Spray 2D Non-photorealistic Rendering System on Mobile Devices

Enroll. No. - 122207
Name of Student - Ashutosh Soni
Name of supervisor(s) - Prof S.P. Ghrera (Guide)
Mr. Kapil Saini(Co-Guide)

May – 2014

Submitted in partial fulfillment of the Degree of
Master of Technology

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT
SOLAN, HIMACHAL PRADESH
INDIA

2014

INDEX

CERTIFICATE

This is to certify that the work titled "**Spray 2D Non-photorealistic Rendering System on Mobile Device**" submitted by "**Ashutosh Soni**" in partial fulfillment for the award of degree of **M. Tech CSE** of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor ……………………..
Name of Supervisor Prof S.P. Ghrera
Designation ……………………..
Date ……………………..

Signature of Co-Supervisor ……………………..
Name of Co-Supervisor Mr. Kapil Saini
Designation ……………………..
Date ……………………..

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor Prof S.P. Ghrera (Guide), Mr. Kapil Saini (Co-Guide) for the continuous support of my M Tech study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my M Tech study.

Besides my advisor, I would like to thank the rest of my thesis Evaluation committee: Prof. Deepak Dahiya, Dr P.K. Gupta, Dr Pooja Jain, Mr. Suman Saha and Mr. Shailendra Shukla for their encouragement, insightful comments, and hard questions.

Last but not the least, I would like to thank my family: my parents Manju Soni and Hitendra Soni, for giving birth to me in the first place and supporting me spiritually throughout my life.

Signature of the student ……………………..

Name of Student ……………………..

Date ……………………..

SUMMARY

This thesis addresses real-time interactive non-photorealistic rendering technique to render an image as *a Watercolor Spry painting*. In the recent past, mobile devices have seen exponential growth in computer graphic hardware with the inclusion of high précised embedded motion sensors. These factors yield platform for the development of real-time NPR on a mobile base.

A raster image is taken as input to an algorithm that produces a painting-image composed of spray droplet rather than pixels. The algorithm consists of two stages. The first stage is *Spray Handle-fluid color which* simulates the dynamics of spray tools and ink. Second stage is *Spray droplet-canvas Interaction* which simulates dispersion and absorption of the spray color droplet on canvas using the Lattice-Boltzmann methods (LBA). Ultimately the pattern of spray droplet representing an interpreted image can be rendered in pixel form. Unlike the previous interactive methods based on brush model, this algorithm has incorporated the spray tool to draw the spray droplets. . The spray simulation process creates the image reminiscent of modern realistic spray painters who desire a diverse practice of spray in their deeds.

Signature of Student	Signature of Supervisor	Signature of Co-Supervisor
Name	Name: Prof S.P. Ghrera	Name: Mr. Kapil Saini
Date	Date	Date

LIST OF FIGURES

LIST 0F SYMBOL

S. NO.	SYMBOL	PURPOSE
1.	$\vec{u}$	fluid Velocity
2.	ρ	Density of The fluid
3.	v	Kinematic Viscosity of the fluid
4.	$\vec{f}$	Acceleration due to External Forces Acting upon the fluid Element
5.	T	Time
6.	Eq	Equilibrium
7.	W	Weight
8.	C	Propagation Speed
9.	P	Pressure
10.	Cellx	X Coordinate of a Cell
11.	Celly	Y Coordinate of a Cell
12.	***cross-pts***	Number of Cross Point
13.	Height	Height from the Handle to Canvas
14.	PixelX	X Coordinate of the Pixel Bellow the Spray Handle
15.	PixelY	X Coordinate of the Pixel Bellow the Spray Handle
16.	C	Position of the Spray Jet
17.	P	Target Position on Canvas
18.	U	Position of Spray Handle
19.	$\boldsymbol{t_d}$	The Expected Time a Droplet Takes to Fall on the Canvas
20.	D	Density
21.	M	Mass
22.	$S_{pressure}$	Internal Pressure of the Spray Handle
23.	S_{tilt}	Tilt Angle of the Handle
24.	$S_{velocity}$	Velocity of Handle
25.	Q_{tilt}	Tilt Angle of the Spray Hole
26.	$Q_{velocity}$	Velocity of Spray Hole
27.	$Q_{pressure}$	Pressure of the Spray Hole
28.	u_{canvas}	Velocity of the Droplet on Canvas
29.	M_1	Major Axis of Ellipse
30.	M_2	Minor Axis of The Ellipse
31.	$c_{dropSpeed}$	Dropping Velocity of the Droplet
32.	r_{M_2}	Radius of Ellipse along Major Axis
33.	r_{M_1}	Radius of Ellipse along Minor Axis

Chapter 1

INTRODUCTION

- **Physical Spray Process**
- **Airless Spray Method**
- **Spray Painting Framework**
- **Motivation**

1. INTRODUCTION

Since its beginning in the early 1960s, computer graphics have been focused on the aim of developing techniques that mimic the effect of a custom photographic camera. At the time, the term ‘Photorealism’ was taken from a painting method that was known for creating the painting version of the any object with the objective of keeping the look and feel of the original exactly the same. Therefore, the methods were perfected to the issue where the resultant handmade images could hardly be differentiated from genuine images. Thus, the term photorealistic computer graphics were selected to denote algorithmic methods that contribute to resemble the outcome of the photographic camera. After over 30 years of study and development, most of the problem regarding modeling and rendering of things by the smooth and normal formal form have been solved. Even very complex scenes with numerous things discovered in nature can be developed. Whereas it is very tough to identify who really pioneered Photorealism, Ivan Sutherland is most well known computer graphic researcher who employed it in the early 60s. Photorealism has a number of appealing attributes from a research point of view. More significant, however, is advancement in the scientific technique to directly compare generated images with images taken by a camera. Apparently research community can realize the aim and can assess its advancement by easy examination. These are the absolutely vital ingredients that have assisted in the success of this area of technical endeavor.

The goal of NPR may be assumed as a test to emulate human’s facilities for constructing the graphics by hand. As an expression, using a pointed pencil to draw makes it tiring to shade a surface unquestionably. Considering this problem, cross-hatching continuous crisp lines has evolved as a methodology of approximation. Although, the goals of the AT community go well after NPR and emphasize added adaptively in client interfaces. Still, this approach is under evolution.

In numerous cases simulating truth is not as significant as giving an observer with just enough data to create the illusion of reality. A new area of computer graphics is fast growing in the vigilance that it is granted, and the amount of study that is being finished. This new part of computer graphics have been grouped under the name non-photorealistic rendering (NPR). Although it is strange that it is characterized by what it is not, it does provide a procedure to categorize work that has before been considered of as an amusing disruption. The going by car force of computer graphics since its conception has been realism. The term photorealistic rendering has been utilized to recount this large area of computer graphics.

In photorealistic rendering the approach is generally physically founded, replicating the lightweight, material properties, reflections, and refractions of things. Whereas NPR engages stylization and connection, generally propelled by human insight. NPR brings together art and research, concentrating less on the method and more on the connection content of an image. Photorealistic rendering can be considered of as a target, whereas NPR is mostly personal. In photorealistic rendering, it is hard to neglect detail; in fact the highest level of detail is usually preferred. Although, the rating of detail in NPR sprints the variety and the rating is usually adapted across the image to aim the viewer's attention. Imagery generated by creative person's supplies data about things that may not be readily apparent in images or real life. That the identical goal should apply to computer generated images are the driving force behind NPR. Studying NPR, I try to find out all the important aspect of the it that is required to consider. One fact is that computers can join to fine detail and repetitive jobs, but without a user, simulating creative sign (e.g. Stroke) is tough and, some hold, impossible. Numerous computer graphics researchers are discovering NPR methods as an alternate to realistic rendering. More significantly, NPR is now being acknowledged for its ability to broadcast crucial concepts. Methods which have long been used by creative persons can be directed to computer graphics to emphasize specific characteristics, expose subtle attributes, and omit extra information. The benefits of NPR are maximized when one thinks about the subject issue, the view composition, and the purpose of the likeness.

The input to a two-dimensional NPR system is most commonly an image; however, there are systems that take 3D geometry information as input and produce a 2D image or video as an output. Again, many of the systems are intended to mimic a desired artistic style, such as watercolor, impressionism, or pen and ink drawing. Sketchy style is a rendering form that mimics hand-drawn graph primitive.

Users who are interested in having much more control of the NPR process may be more interested in interactive techniques. Many of these NPR systems provide the user with a canvas that they can "paint" on using the cursor — as the user paints, a stylized version of the image is revealed on the canvas. This is especially useful for people who want to simulate different sizes of brush strokes, according to different areas of the image.

In contrast to the methods mentioned previously, another technique in NPR is simulating the painter's medium. Methods include simulating the diffusion of ink through different kinds of paper, and also of pigments through water for simulation of watercolor.

1.1 PHYSICAL SPRAY PROCESS[23]

Atomization refers to the process of breaking up bulk liquids into droplets. Common examples of atomization include:

- Showerheads
- Perfume sprays
- Garden hoses
- Deodorant or hair sprays

A classic example of atomization occurring naturally involves pouring liquid from a pitcher. As anyone is pouring, gradually lift the pitcher higher. As the pitcher higher is tilted higher, the stream of liquid elongates and breaks into droplets at some point. This breakup of a liquid is a simplistic example of atomization.

The thickness or viscosity of the liquid being atomized is also an important factor in the ability of any liquid to be atomized. A spray is a collection of moving droplets that are the result of atomization. Naturally occurring sprays are rain and ocean sprays. See Figures 1 and 2 for an illustration of a spray from the nozzle of a spray gun. Note that there are a variety of droplet sizes in the spray illustration. It is also important to understand that a droplet is a small particle of liquid. Droplets are also known as particles.

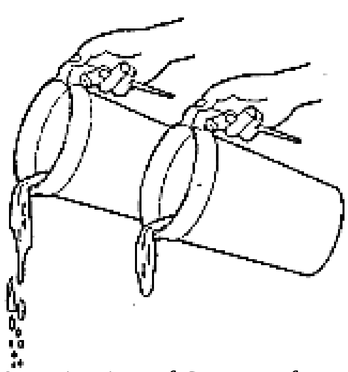


fig.1 Atomization of Stream of liquid [23]

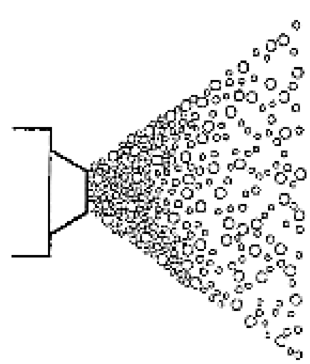


fig.2 A spray stream with a variety of droplet sizes. [23]

There are four major spray systems·

- Airless
- Conventional
- Electrostatic
- High Volume Low Pressure (HVLP)

Even though each of the four spray systems is different from one another, they all function as equipment that mechanically applies coatings to a variety of substrates.

1.2 AIRLESS SPRAY METHOD

The most widely used of the four major spray systems is airless spraying. Airless spraying, as the name implies, does not require air to atomize the spray. Simply speaking, airless sprayers pick up paint from a paint source, and with a pump, push the paint through a spray tip.

Spray tips come in a variety of types and sizes. It takes tremendous pressure to push the paint through a spray tip. The pump is responsible for creating enough pressure for the paint to be pushed through the spray tip. Atomization takes place after the pushed paint leaves the spray tip due to the pressure being created by the spray pump. The pressure of the paint, leaving the spray tip is measured in pounds per square inch (psi). Airless sprayers can create anywhere from 500 to 5000 psi. The major components of an airless system are:

- Pump (picks up paint from paint source and delivers pressurized paint to the spray gun via hoses)
- Hoses (deliver paint from the pump to the spray gun)
- Spray gun (receives pressurized paint from the pump, via hoses, and atomizes the paint as it leaves the spray gun)

1.2.1 VISCOSITY

Viscosity, the thickness of a coating, is the main subject in determining droplet size, which in turn directly affects how easily al include atomizes. Mobility of the fluid is affected adversely by the effect of the viscosity, tending to prevent its breakup and leading to a bigger average droplet size upon atomization. Thickness of the fluid causes the coating to be more atomized. As in Figure 3, thicker viscosity materials show poor atomization characteristics, while thinner viscosity materials tend to have increased atomization properties.

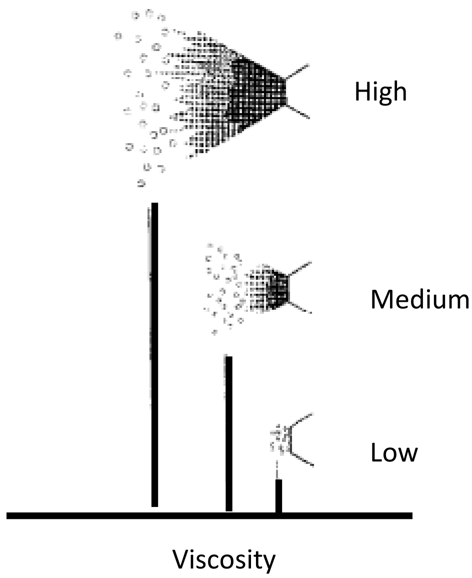


fig.3 Viscosity , droplet size, and when atomization occurs [23]

1.2.2 TRANSFER EFFICIENCY

The transfer efficiency of a spray finishing process is the quantity of paint that adheres to the target compared to the quantity of paint that was actually sprayed toward the object. Transfer efficiency is defined as the ratio of the weight of solids sprayed paint versus the weight of paint received by the target. As painters, we call our 'target' a substrate. As an example, .5 transfer efficiency means that 50 Percent of the weight of the sprayed paint in the source paint volume that was sprayed actually reached the target. The remaining 50%, which did not reach the target, is commonly referred to as overspray.

When a greater percentage of paint adheres to the target, Spray process is needed to be continues less to achieve the desired coating.

There are many factors that control the efficiency of the transfer of paint to the target. The factors that affect transfer efficiency must be understood if the goal optimal transfer efficiency is to be met. The primary spray finishing conditions that impact transfer efficiency are:

- Substrate characteristics, such as size and shape
- Skill level of sprayer
- Viscosity, atomization and spray finishing methods
- Equipment characteristics

1.2.3 OVERSPRAY

As mentioned when defining the concept Of transfer efficiency. Overspray is (the amount of material atomized that does not reach the intended target substrate). Overspray is a key apprehension that has to be dealt with in any spray application. Specific areas that must be addressed concerning overspray:

- Drifting (e.g. The wind is capable of moving overspray away from the spraying operation).
- Atomization pressure (the more you increase air or fluid pressure to a spray gun, the more overspray you have).

The choice of spray equipment matters, since it is related to the overspray proportion directly. When using an airless spraying system, the airless spray is atomized (broken up) into a fluid composed of tiny droplets, without the use of compressed air. A coating is pumped to the spray gun by the application of the pressure generated by the fluid hose, which originates the coating through the tip. The tip manages the flow of paints and creates the spray pattern; The specific functions of an airless spray tip are;

1. Determine the fluid flow or the amount of the coating applied.
2. Create back pressure in the line for an evenly atomized pattern.
3. Create the spray pattern and tan width.

