Q., SATO Y., TAKAHASHI J., MURAOKA K., CHIBA N.: Simple cellular automaton-based simulation of ink behavior and its application to suibokuga-like 3D rendering of trees. *J. Vis. Comput. Animat.* 10, 1 (1999), 27–37.