Spray tips come in various styles. But the most common style of spray tip is the “RAG” tip [26]. ‘RAC' is an abbreviation for **reverse–a-clean** (or commonly referred to in the field as a reversible tip). Their one great advantage of using RAC tips. That is, these tips are easy to unclog.

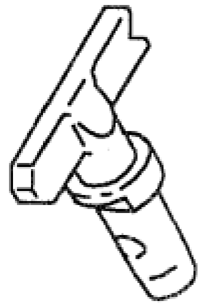


fig.4 Common RAC Tip [26]

When using a RAC tip, it is important that you rotate the tip all the way until it stops before triggering the spray gun. It is important to know that the top of RAC tips has an arrow shape. The ‘notch’ is at the back and the point is at the front of the tip. Spray tips are usually identified by a numerical system, by looking at the notched part of the spray tip (see Figure 4).

1.3 SPRAY PAINTING FRAMEWORK

Spray Painting is a significant artwork formation procedure where a device scatters droplets of liquid colors through the air onto the surface. Artists’ craft spray painting by jerking of tugging Brush, having hard bristles, towards the canvas. Hovering the paint dripping brush over the paper also another method. Experienced Artist with supreme control over brush moment techniques can conceive an image into spray version. Otherwise Spray’s tendency of being highly uneven and fickle makes the creation of required pattern a time-consuming trial and error process.

Understanding the necessity of flexibility in terms of liberty to control formation of an image, a digital painting tool with undo and redo functions is usually needed by the artists. Unfortunately, the entire modern image editing tools, software incorporate only digital brush with predefined set of patterns. Lack of alternative in creation spray causes the repetition of the patterns. Usually the artist hides this repletion by smoothly shifting to different patterns of intelligently randomizing parameter associated with the each brush stroke. But randomizing the spray process is times consuming with the possibility of losing the prerequisite need to maintain the style of the formation following some artistic desired goal.

My approach has been proposed for mobile base. Growth in mobile technology and advancement in Hardware has opened wider scope for evaluation of non-photorealistic rendering Techniques. Wider range of sensor inbuilt in device gives exciting and innovating way to create the pattern of brush stroke and new ways for stroke placement. Sensors like accelerometer, gyroscope and proximity sensor have.

Incorporating new stroke formation idea, governed by the sensor input, my experiments show the creation of realistic and organic spray pattern. Approach encapsulates the Physical aspects of the spray process by engraving physical simulation consisting of factors of viscosity and surface tension of colored fluid ink.

Pioneering research prototypes tackle the problem with simplified physical simulation. Lee al focuses on the specific painting style of Jackson Pollock and Chu et al .

[9] ignore ink dynamics in the air, but focus on the interaction between ink and paper

Spray painting system (SPS) has been designed with two stage spray simulation framework.

The two phase simulation skeleton simulates following realistic and organic spray process:

- **Spray Handle stage**: This stage simulates the Spray outlet point dynamics and ink movement within the brush.
- **Spray droplet on paper stage**: that simulates the behavior of the flow of the color droplet on the canvas surface.

In our approach movement of the ink through the path because of air movement and dynamics is approximated. The effect of the air has been generalized for all the cases.

My main contribution is an artistic version of Spray system to smooth the progress of the creation of spray painting. My approach integrates fluid simulation and Sensor based interaction modes. By giving the artists a better control of the desired spray patterns, they are permitted to create the spray painting expression that is monotonous in the real world.

1.4 MOTIVATION

Latest improvement in computer graphics has made progress in reproducing the look of the decoration by the development of various expressive methods. This task discovers the advantages of recreating the "sight, feel, activity and feeling" of the artistic process itself. Non photorealistic rendering has commendably enhanced role in computer graphics, letting users to capitalize artistic and expressive style to reform computer graphic entities. Conventionally, computer graphics research on NPR has centralized on desktop environments using well-known frameworks such as OpenGL. That approach might become obsolete in a new context in which developers must conceive applications for intelligent telephones, tablets, and wealthy Internet submissions.

To achieve this, I have conceived a physically-based, deformable, 2D Virtual Spray model and bi-directional, two-layer, paint-form to render Spray painting, exploiting inbuilt motion sensors such as accelerometer, gyroscope. Motion sensors embedded in the device permit the client to make a convoluted spray pattern interactively. The response enhances the sense of realism and presents tactile cues that enable the client to better manipulate the spray painting.

This is a unique pattern of art that is usually performed on the wall in large metropolitan towns conceiving surreal countryside of satellites, comets, pyramids, towns, nature scenes as well as easy one-colored backgrounds or even multi-colored backgrounds where the decorate swirls simultaneously or fades from one hue: Red to the other through a hue: Red sequence of differing values (of each color).

fig.5 Example 1 of real spray painting

fig.6 Example 2 of real spray painting

Chapter 2

LITERATURE REVIEW

- **Type of Non Photorealistic Rendering Airless Spray Method**
- **Literature Review on Various Ink Painting and Diffusion.**
- **Challenges in mobile Nonphotorealistic Rendering**

2. LITERATURE REVIEW

As mobile devices have been rapidly spreading up throughout the world, various services have been provided through mobile devices recently. Many services of them have been developed by using built-in sensors mounted on the mobile devices. Non-photorealistic rendering techniques have been also applied to mobile devices for the purpose of user friendliness or attractions. Nonphotorealistic rendering (NPR) allows the user to apply numerous artistic or expressive styles to normal computer Graphics entities. Recently, mobile devices have gone through remarkable growth in both hardware and software. Processing of graphical entities has tremendously improved by the utilization of GPUs. GPU holds the responsibility of processing graphical intensive job, offloading the burden of the CPU. Addition to this, high précised motion sensors has been embedded in devices. These advancements open up the scope for adoption of computer based NPR methodology for mobile devices.

I have divided literature review chapter in three sections. In the first section I have shown the basic types of the NPR methodologies – 1) Stroke based rendering that includes Brush Stroke Techniques, Mosaicking, Tiling and Stippling 2) Region-Based Techniques 3) Example-Based Techniques. 4) Image Processing and Filtering. In Second section, I have done the literature review on various Ink Painting and Diffusion Techniques. This section includes some papers that are relevant to my proposed method of the spray painting framework as spray is the process of injecting ink droplet on the surface of the paper. My last section includes Mobile NPR challenges and solution.

2.1 Types of Nonphotorealistic rendering are following:-

2.1.1 Stroke-based Rendering (SBR)

SBR algorithms have been proposed to stylize the 2D image with automatic rendering of stroke following the instruction by desired end goal, particular style. In simple words, SBR uses the eponymous virtual brush stroke. But the SBR is free to utilize any primitives that can be titled, Stipples, and hatch marks. SBR also supports multiple stroke type to draw an image.

2.1.1.1 BRUSH STROKE TECHNIQUES [3]

SBR algorithms are diverse in nature. But the most acceptable form is SBR algorithms that use short dabs of the paint, or the long curved brush strokes as the rendering primitives. Stroke placement on the canvas has been broadly categorized into two categories – local and global.

In local rendering approach, an SBR takes the decision on the basis of the local information available e.g. pixels in the spatial neighborhood of the rendering primitive. This can be explicit in the approach (e. g., image moments within a window) or implicit due to a prior convolution (e. g., Sobel edges). So it means that if the image has been changed at specific places on the canvas, then only stroked falling in the locality are affected.

In Global rendering approach, SBR algorithm is meant to optimize the placements of the all the strokes that is motivated by the objective of minimizing some objective function. There has been too many proposed Global stroke placement approach like snake relaxation, evolutionary algorithms and Monte Carlo optimization. . Each of these cases are designed to retain only perceptually important details of the intended goal relating to to retention of detail, for example, encouraging maximal retention of visual detail using low-level operators or higher level measures such as image salience.

fig.7 example of artistic vision: automatic digital painting using computer vision algorithms by Bruce Gooch

On the side of the SBR classification addressing the local side, algorithms can be categorized into used-assisted and automatic process. The methods for the automation process can be, as with the global SBR, categorized into lower and higher level analysis, guided by the importance of the field that dominates the features of the artwork.

In Sibling SBR branch that is of semi automated (user assisted) methods, the low and high level differences are mirrored. Early techniques have the dependence on image filters to orient brush strokes, but in the later work—technique have relayed on automated measures for emphasis—using gaze trackers to directly bind the perceptual measures inbuilt in the human visual system.

In recent work, the stroke placement is derived from the even more abstract higher level contextual parameters like emotion and mood. Most recently, the trend has shifted toward the process of interaction, developing semi-automated tools for painterly video that enable key framing of the fields used to arrange strokes. For automatic techniques, a clear distinction can be made between those operating over images versus video content. Video extensions of SBR are non-trivial as strokes must not scintillate (flicker) and their motion must match the underlying video content. In the SBR branch of the taxonomy this problem has largely been addressed—though by no means solved—using optical flow. Elsewhere, nonlinear filters and segmentation have been applied.

2.1.1.2 MOSAICKING, TILING AND STIPPLING [6]

There have been categories of the SBR that approximate the image using a medium other that colored pixels or paints, packing the image with a multitude of atomic rendering primitives. An approximation can be achieved by following method.

(i) ***Stippling****:* Stippling is the distribution of small points (stipples) often for the purpose of *tonal depicting* where tonal can be defined as the different shade color of the same main color group;

(ii) ***Hatching:*** In hatching method patterns and curves are placed for the same purposed described above.

(iii) ***Mosaicking:*** In Mosaicking algorithms, approximation of the image is done with the backing of the tiles.

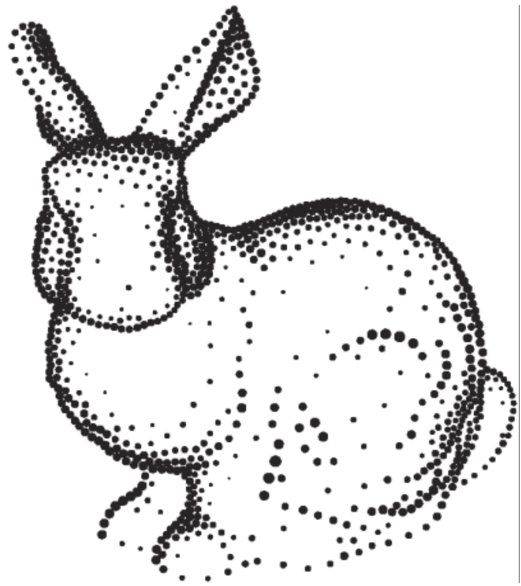


fig.8 Stippling using Secord's technique [2002].

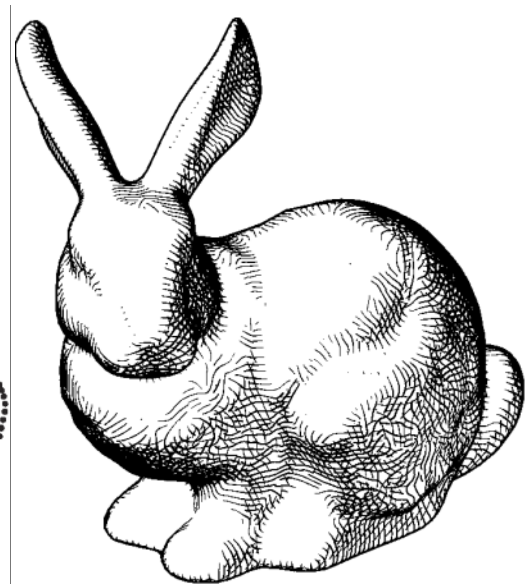


fig.9 Decorative mosaics using Schlechtweg et al.'s Render-Bots [2005].

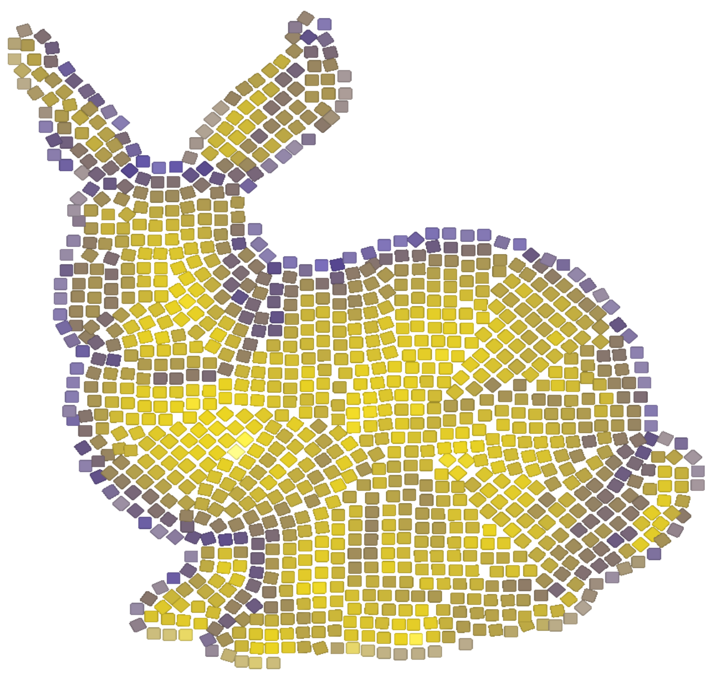


fig.10 Hatching using Zander et al.'s technique [2004].

These three techniques inherits the idea from the digital half toning and dithering algorithms that locally approximate the regions using dot patterns for the purpose of the displaying the local brightness or with an additional artistic intent. Half-toning techniques have been proposed, taking in use greedy strategies for populating regions with stipples to avoid artifacts due to aliasing. All these techniques have been based on the placing date considering local information for decision making. This culminated most recently in techniques designed to emphasize image structure, following the trend toward perceptual analysis in SBR. But stippling, in disparity to the half toning having tendency to put a white or black pixel on the regular grid, place larger dots, with the shared goal to represent the brightness and to (typically) avoid visible patterns.

Stippling techniques were developed using the brush stroke method earlier. However, most of the SBR method tended to transform into global relaxation approaches, so global approach was adopted by many stippling techniques for stipple placement. Now stippling technique's nature has shifted to capture and imitate the stippling style of artists of to form the none repetitive patterns. Some techniques have explored the idea of approximating images using lines and curves. Apart from dedicated image-based hatching approaches, some techniques grow labyrinthine patterns using space- filling curves or reaction diffusion processes that adapt to the intensity of the image. Artistic mosaicking algorithms are closely related to packing problems, and so are approached almost universally as global optimization problems. Variation in packing strategies is very high. However, on spatial and spatial-temporal basis, they can be categorized. Spatial-Temporal techniques are especially difficult because a balance must be kept between a authentic approximation of frame content and temporal incoherence caused by frequent update of the tile or glyph chosen to represent a particular spatial region.

2.1.2 REGION-BASED TECHNIQUES [10]

NPR Techniques were mostly focused on the low level image processing. Decade after 2000, focus shifted to mid level computer vision techniques. Segmentation emerged as the new prime prior image discrimination technique, enabling adaptation of rendering according to the content in the regions. Segmentation provides liberty to render the interior of regions separately.

Along with it independent region based rendering, segmentation has also explored the other sterilization technique, including cartoon flat shading and new materials such as stainless glass. For images, SBR can be classify region-based methods into those considering the arrangement of rendering primitives (e. g., strokes) within the interiors of regions and those manipulating shapes, form, and composition of regions.

A further Classification leads to exploration of the method that are based on image pyramids. A specific image region containment hierarchy, constructed be segmenting successively lower resolution versions of the source image, is browsed by various interactive techniques (human gaze-trackers, importance maps). An image can be rendered at a high level of abstraction by drawing only common large regions near the top of the hierarchy, or particular regions can be rendered in greater detail at lower levels. This enables local control over the level of detail. All such methods and approaches are basic and initial level IB-AR algorithms are significant by being among the first to consider the perceptual importance.

fig.11 Painterly rendering: strokes represent leaves or wood shingles [10]

Region based technique has also advantages over the SBR techniques, making it base dependent on the optical flow. Video segmentation is a well-studied problem in computer vision and is largely classified in to Two :

(i) techniques that segment frames independently and associate regions over time (2D+*t*)

(ii) Techniques that segment video as a spatiotemporal (*x*, *y*, *t*) volume (3D).

Both methodologies have seen applications to IB-AR for the purpose of cartooning or otherwise stylizing the appearance of video. All techniques share the observation that once video has been coherently segmented into regions (a non-trivial problem), the problem of hatching, sketching, or painting with temporal coherence can be solved by attaching strokes to a rigid or deforming region. This frames the problem of IB-AR as one of automated rotoscoping. Finally, when considering regions, it is possible to track and analyze the motion of objects. This gives rise to a complementary form of video stylization—that of artistically manipulating object motion.

2.1.3 EXAMPLE-BASED RENDERING [4]

Most of the IB-AR algorithms emulate the artistic practices with the aim of accurately produce the prescribed style. Example based rendering pioneered by the Hertzmannn is the complementary to IB-AR. This approach derives a mapping between an exemplary pair a source image and artistic version of that image. Any image can be transformed to the prescribed exemplary style by being applied by learned-mapping.

Example-based rendering (EBR) can be categorized as performing either texture or color transfer. Color EBR typically performs a piecewise mapping between the color histograms of two images to effect a Nonphotorealistic recoloring. Often there is only weak enforcement of spatial coherence in the color mapping process. By contrast, texture-based EBR shares similarities with patch-based texture in-filling techniques, which seek to fill holes in images by searching for visually similar patches elsewhere in the image. However, in the case of EBR the patches are not matched within the source image to be rendered but instead within the exemplar source image. The corresponding patch from the exemplar artistic image is then pasted into place in the output rendering.

As with texture in-filling, a careful balance must be maintained between fidelity of the patch matching and the spatial coherence in the rendering.

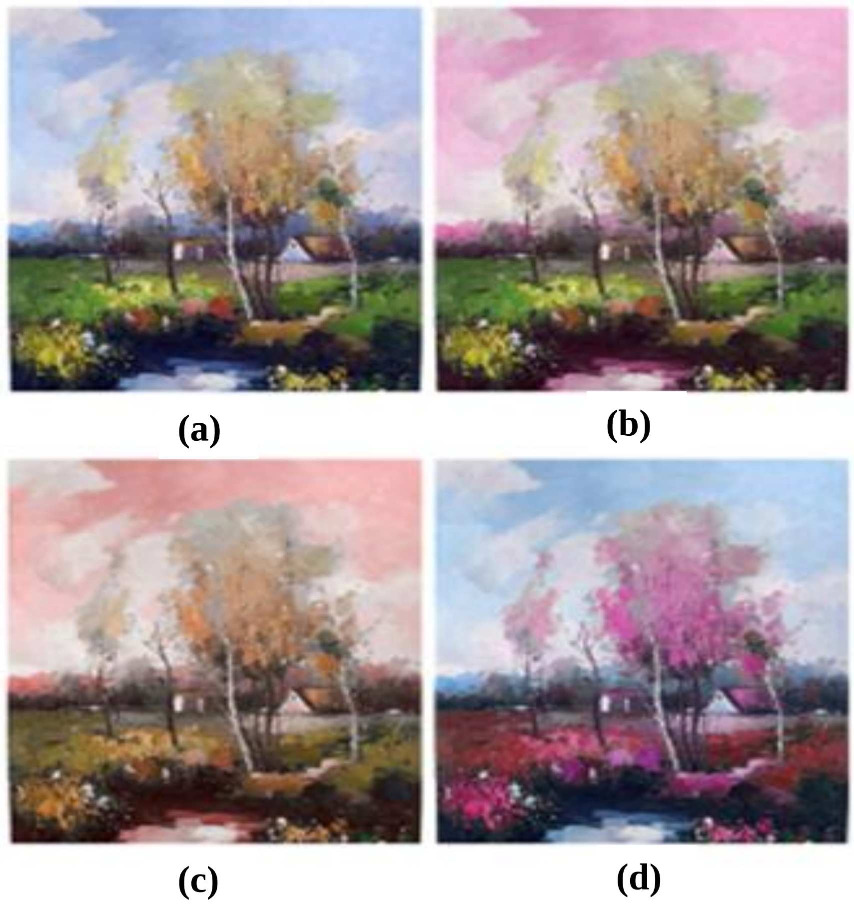


fig.12 Example-based painting guided by color features Hua Huang, Yu Zang ,Chen-Feng L [4]

2.1.4 IMAGE PROCESSING AND FILTERING [5]

There are many image processing techniques and filters, but a few of them falls in the category of IB-AR, giving the desired result from an artistic point of view. This is perhaps because these filters are designed to restore and recover photorealistic imagery. In contrary, IB-AR generally aims for simplification. For the purpose influence by artistic viewpoint, IP and filtering can be classified into two major categories. The first is the classical spatial domain where the gray or color value of a pixel is replaced based on the values of its neighboring pixels. Generally, all techniques that have been derived from classic image processing methods fall into this category.

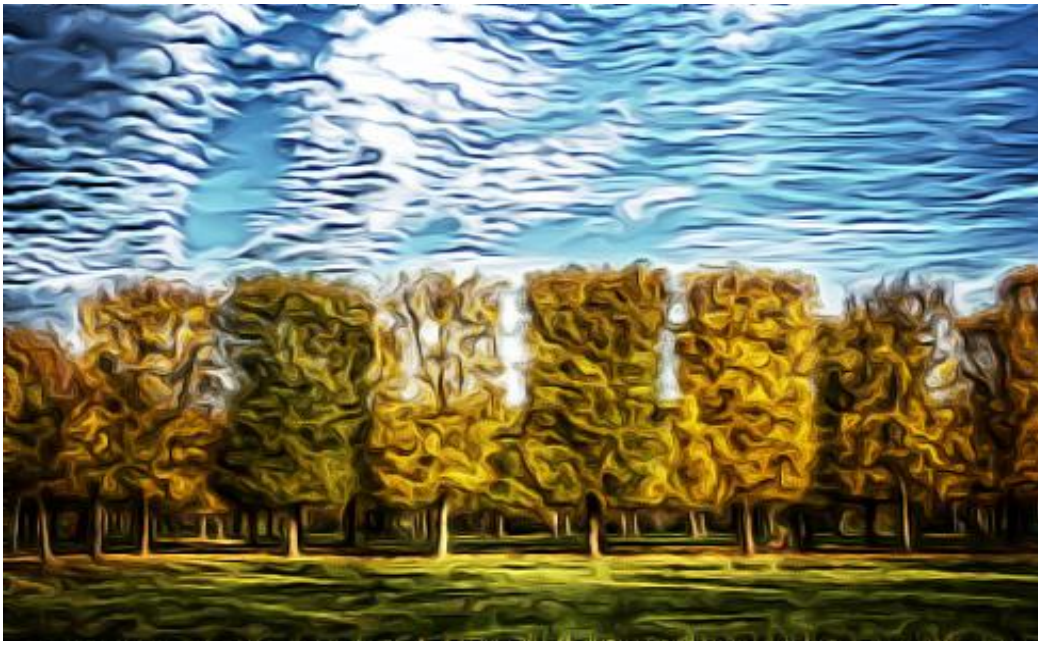


fig.13 Painterly rendering by Fracalius- like filter [5]

Second category techniques operate in the gradient domain.

In the second category, techniques operate in the gradient domain, a relatively undermined researched area of IBAR. In the spatial domain, most techniques are automatic processes. These techniques can be further subdivided by their type of output; either outlines of stylization of solid areas. Techniques creating outlines can be further classified by the type of edge detector used. We adopt the usual distinction between first and second order derivative methods. Techniques image regions are further classified into edge-preserving smoothing 1) approaches that utilize some form of anisotropic diffusion, 2) approaches based on local image statistics, and 3) approaches based on morphological filtering.

Techniques based on the bilateral filter fall into the anisotropic diffusion category since the bilateral filter can be interpreted as fast filter-based approximation of anisotropic diffusion. While fast isotropic variants of the bilateral filter have seen application elsewhere in the graphics, IB-AR variants of the bilateral filter were developed to focus on increased quality of separable approximations and focused on the enhancement of anisotropic image structures.

2.2.1 PHYSICAL-BASED MODEL OF INK DIFFUSION IN CHINESE INK PAINTINGS [6]

Sheng-Wen Huang and his fellow authors has proposed a new method to produce the pattern of ink diffusion in Chinese ink painting. Paper or canvas, which is the surface for ink to flow on, \ is conceptualized as several X-Y plane layers, each separated into paper cells. Water particles flow into holes or spaces between fibers in the paper mesh by capillarity action. The carbon particles float and propagate in this liquid since they collide with water particles. The direction and amount of flowing water are resolute from absorbency, the alignment of the fibers and the inertia due to each paper seal. The proposed ink diffusion algorithm is easily adaptable to paint many subjects in the style of Chinese ink painting with ink diffusion patterns. Significant contributions of their works are:

fig.14   Original painting [6] fig.15 Simulated painting [6]

1.The proposed algorithms are based on physical theory and analysis of observations. The resulting images are very realistic.

2.The most important contribution of this proposed method is the expression of a mixture of verities of brush strokes, such as those of two wet brushes. “Dense brush following dilute brush”, is a typical example in Chinese ink painting.

3.The diffusion of brush strokes can be easily controlled, according to experimental data, by specifying parameters. Users can easily use these parameters to control local and global variation and achieve their desired effects.

Several unknown factors affect the diffusion of ink and should be addressed. According to the proposed method, ink diffusion yields strokes that are uniform and regular on the boundary. For example, glue is a common ingredient in paper, and reduces the paper’s ability to absorb water.

Limitation

1. Only for black ink , no scope for multicolor painting
2. Paper properties like degree of roughness on the surface and glue off the surface, etc. are have not been included

2.2.3. A GPU-BASED METHOD FOR REAL-TIME SIMULATION OF EASTERN PAINTING [7]

Goal of Kiet Lu and his fellow author’s research work aims to “capture” expressivenesses of the Eastern Arts, in digitized form, which will open to new opportunities of exploration of painting techniques on the new digital Media, as well as of the easiness of calligraphically practices, complementary to the traditional way they have proposed a GPU-based method for real-time simulation of Chinese painting. It includes physically-based brush deformation and seamless integration with ink diffusion rendering on “Xuan” paper structure. 3D Brush is modeled as a large number of small bristles.

Each bristle is represented by a piece of cubic parametric Bezier curve. The deformation is physically based that takes into account in determining the balance state by minimization the actual physical bend potential energy and frictional energy of the bristles.

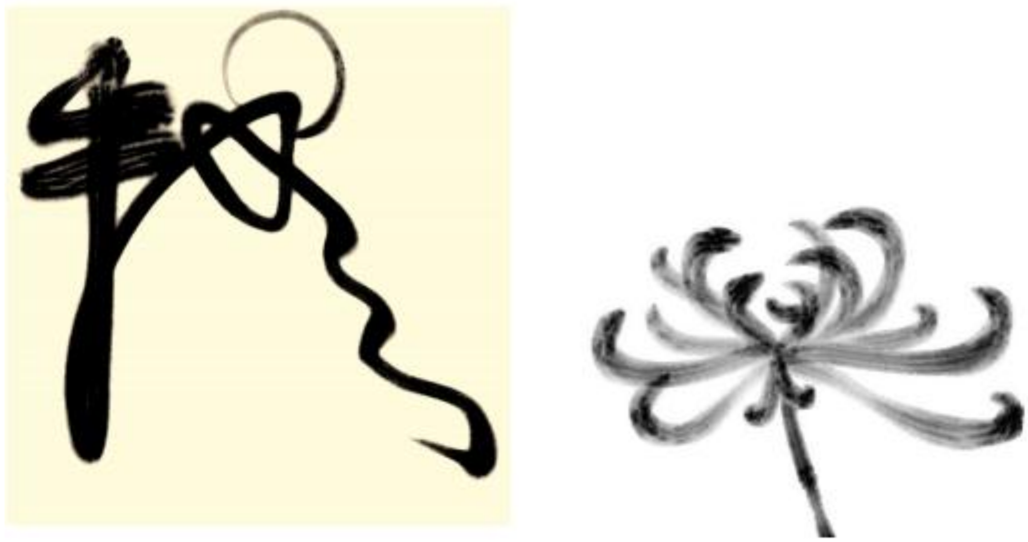


fig.16 GPU-Based Method for Real-Time Simulation of Eastern Painting [7]

Bristles splitting and clumping effect is simulated by accessing to the wetness level as in real brush painting. The footprint is generated by protecting the Bezier curves onto the paper plan which allows higher resolution output compared to previous methods. Ink diffusion is simulated based on “Xuan” paper structure. The contributions of this work are as follows:

1) A novel 3D brush simulation method can model up to thousands of each individual bristles that altogether form the 3D brush.

2) Mapping of the algorithm onto GPU parallel exploits programmable graphics hardware (GPU) to achieve interactive simulation speed.

3) Post-rendering ink diffusion effect based on physical “Xuan” paper structure is implemented and integrated seamlessly into 3D brush simulation to achieve more the realistic results.

Limitation

- The first limitation of this Bezier-based simulation is the assumption on P3 and P4 controlling points that they both have to lie on a paper plane. The purpose is to reduce the complexity of the energy minimization problem from 6 parameters to 4 parameters (Y value of P3 and P4 is assumed to be Zero). This results with unrealistic results in some cases of extremely fast stroke in Chinese, calligraphy, which is currently out of our research range.

- Another limitation is that the global behaviors for each bristle have not been touched, namely self-collision between neighboring bristles as well as the attraction force from the ink liquid that holds the bristles together.

2.2.4. REALISTIC PAINT SIMULATION BASED ON FLUIDITY, DIFFUSION, AND ABSORPTION [8]

Me You and his fellow authors has proposed a new method to create a realistic paint simulation, utilizing the characteristics of paint, such as fluidity, diffusion, and absorption. They have treated the painting elements separately as paper, pigment, solvent and binder. Adopting smoothed-particle hydrodynamics together with a consideration of viscoelastic movement, their method simulates the fluid motion of the paint and the solvent.

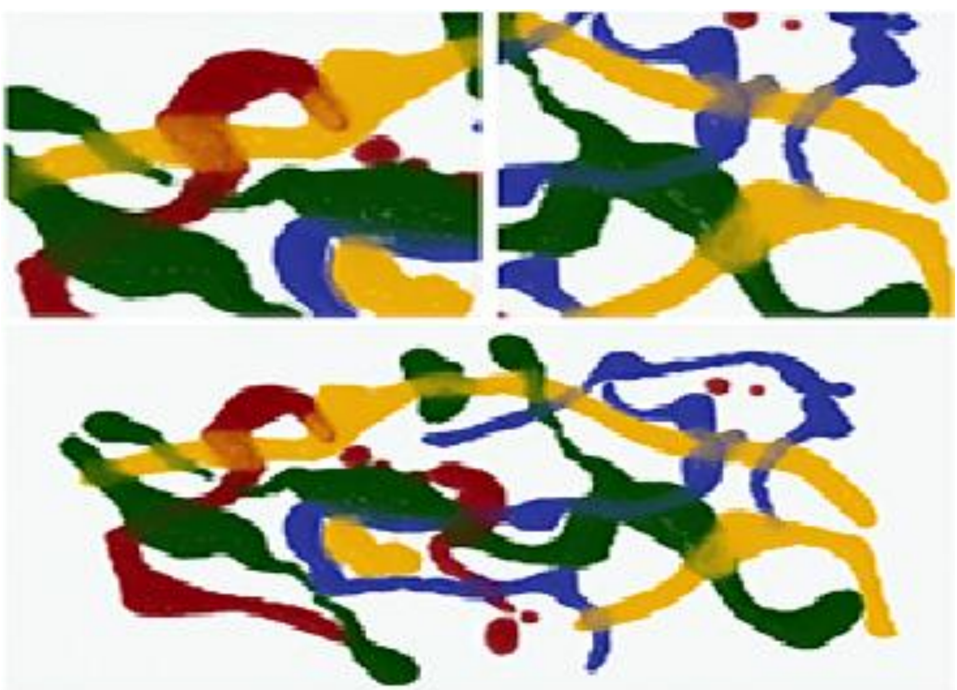


fig.17 Realistic simulation based on [8]

To handle the diffusion of the pigment in the solvent, they have developed the mass transfer method. Following the Fick's law, the concentration of pigment changes and each pigment particle is diffused to the neighborhood accordingly. As time elapses, the binder and the solvent are either evaporated or absorbed, and for the major sections, the pigment remains on the paper. The Lucas–Washburn equation determines the distance of absorption.

They have attempted to combine the NPR features of painting with fluid simulation. Incorporation of viscoelasticity into the SPH fluid simulation faithfully reflects the property of fluidity of painting. The absorption is also included to model the interaction of paint with paper. This approach allows our paint simulation to generate various types of painting.

Limitations

1. Information on interaction with paper is very limited, not allow for more detailed and natural expressions of paint effects.
2. Interactive applications such as brush interfaces have not been incorporated to great extend. Scope for inclusion of input from the sensors is very limited.

2.2.5. INTERACTIVE 3D FLUID JET PAINTING [9]

Sangwon Lee and his fellow authors have created an interactive model that allows the used to draw the paint droplet, splatter and 3D viscous fluid jets to draw in a Jackson Pollock style. Pollock painting is the kind of the color drawing where the paint color is utilized to form semi-random patterns on the canvas. To conceptualize the formation of the jet, they have proposed the two stages a Navier-Stokes solver for an axis-symmetric fluid column and a linked-mass system for tracking the three dimensional motion of the jet's axis line.

The paint trails left by the jets are represented using implicit surfaces. Their system also includes an algorithm for generating the splatter patterns created by the impacts of a high-speed fluid drops. They allow users to analyze the fractal properties of the images they create, comparing them to those known to exist in Pollock's own paintings.

fig.18 A result inspired by Autumn Rhythm: Number 30, 1950[9]

2.2.6. IMAGE-BASED COLOR INK DIFFUSION RENDERING [10]

Chung-Ming and his fellow authors has proposed an image-based painterly rendering algorithm for automatically synthesizing an image with color ink diffusion. They have given mathematical model with a physical base to simulate the phenomenon of color colloidal ink diffusing into absorbent paper. Their algorithm is a three phase process described as follows

1. **Feature extraction phase** – simplification of information about reference image by luminance division and color segmentation.
2. **Kubelka-Munk (KM) color mixing phase**- KM theory is incorporated to approximate the result when one pigment is painted upon another pigment layer.
3. **Color ink diffusion synthesis phase**- the physically-based model that we propose is employed to simulate the result of color ink diffusion in absorbent paper using a texture.

fig.19 Rendering by [10] method with different parameter values

The main features of our IBCIDR include:

- The color ink diffusion model is based on physical foundations. It simulates the diffusion of water on absorbent paper, the diffusion of pigment particles in the water, the diffusion of color associated with pigment particles, and the influence of gravity.
- The physically-based color ink diffusion model that we proposed improves and generalizes Kunii's approach, allowing us to simulate all colors of ink, not just black ink.
- The deposited layer and diffusion layer makes the diffused effects plentiful and make the diffused result consistent with the paper grain.
- The paper is generated by a patched-based texture synthesis technique, providing a visually realistic and plausible appearance.
- The simulated results are rendered by a non-SBR algorithm from light layers to dark layers progressively and automatically.
- The simulated result still maintains important features of the reference image.

The contribution of this work is that the proposed IBCIDR eliminates the limitation of conventional ink simulations, which only deal with black ink.

Limitation

- Although a physically-based approach has the benefit of realistic simulation, it also has a problem with computation.
- In Chinese ink painting, artists usually use a limited number of conventional pigments. But, a real image may contain more than a million colors.

The Author has written the feature article on the state of the state of mobile-NPR research. NPR techniques, implemented to encompass mobile computing unique features, are the main focus of this paper.

2.2.7. DROPLET: AVIRTUAL BRUSH MODEL TO SIMUALTE CHINESE CALLIGRAPHY AND PAINTING [24]

Xiao-Feng Mi and his fellow authors has present the virtual brush model based on a droplet operation to simulate Chinese calligraphy and traditional Chinese painting in real time. They have proposed the parameterized brush model that uses simplified evaluation algorithms to approximate the effect while maintaining the ability to express most of the special effects of Chinese calligraphy. They have derived the droplet model to simulate the tangent area between the brush and model because it resembles the real area.

(a)

fig.20 Result of brush with sparsely distributed bristles [24]

(b)

fig.21 Stroke stress with flying white [24]

Limitations

- In Chinese ink painting, artists usually use a limited number of conventional pigments. But, a real image may contain more than a million colors.
- Pattern has been approximated in the form of the “droplet” or “sibling of droplet” That does not really follow the real fluid simulation process, leading to have limited patterns.

2.2.8. INTERACTIVE VISUAL SIMULATION OF DYNAMIC INK DIFFUSION EFFECTS [25]

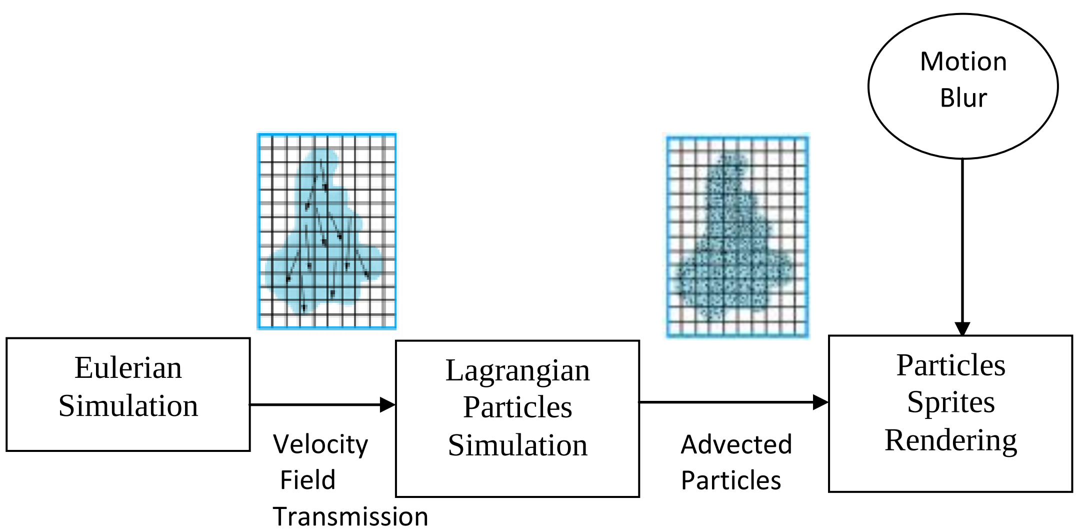


fig.22 A GUP-optimized Eulerian and incompressible Navier-Stokes A GPU optimized Lagrangian particles simulation; A hardware particle sprites rendering system with motion blur[25].

Shibiao Xu and his fellow authors have proposed the effective method that simulates the ink diffusion process with visual plausible effects and real time performance. They have combined the Euler method and Lagrange Method to define the hybrid grid particle. Both particles and grid cells represent fluid in their simulation. The particle is used to represent the ink molecules and corresponding grid cell is considered as a water molecule. Ink particle is affected along the water velocity so are kept in the region of grid water.

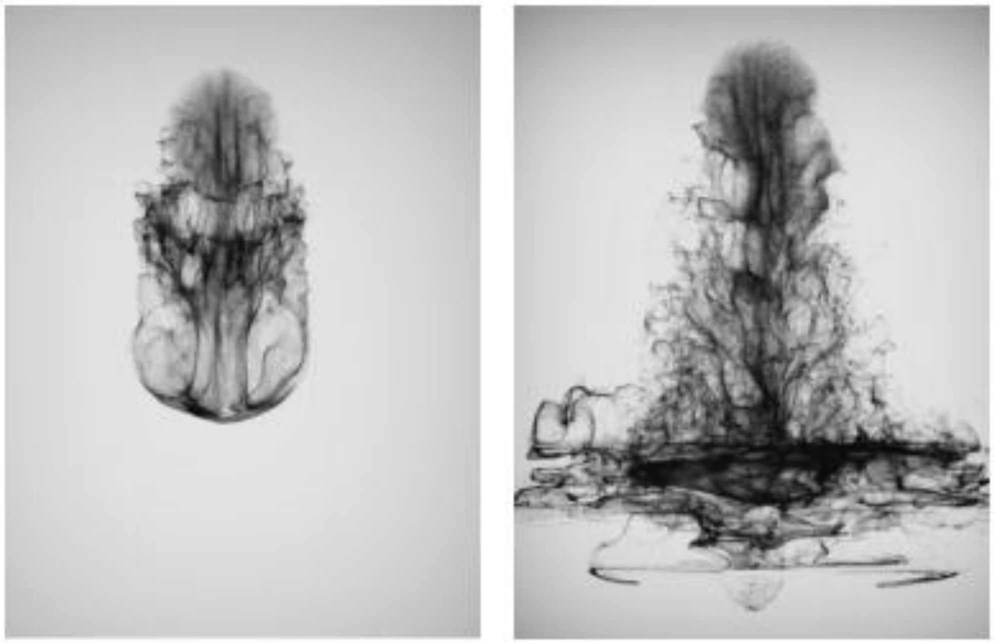


fig.23   Interactive Visual Simulation of Dynamic Ink Diffusion Effects[25]

2.3 CHALLENGES IN MOBILE NONPHOTOREALISTIC RENDERING [11]

Since last couple of years, high-end mobile device have been developed involving processors with 1 GHz plus capacity, GPU that can render millions of polygons per second, and long lasting battery power. However, overheating problem has been persisting since the evolution of the high end mobile. Additionally, most of the users use devices with limited processing and graphics resources.

Some NPR techniques such are stroke based rendering results degraded rendering quality, in lack of specialized graphics hardware. To meet the prerequisite style, innovative idea is required.

Issues related to Implementing NPR techniques for mobile devices, mentioned in the paper, are following

1. General issues: these issue points out the limitation associated with the graphics hardware and software generally. There are five general issues which are following
    I. Floating point unit (FPU) has not been incorporated in most of the mobile devices except high-end device.
    II. Graphics API, such as OpenGL ES, stores the vertex data in the vertex array. However, there is no depth buffer to store depth coordinate. Image processing algorithms based on depth buffer does not perform on a mobile base.
    III. There is no support for the either stencil buffer or hardware-accelerated fragment operations.
    IV. In most of the mobile devices, data catch's size is limited that adversely affect the rendering performance.
    V. For remote rendering, network connectivity fluctuations reduce the systems' reliability.
2. Application specific Concerns: these issues are related to the problem associated with implementing mobile NPR arose due to inability of mobile devices. These issues are mentioned below
    I. Involvement of the interaction with a user interface for the creation of the hand drawn strokes poses difficulties on mobile devices.
    II. Convenience of NPR content creation is directly proportional to the size of the screen
    III. Limited memory in device adversely affects the methods of image processing that store large images during the processing.
    IV. For the purposes of the 3D graphics, frame rate declines as scene complexity increases (e.g. Number of polygons or boxes in dataset.)

Chapter 3

PROPOSED APPROACH

- **Idea**
- **Simulation of Medium**
- **Spray Painting Framework**
- **Sprayer Control Panel (User Interface)**
- **Spray Color Selection Procedure**
- **Cellular Automaton Model for Spray Painting**

3 PROPOSED APPROACH

3.1. IDEA

Nonphotorealistic rendering in desktop and laptop doesn't have the advantage of inbuilt sensors, like those which are embedded in Smart phones. These sensors have the potential to enhance the power of the NPR, providing a wider scope for stylization. Exploiting the abilities of sensors to capture the various type of motion, I have proposed my approach to imitate the artistic spray painting. Therefore, in my approach, I have combined these sensor's functionalities with traditional unrealistic rendering methodology. The proposed approach has been focused on the simulating the airless water-color spray painting version of the image.

Basically, in Sprayer painting, a spray nozzle pours out color droplet of the specified radius at specific rate form the specific height (all these three parameters are specified in section 3.2.2), over the canvas. *Spray Handle* moves across the length and breadth of the canvas. *Spray Handle* moves as user shakes the mobile that exposes the different area of the canvas to spray nozzle. This way color spread in the form of the Spray droplet across the length and breadth of the canvas.

Suggested approach for the Spray decorated painting is Particle based methodology (cellular automata). Cellular automata (CA) are easy mathematical idealizations of convoluted schemes. They comprise of a lattice of discrete, identical sites (cells) holding random standards. These standards develop over discrete time steps, ruled by directions working out each cell's interaction with its neighbors.

1.2 SIMULATION OF MEDIUM

Simulating artistic medium can be further classified into those that simulate the physics of art and one that create only the look and feel of a particular medium technique. In my approach I have embodied the former method. Smart device has verities of sensors. Therefore, given the wider range of inputs from these sensors that are also highly précised in monitoring and capturing the motion in the form of the numerical value, simulating the physics of the spray art is more possible with Mobile NPR. Simulated medium used in my approach is followed:-

3.2.1 CANVAS

The canvas, in my approach, has been modeled by comprising a random fiber network proposed by Kallmes and Vote in 1960 [28]. The model was proposed to be used in research in Paper manufacturing. Later it was applied in many digital painting simulations.

For more realistic Canvas effect, the shape of the fiber is formed by a curved line segment using a sine function which is smooth, with additional advantages of simple computation and symmetric nature about the origin [27]. The properties of fiber allocation are confined in such a way that the local position and orientation of fibers vary randomly while the global property of fiber distribution is homogeneous. Considering all these prerequisite properties, a fiber mesh is developed that contain all these properties by dividing the Canvas plane into square regions, called cells, and placing the fiber according to rule; i.e., for each region the average fiber density is the same, although within each region, fiber centers and orientations vary randomly.

Region I selected a cell of square dimension because it is the simplest geometric shape which allow to cover whole rectangular sheet of the Canvas, unlike none corner areas which would not cover the whole area [27]. Regarding the optimal size of the region, which is a significant parameter to obtain stability of the global homogeneous and local random property of the Canvas, I have selected double the length of a fiber as the length of a side; a size in which each region can contain the segments of fibers that pass through, cease at, or lie completely within.

.

A fiber is defined as

$$y = c \sin\frac{2\pi x}{l}, \frac{-l}{2} \leq x \leq \frac{l}{2} \qquad (3.1)$$

Where c is the curvature of a fiber and *l* is its length. Fibers with various curvatures and lengths can be obtained by changing these values. A fiber can be oriented and located at a new position by applying 2D transformations to the coordinate vectors.

To ensure the local fiber density and orientation are random while the global density and orientation are uniform, the large Canvas field is evenly subdivided into small rectangular sub-areas having the same average fiber density, although the fiber distribution varies randomly from one Subarea to another.

Once fibers are distributed over the Canvas field, a fiber mesh structure is determined at every pixel of the Canvas such that each one can be represented as a data structure, i.e.

```
Class cell {
    int cellx, celly;   // coordinate
    short captub[8], // capillary tubes
    Captub-sum;
    Short cross-pts; // number of cross points
    Short water, granule, diff_type; // liquid ink characteristics
    }
```

Where the cell P has eight neighbor cells $P_0 \ldots, P_8$, captub[i] denotes the number of fibers passing P_i and P_j, and *cross-pts* denotes the number of cross points where two or more fibers cross each other (Fig. 6). It is assumed that the number of capillary tubes connecting P_i and P_j is proportional to the number of fibers passing P_i and P_j .

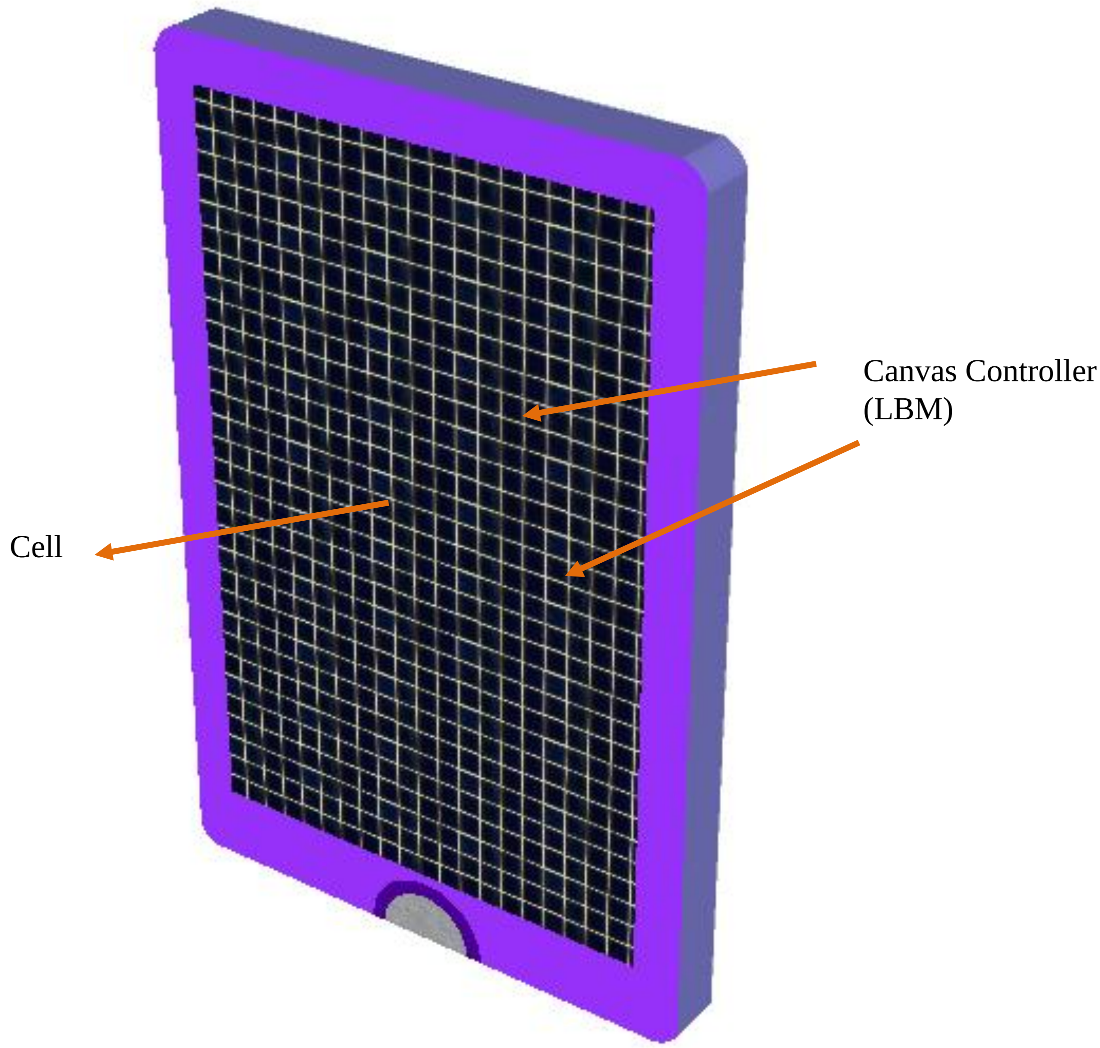


fig.24 Cellular Model Of Mobile Device

3.2.2 MODELING SPRAYER NOZZLE

All the existing stroke base NPR techniques are based on the Virtual models. One of the earliest attempts was Strassmann's proposed brush model. Therefore, Stroke trajectory was derived from a parameterized model of brush, conceiving brush hair properties and the diversification of ink deposition model.

To differentiate my approach, I have proposed the concept of the virtual parameterized Sprayer. Motive of approach is to simulate the art of spray painting. This virtual sprayer uses the simplified evaluation algorithm to approximate the shape of the droplet volume and a drop falling phenomena.

Basic virtual sprayer components are following:-

1. **Sprayer handles:** It is the component the spryer which provides the interface for passage of the interaction information in the form of the parameter from either user and Canvas and sensors. I have defined three types of the interactive activities. Like Dipping brush activity, prominently used in most of the virtual brush modes, I have defined the *Spray container filling activity* (described in the second point) to fill the container (2 dimensional array). Second activity *is Positioning sprayer and spray generation activity* triggered by the spray pressure generator (described in third point). The third activity is the *spray droplet formation process* dependent on the size of the sprayer nozzle hole. Unlike brushes, in my approach Spray handle remain rigid its position. User guided mobile shaking activity makes canvas moves around the *Spray Handle*. *Sprayer Handle* is informed about the position of the canvas below it. Position information is updated as there is the any change in the canvas orientation.

   Class sprayHandle

   {

   int cellX, cellY *// position of the cell laying at the center of the Sprayer nozzle face*

   int height *// Distance from the canvas to spray handle*

   Boolean active // *set the status of the Spray process either active or inactive.*

   }

2. **Sprayer color container:** Like brush-dipping process to hold the color, I have innovated Spray container filling process. Characteristics of brush is that all the bristles hold the same color which is highly unlikely to Spray generating process as in a spray process color is bound to disperse to larger area by nature. This is highly erroneous in terms of holding the basic look and feel of the referenced image. To reduce the deformation factor, I have proposed the idea of keeping color in spray nozzle outlet point based on the its current position upon the canvas. Therefore, container is divided into the section (array list). Where each section is assigned to an individual *Spray outlet hole*. Therefore, at a time, the different Spray outlet hole may contain different color. Assignment of the color to a section depends on the position of the *Spray outlet hole* over the canvas.

   Class SprayContainer

   {

   int PThreshold //

Struct nozzleContainter

{

int pixelX , pixelY; *// position of the pixel below at the center of the nozzle outlet hole*

int color ;

}

Arraylist < NC> nozzelCointainer;

}

3. **Sprayer Pressure generator:** Real spray device has the internal mechanism to generate the pressure that is required to be applied to the container in order to spray. By the application of the pressure spray process starts. In my approach I have derived the definition of the pressure from Spray rate parameter and the shaking movement of the mobile. If the shaking frequency is higher than it is translated into to the more pressure.
4. **Sprayer nozzle face:** it is the interface that represents the *Sprayer outlet holes* distributed randomly over it. A *Sprayer outlet hole* is not allowed to overlap the position of any other *Sprayer outlet holes*.

   Class nozzle face

   {

   Random R; *// randomization of how the Spray outletholess are embedded on the face of the sprayer*

   int   n; *// number of the nozzle outlet holes*

   int radiusF // radius *of the Spray nozzle face*

   }

5. **Sprayer outlet holes:** *Outlet hole* is the source for the color to come off the Sprayer. The size of the *outlet hole* defines the initial radius of the *droplet.*

   Class nozzleOutlet

   {

   int radiusN}

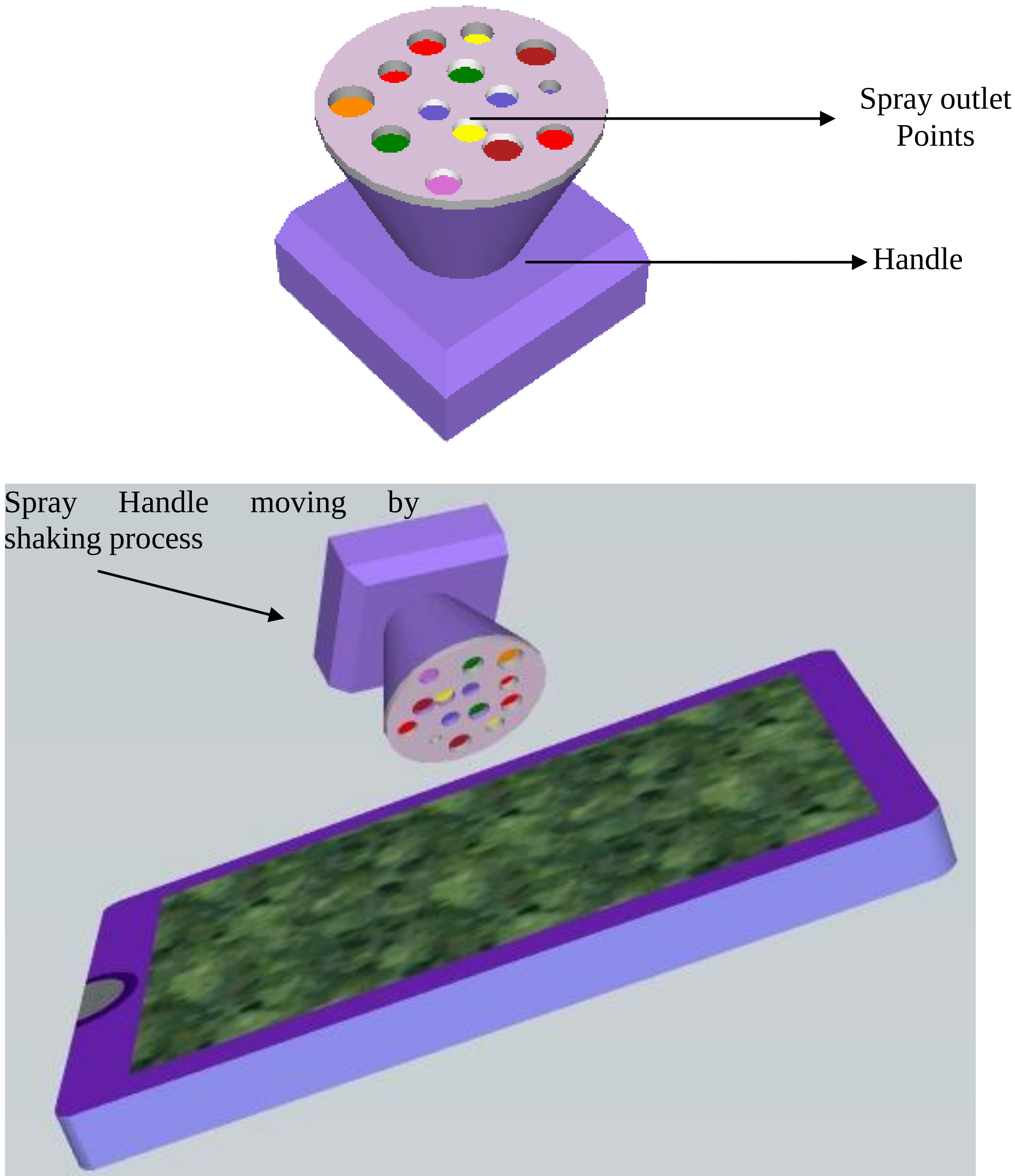


fig.25 Spray handle and position of the spray handle over canvas

3.2.3 INK AND ITS PROPERTIES FOR SPRAY PAINTING

In my approach I have given the option of only one type of color – watercolor. Color is defined in terms of the viscosity (v), mixing nature (э), solid liquid interaction nature. Color droplet laying on the canvas surface shows their nature when the user shakes or tilts the mobile. As the result, droplet starts moving on the surface. So it might be the possibility that it would collide with the other droplets. Therefore, my approach decides how interaction will take place. Similarly, the color droplet behavior under the influence gravity, when it detaches from the *Spray outlet hole*, is also defined .

Diffusion of the ink is perhaps the most admired features of painting falling in the watercolor category. The appearance of the spray droplet becomes more realistic by the presence of halos around the droplet which adds mysterious touch to it. This halo is drawn by letting the ink spread beyond the droplet original border, while ink passes slowly into special paper with high absorbency creates a feathery, blurred edge. These diffusion features are complex physical phenomena which cannot be accurately imitated on canvas by simulation conducted degradation functions, fractals, or texture mapping techniques, since purely mathematical methods generally result in flatly blurred images which are different compared to realistic diffusion images.

To develop a suitable framework for simulating ink diffusion, there is a need to focus on the physical mechanism. The typical canvas is web of fiber. Gape in between the fiber forms the holes acts as the thin capillary of tubes for holding water away from the initial area. The color pigments float and move in this liquid due to collisions with molecules of water and other color pigments. Accordingly, an appropriate simulation of ink diffusion must include careful modeling of the following aspects:

1. Fiber material and mesh structure of the paper(cellular model of canvas),
2. Quantity of water,
3. Ink density at a point,
4. States of the surrounding points.

Depending on which type of paper is used, the ink absorbency, diffusion directions, and diffusion patterns will vary. The type of fiber material and fabrication of fiber are responsible for this [27].

Ink particles are carried away from the initial area by water. As the water along with the ink particles move, some of the water is absorbed by the fiber. The rest of the water continuously flows along the fiber until it is completely absorbed. The quantity of water, and not the density of ink, accordingly determines the span of the diffusion of the image or the number of diffusion steps.

Under the influence of the motion of water molecules, suspended-ink particles move in a manner called Brownian motion. Diffusion is affected by the difference of density of ink particle across the regions, transferring the ink particles from the region with higher density to the region with lower density. The density of liquid ink surrounding a particular point will therefore determine the direction in which diffusion takes place.

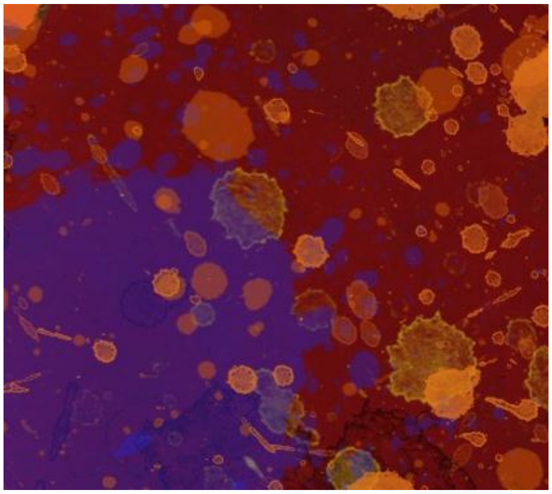


fig.26 Image of spray splatter

fig.27 Image of droplet

3.3 SPRAY PAINTING FRAMWORK

The model has approximated only those properties of fluid motion that are important for the look of watercolor spray painting appearance, and ignored the rest. The turbulent motion of the fluid is not needed for painting applications.

The artist will start by creating a mental layout of how he desires his image to proceed. One needs to understand where exact details will go (such as satellites, trees, mountains, stars, haze, water, etc.)in alignment of colors appropriately in the right places. The method is almost like paint rearwards; the creative person must lay the paint for the painting's foreground things first.

Fig 24 demonstrates the proposed Spray Painting Framework that shows the user parameters, sensor inputs, *spray handle*- process, color selection process, various stages of simulation and how all these are interlinked with each other.

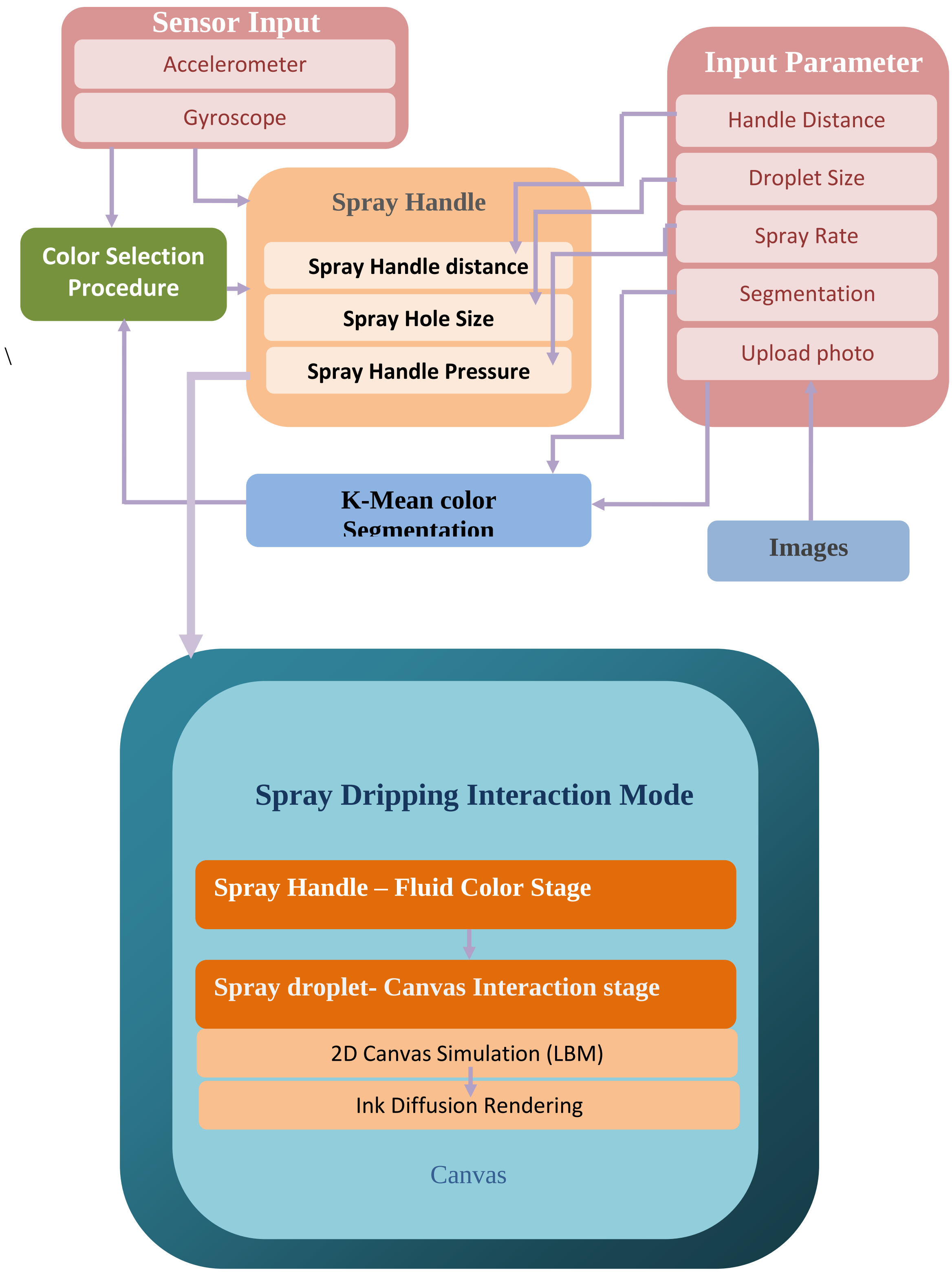


fig.28 Spray framework

3.4. SPRAYER CONTROL PANEL (USER INTERACTIVE)

My approach is user interactive. Three parameters are defined whose value is taken from the user. These parameters are followed-

1. **Handle distance (h)** – Handle distance means how far *Spray handle* is from the canvas. This parameter is essential because when droplet, coming from a *Sprayer Outlet Hole*, drops to canvas surface, deformation of the droplet becomes directly proportional to Handle distance (h).

$$Deformation(droplet) \xRightarrow[propotional]{} h$$

2. **Droplet Size (d)** - Droplet Size mean the radius of the Droplet . Droplet with bigger size spreads more than a droplet with small size. Reason is that bigger radius Droplet has more color, heaped on small volume. So, when the user shakes of tilt the screen, bigger droplet has more tendency to move and greater potential to overcome the surface tension. Droplet size is determined by the size of the *Spry Outlet Hole*'s size.
3. **Spray rate (r) -** Spray rate can be defined s number of droplet falling on a cell per unit of time. Spray rate has a vital role in the painting process. In my approach, Spray rate is the factor that is used to derive the internal sprayer pressure in *Spray Handle.*

3.4.1 SPRAY PROCESS

The spray process consists of the following activities-

1. Upload photo – user first click the upload button to select the desired photo of any format that the user wants to convert into spray painting.

2. Set Parameters-

- **Handle distance**
- **Droplet Size**
- **Spray rate**

3. Segmentation – User will provide the information about the segmentation level. Increasing the segmentation level, users can get the more detailed Spray version of the original image.

4. **Click Spray process-** After setting up all the parameters and uploading photo. User clicks the Spray button. After clicking a button. Spray control Panel disappears as the whole screen is given to canvas.
5. **Shaking mobile** –Shaking process is the movement process that approach has incorporated, that is needed to cover the whole canvas for being sprayed all over.

The user is allowed to shake mobile in any direction.

The motive of the Spray process is to drop color on the surface of the canvas. In my approach, I have tried to create the realistic version of the Spar process, seeking for real physical phenomena.

All the realistic Phenomena in the case of the Spray painting are following.

- ❖ Depending on the size of the droplet and the height of the spray handle, droplet spreads over the surface in all the directions after striking with the surface.
- ❖ Droplet on the surface of the canvas tends to flow on the surface due to shaking process.
- ❖ The droplet will have certain surface tension that it needs to overcome to flow.
- ❖ Droplets will also act under gravitation force.
- ❖ **Time Phenomena -** I have associated the time factor with the wetness of the droplet. Droplet takes certain time to dry on the canvas surface. Once the droplet is dry, then it does not intend to follow. Rate of dry is fixed in my approach that is different for the all oil and water. That implies that depending on the volume (radius), droplet having bigger radius dry longer than droplet with smaller radius.

3.5 SPRY COLOR SELECTION PROCEDURE

It is a crucial stage in the spray process. Strength of approach in terms of the non photorealistic color artifacts lies on how to pick the color. The basic concept is that we select the color from the image. For the artistic variation purpose, I have incorporated the concept of the image segmentation. Varying the segmentation level will give a different look to the resultant image produced after the spray process.

I have deployed the K-mean clustering algorithm of the image segmentation [15]. The K-Means algorithm is used to find the cluster within the data based upon the various parameters. The cluster can be formed in the image based on the Pixel intensity, color, texture, location or some combination of these. For K-Means algorithm the starting position is very important and can make a difference between the optimal and non optimal solution for both are susceptible to termination when achieving a local maxima as opposed to the global maximum. The K - Mean algorithm can be thought of as algorithm relying on the hard assignment of information to a given set of partitions.

At each pass of the algorithm, each data value is assigned to the nearest partition based on the some similarity parameters such as Euclidean distance of intensity. The partition is recalculated based on the hard assignment. With each successive pass, data value can switch partitions, thus altering the values of the partition at every pass. K-Means algorithm quickly converse to a solution quickly as opposed to the other clustering algorithms. Having two options, either clustering by RGB color of each pixels or pixel intensity. I have chosen RGB color.

Algorithm 1: K-Means Algorithm

Input: $E = \{e_1, e_2, e_n\}$ (set of entities to be clustered)
    k /* number of clusters */
    Maxlters /*limit of iterations*/
Output: $C = \{c_1, c_2, \ldots, c_k\}$ /*set of cluster centroids*/
    $L = (l(e) \mid e = 1,2, \ldots, n)$ /*set of cluster labels of E */
Foreach $ci \in E$ *do*
    $c_i \leftarrow e_i \in E$ /*e.g. random selection*/
end
foreach $e_j \in E$ *do*
    $l(e_i) \leftarrow argminDistance(e_i, c_j)\ j \in \{1\ldots k\};$
end
changed ← *false;*
iter ← *0;*
repeat
    foreach $c_i \in C$ *do*
        $UpdateCluster(c_i);$
    end
foreach $e_i \in E$ *do*
    $minDist \leftarrow argminDistance(e_i, c_j)\ j \in \{1\ldots k\};$
    If $minDist \neq l(e_i)$ **then**
        $l(e_i) \leftarrow minDist;$
        changed ← *true;*

In trying to improve the runtime and results of the algorithm, partitions which are identically positioned were chosen. Primarily, eight partitions were chosen. The partitions comprised red, green, blue, white, black, yellow, magenta, and cyan or the corners of the “color cube”. With the primary partitions selected, the algorithm can begin. Every pixel in the input image is compared against the primary partitions and the closest partition is selected and noted. Then, the signify in periods of the RGB hue of all pixels inside a granted partition is determined. This signify is then used as the new value for the granted partition. If a partition has no pixels associated with it, it remains unchanged. In some implementations, a partition with no pixels affiliated with it would be taken; however, these partitions are simply disregarded in this implementation. One time the new partition standards have been very resolute, the algorithm returns to assigning each pixel to the closest partition. The algorithm continues until pixels no longer alter which partition they are associated with or, as is the case here, until none of the partition standards alterations by more than a set small allowance. In alignment to accumulate more results, the initial partitions utilized were diverse by supplementing and removing partitions.

I have described that spray handle remains rigid and the canvas around it moves.

Spry Color selection procedure steps are given below

1. The procedure starts from the center position of the screen where the *Spray Handle* is initialized.
2. As particular area of the canvas is exposed to *Spray Handle*, Spray handle demand the corresponding area of the cited image, corresponding to coordinates in canvas, to fetch the color from the image.
3. From the corresponding Pixel position on the image, all the *Spray Outlet Holes* present on *Spray Nozzle Face* of the *Spray Handle* pick the color.
4. Color in *Spray Handle* then mixes with the selected fluid with the specified viscosity, diffusion rate, for the purpose of creating the watercolor spray painting version.

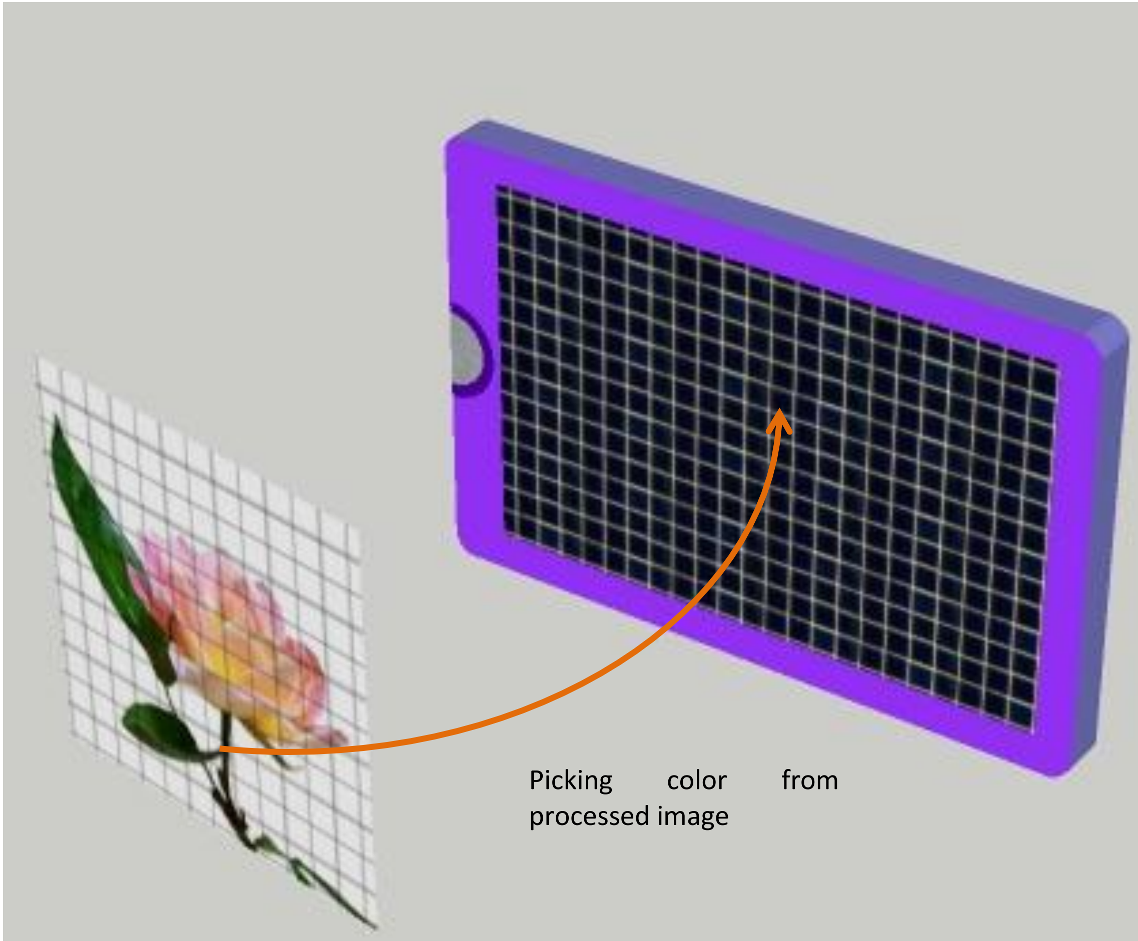


fig.29 Procedure of color selection

3.6 CELLULAR AUTOMATON MODEL FOR SPRAY PAINTING SYSTEM (SPS)

I have proposed the cellular Automaton Model for Spray Painting. As Spray process starts, it Continuously drops the wet color droplets on the canvas face . Droplet of the color drops on the cell surface that is currently lying beneath on the canvas surface that has come to this position after shaking. The size of the droplet on canvas is determined by the number of cells over the surface of the canvas, the droplet will spread on.

There is one more thing that needs to understand that color droplet is consisting of the color pigment and the fluid.

Therefore, each cell holds values that denote the amount of paint present in that cell. This value represents the quantity of fluid and a quantity of pigment. Each cell also carries values for the amount of pigment and fluid that has been absorbed into the paper at that cell. The motion of paint on top of the paper, "*Surface effects*", and the motion of the paint within the paper "*Substrate effects*" are dealt separately, and paint can move from the surface layer into the substrate layer

The motion of a color droplet in the surface layer is influenced by a displacement force, F_d, which is evaluated for each cell in both the horizontal and vertical directions . F_d is consist of the forces of surface tension and "spreading" (diffusion) and Gravity. It also implies that Fluid residing in the particular cell is pulled toward the fluid over the near big heap of the fluid to balance the level of fluid. In the *surface layer* pigment moves along with the fluid in equal proportion to it. At Each time step, surface fluid value is updated for each cell by first reducing from the last step's value, the fluid coming out of the cell because the displacement force, F_d, and then adding on the fluid moving into the cell from its neighbors due to their displacement forces.

3.6.1 MODELING SPRAY PAINTING USING PARTICLES

System of particles cannot directly model fluid motion. Calculation of movement of the water containing color pigment is based on the assumption that flow is continuous and particle, by nature, is discrete. In this case, however, it is not necessary to have even a fairly precise simulation as long as the result looks pleasing to the casual observer. In fact, it is perfectly adequate to have the particles' motion defined by a static vector field representing the texture of the paper, as this is the most important effect on the fluid movement.

In the proposed model, shaking is the movement being exerted on the fluid as an external force. Particle within the droplet comes in motion due to shaking process. Let D is the direction of the shaking of the mobile by the user. As the user is allowed to shake the mobile freely, Direction of shaking updates quickly.

In the proposed Model, Spray color is actually consist of the particles, Each Particle is Represented set of the values, which are following.

1. Wetness (Amount of Fluid)
2. Color

Each particle is big enough to fit in one Pixel space. Wetness factor is responsible for the flow of the particle which also carry the color with it.

Spray painting system (SPS) has been designed with two stage spray simulation framework. The two phase simulation skeleton simulates following realistic and organic spray process:

- **Spray handle – fluid color stage:** Simulation process at this stage is begun depending on information on the values of parameter set by the user by the user interface. The user sets the rate of spray, sprayer nozzle radius. This stage simulates the dynamics of Sprayer handle and loaded color fluid.
- **Spray droplet- Canvas Interaction stage:** This stage starts after color droplet hits the canvas. I simulate the dispersion and absorption of the spray color droplet on canvas using the Lattice-Boltzmann methods (LBA).

In our approach movement of the ink through the path because of air movement and dynamics is approximated (air-less method). The effect of the air has been neglected for all the cases.

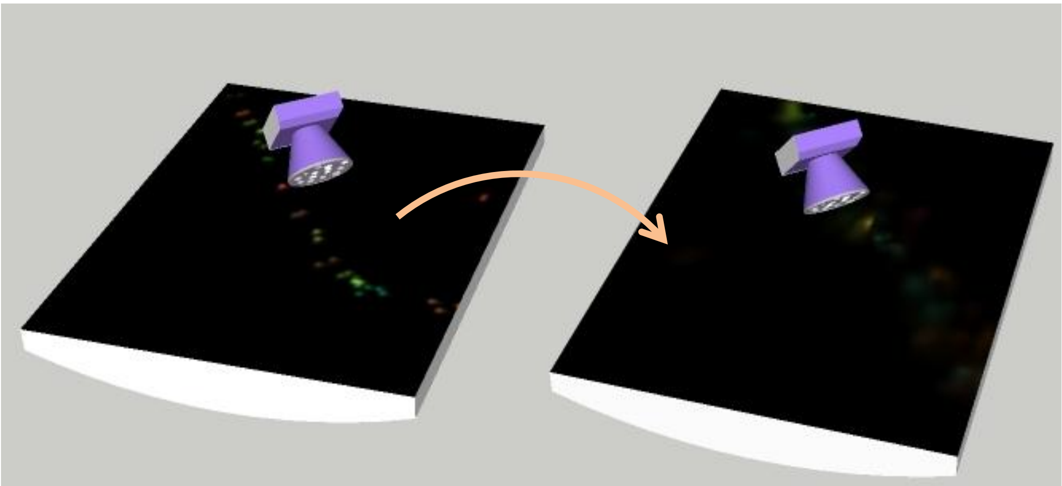
3.6.2 SPRAY DRIPPING INTERACTION MODE

Before I start explaining above mentioned 2 stage process, there is a need to understand Spray dripping interaction mode. The artist creates the spray painting art with the dripping type of motion of the color droplets from the Sprayer. In dripping mode artist load the sprayer (attached to the container) with the Liquid color. Some sprayers have also option for setting the speed of the spray (Spray rate). The sprayer has the inner mechanism to set the pressure on the color. Pressure can vary within a particular range. Change In pressure directly affect speed of the color droplet coming out of the *Spray outlet holes*. After color droplet leaves the sprayer handle, gravity does the most of the work. One can direct the flow in the mid –air by motions, but I have ignored the air stage as I have focused on airless method. The fig .30 shows the Spray created by the dripping motion.

I have proposed our approach for the smart phone and tablet. I have incorporated the sensor inbuilt in device for the motion capturing purpose.

Modeling spray color dripping motion using a Smart graphic tablet and phone, first we set out to understand the input data that they can give. A Smart device informs us with five kinds of meaningful data that is useful for painting context: Location of object on Screen (2D), tilt and bearing (in degrees), *Spray Handle* control coordinates in Three dimensional space at any location l, where $\mathbf{l_x}$, $\mathbf{l_y}$ are the corresponding ***cellX*** and ***cellY*** coordinate of Handle on screen. For the Spray handle motion purpose, we have utilized the motion sensors, Accelerometer captures the motion of the device. This motion is directly translated into motion of the Spray handle across the Canvas. So the velocity of the *Spray Handle* (q_{speed}) is calculated by measuring the displacement of the **l** between the two time steps . The coordinates of **l**, q_{speed} are updated at each timestep from the Spray handle. Using these data, I have deviced a controlling scheme for spray color dripping interface.

Color dripping stage emulates a *Spray Handle* motion controlled spray –jet that pours out color droplets based on the Handle angle. Unlike the common digital painting software for tablets and phones where stylus pressure on the surface controls the paint nozzle, in our approach user will set the *Spray Outlet Hole* size at any point in time during the spray process. The amount of paint being released is constant. . It means that all the droplet of the particular size will have fixed amount of the color pigments. To create a directional spray droplet, We use angle (tilt angle) to control the orientation of the Spray path. $\mathbf{q_{speed}}$ is used to alter the velocity of jet to simulate momentum.



.

fig.30 The input I received from the device, and corresponding path of spray formed by the motion of the handle. A sudden application of pressure creates a fast shacking motion

I have given emphasis to the fact that as the artist does like to have the control where the droplet should be landed; I have bound the spray handle position to the shaking motion of the device, captured by the sensors. So the position of the paint source is calculated and updated accordingly.

In my approach, there is the cursor whose position is controlled by the *Sprayer handle*. The Spray rate (velocity) and the inclination angle of the device control the size and the direction of the spray droplet. This in turn gives the control over the final size and the spray direction of the spray pattern, so if the **p** is the target location on the canvas and **u** is the position where the Sprayer handle is currently staying on ,The position of the spray jet is calculated by:

$$\boldsymbol{C_z} = \boldsymbol{H_{dis}} \qquad (3.2)$$

$$\boldsymbol{C}_{(x,y)} = \boldsymbol{P}_{(x,y)} + \boldsymbol{U}_{(x,y)} \times \boldsymbol{t_d} \qquad (3.3)$$

$$\boldsymbol{t_d} = \frac{2C_z}{U_z + \sqrt{U_z + 2gC_z}} \qquad (3.4)$$

$\mathbf{t_d}$ is the expected time a droplet takes to fall on the canvas after having been thrown out from the Spray nozzle. I have conceived our formula form the basic equation of motion to neglect the other detailed calculation, having no substantial effect on the overall formation of Spray droplets over the canvas. While each particle in the spray jet does not have the tendency to fall on the exact target location, the drop path does not deviate too much from the basic equations of motion.

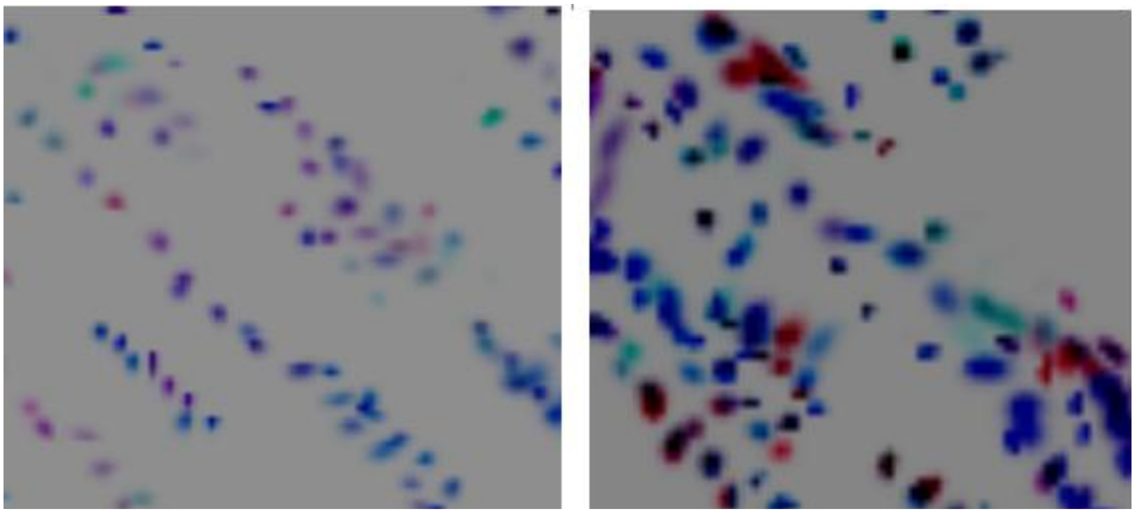
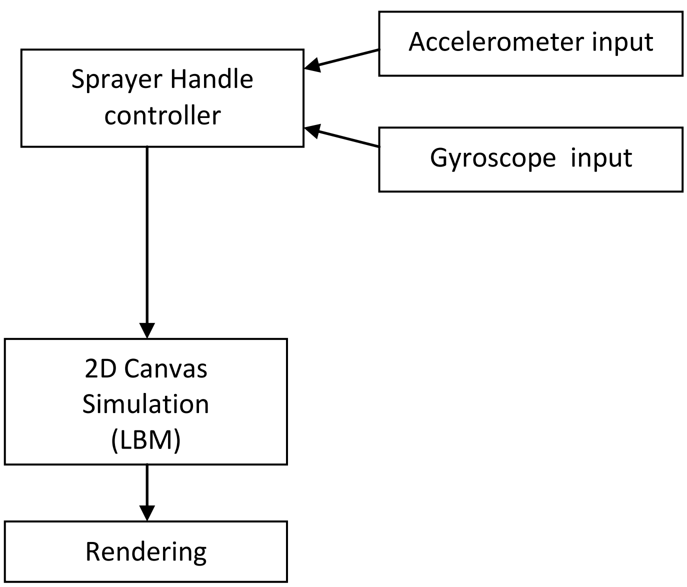


fig.31 shows spray patterns derived through same path but initiated with different parameters.

3.6.3 SPRAY HANDLE – FLUID COLOR STAGE:

Physical behavior of the *Spray Handle* is required to simulate before the colored fluid particle are created.



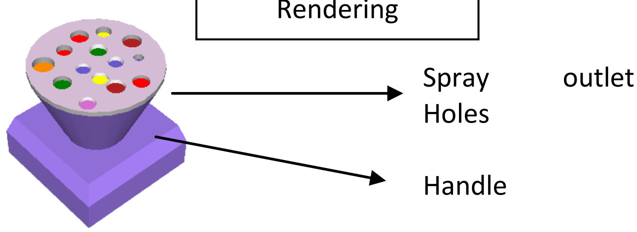


fig.32 Fluid color Stage

In my approach I have not limited the amount of color that sprayer can Spray . I have assumed that color droplet come infinitely out from the spray nozzle until the user stops the process. The radius of the nozzle outlet is a parameter that is set by the user before the spray process start. Particle emitted through the S*pray outlet hole* are loosely bound to fall within the area of the on the canvas. Therefore, I have set the range in terms of cone angle **ϕ** . This means droplet are needed to fall in the area derived from an angular projection by angle **ϕ** from the periphery of the *Sprayer nozzle face* to canvas separated by *Handle distance.* Therefore, emitted particles are radially within a circular cone with fan angle **ϕ.**

The advantage of the spray process for the splatter color artist who ticks paint over the canvas with normal brush is that calculation required for the simulation of the Brush spine, surrounded by the color of dripping, does not require in Spray process. By adjusting the parameters our approach can emulate the Ink splattered pattern with less calculation, saving time.

To model spray handle Loaded with the color, each Spray outlet hole is given a certain amount of paint V_i , for a single timestep **t.** V_i is dependent upon the radius of the sprayer outlet hole.

Dispersion nature of the fluid color is proportionate to the density $\boldsymbol{\rho_o}$. So therefore the weight of the droplet with the density $\boldsymbol{\rho_o}$ of particles is:

$$W_{color\_i} = \rho_O Vol_i \tag{3.5}$$

Where V_0 is the volume of the droplet

$$V_i = \frac{4}{3}\pi r^3 \tag{3.6}$$

Where r is a user defined parameter (*Droplet radius*).

In Spray handle model. Droplet contain particles is detached from the sprayer handle outlet whole if force $F_i = ma_i$, generated by the pressure simulated by the inner mechanism of the sprayer, on the outlet hole is equal to or greater than threshold φ_i. So the actual amount of paint in color droplet leaving a spray droplet hole would be:

$$W_{detech} = W_{color\_i} - \varphi_i a_i \tag{3.7}$$

Figure 26 shows the process of color droplet detaching. The threshold is the function derived from fluid viscosity $\boldsymbol{\mu}$ measured in kg per meter second, surface tension $\boldsymbol{\sigma}$ measured in Newton per meter, And the current segment length $\boldsymbol{li}$ measured in meter

Viscosity is the internal friction of the particles of the fluid. Density and the Viscosity negatively affect the color droplet detachment process. Surface tension also inversely affects by pulling the fluid inward to prevent the detachment process. Therefore, threshold can be derived as:

$$\boldsymbol{\varphi_i = k_{d_1} * \mu + k_{d_2} * l_i(m) * \sigma} \tag{3.8}$$

Where $\boldsymbol{k_{d_1}}$ and $\boldsymbol{k_{d_2}}$ Are detaching constant that affect the quantity of paint (fluid), leaving in the form of color droplet.

Now I have sorted out quantity of paint detaching from the spray outlet hole at each timestep

So I required to create resultant particles of color for the SPH simulation. In SPH., how particles are placed within system is very important for the system stability. Therefore, the placement of the particle must be uniformly distributed according to density and mass of the particle (m) of the fluid:

$$\boldsymbol{d = \sqrt[3]{\frac{3m}{4\pi\rho_O}}} \tag{3.9}$$

I introduce the simple method to initialize the color particles. For each Spray outlet hole, a color droplet of uniformly distributed particles with radius $r_w = \sqrt[3]{\frac{W_{detach_i}}{\rho_O}}$ is formed.

Since overlapping of the particles creates undesirable effects, I have taken care it by certain measures. A particle is only generated if there is no other particle within interaction radius **h.**

The initial speed of the particle in the droplet is equal to the speed of spray nozzle that detaches the paint. The look of the initial droplet volume is smoothened by SPH simulation in about 4 time steps.

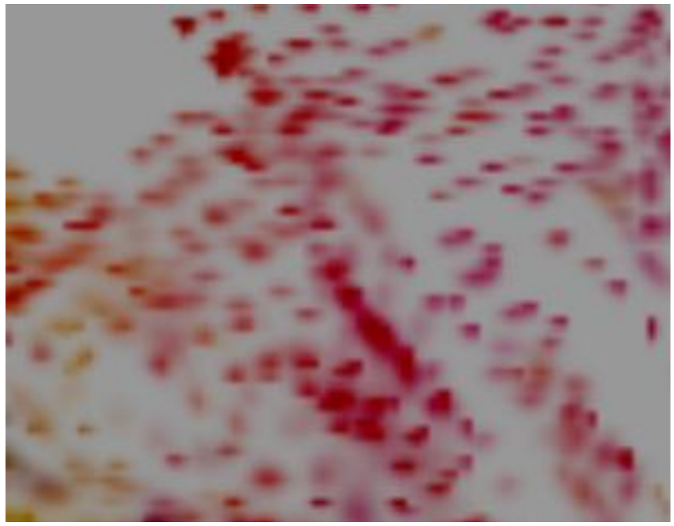


fig.33 the approximate representation of color droplet formed by the particles.

Inclination of the handle governs the direction of the color droplet leaving the Spray outlet points. The speed of the *Spray Handle* movement also affects the direction and the power of the paint thrown. Initially a set of particles is initialized with the initial velocity u:

$$u = u_x\vec{\imath} + u_y\vec{\jmath} + u_z\vec{k} \tag{3.10}$$

$$u_x = \left(S_{velocity}Q_{velocity}\frac{|\overline{l_{\iota-1}}|}{t}\right)_x \tag{3.11}$$

$$u_y = \left(S_{velocity}Q_{velocity}\frac{|\overline{l_{\iota-1}}|}{t}\right)_y \tag{3.12}$$

$$u_z = \left(S_{pressure} * Q_{pressure} + S_{tilt}\right) \tag{3.13}$$

Where l_i is the position of the spray nozzle whole at timestep i. $S_{pressure}$ (pressure in *Spray Handle* generated by internal mechanisms), S_{tilt}, $S_{velocity}$ (velocity of handle) are user adjustable parameters, depending on shacking and Q_{tilt}, $Q_{velocity}$, and $Q_{pressure}$ which are tilt vector, velocity vector and pressure of the individual *spray outlet hole* on *Spray Handle* respectively, translated from shacking processing process of *Handle*. t is the length of the timestep.

3.6.3.1 POINT DROPLET

How particles are rendered on the canvas determines how realistic and effective result would be. I have incorporated [34] to render the particles impression on the canvas with the following strategy:

$$u_{canvas}(i) = u_x(i)\vec{j} + u_y(i)\vec{k} \quad (3.14)$$

Direction of axis of droplet

$$M_1(i) = \frac{u_{canvas}(i)}{|u_{canvas}(i)|} \quad (3.15)$$

$$M_2(i) = M_1(i)^{\perp} \quad (3.16)$$

$$r_{M_2}(i) = (r_w + u_z(i) \times ran(0,1)) \quad (3.17)$$

$$r_{M_1}(i) = (r_w + |u_{canvas}(i)| \times ran(0,1)) \times r_{M_2}(i) \quad (3.18)$$

Particles make the cluster to form the droplet. When the droplet hits the canvas, many particle end up being absorbed.

For each droplet i, $u_{canvas}(i)$ is the velocity projected on the x-y plane (canvas).All the particle in the droplet is uniformly given the same velocity $u_{canvas}(i)$, The spray color droplet takes the shape of the ellipse with its major axis $M_1(i)$ on , $u_{canvas}(i)$. The length of its major and minor axis are $r_{M1}(i)$, and $r_{M2}(i)$ respectively (fig30).

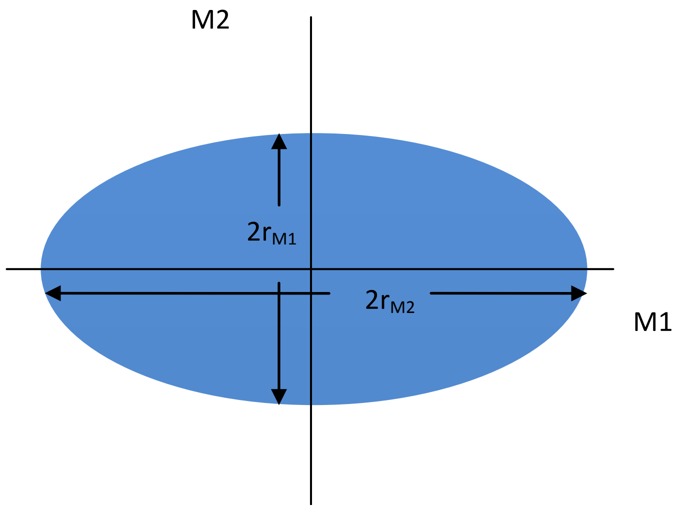


fig.34 major and minor axis are $r_{M1}(i)$, and $r_{M2}(i)$

I have incorporated this strategy because the each particle in the droplet holds a certain amount of fluid. A droplet shows three kinds of nature when it falls to the canvas that depends impact of the dropping droplet on the canvas.

1. If impact > α, the droplet splits into small droplets of varying in number of particles determined by the random function RF. Depending on the $u_{canvas}(i)$ velocity of the initial droplet, child droplet spread on the canvas at the distance derived by $u_{canvas}(i)$, and number of particles in the droplet.
2. If α > impact > β, the fluid volume represented by the droplet is squashed and flattened on the canvas.
3. If the droplet simply does not impact the canvas, droplet leaves a smaller impression since surface tension pulls the volume into a spherical shape.

Following this strategy, I enable my model to create a realistic spray painting that is very close to real spray painting of the artist. The user is given permission to control the parameter $\mathbf{u_{canvas}(i)}$ **,** $\mathbf{u_z(i)}$ that affect the size of the splatter using the parameter $\mathbf{S_{velocity}}$**,** $\mathbf{S_{pressure}}$.

A comparison between the real spray painting and patter generated by our strategy is given in fig 35 and fig 36.

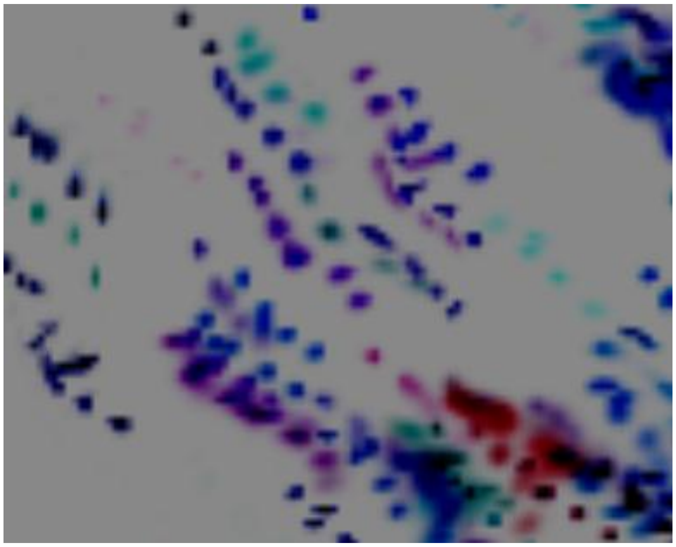


fig.35 Image formed by the my approach

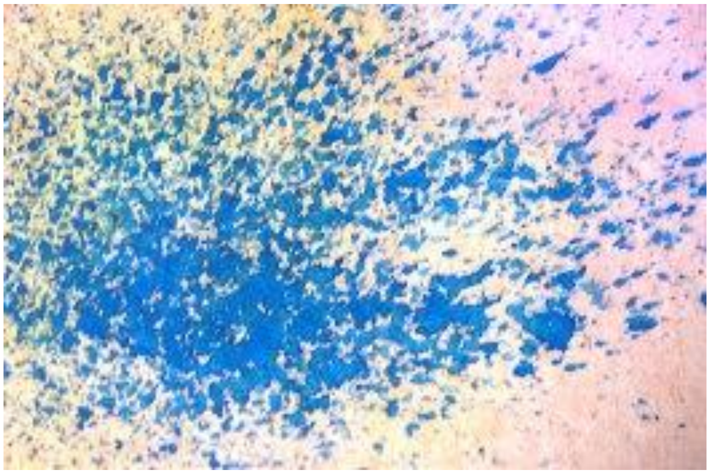


fig.36 Real image of spray painting

3.6.4 SPRAY DROPLET- CANVAS INTERACTION STAGE

When the color droplet falls to the canvas, I use the LBM fluid model to simulate this interaction. A more layered simulation is required to create, the more complex pattern caused by the color pigment flow is impeded by canvas, rough surface or accumulation of color constituents.

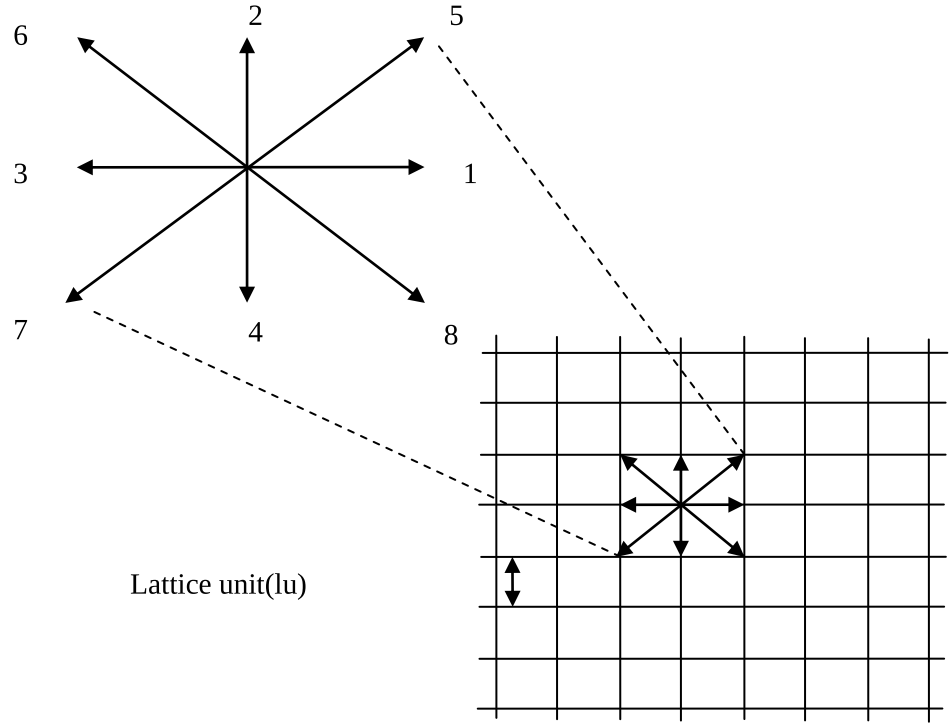


fig.37 Discrete lattices velocity of D2Q9 model.

In LBM models, fluid is simulated by the pseudo particles. These particles go through two process. First one is the propagation (streaming), consisting of movement of particle form one cell(lattice mash) to another. Second is a collision process when particle head on with each other. Since simulation is on the canvas, I have used D2Q9 model which 2D uniform grid. 9 in D2Q9 model denotes the nine distribution f_i, that suggest expected number of particles travelling along the lattice vector e_i.

LBM is sub-categories of Cellular Automata. General characteristics model include:

- Set of connected sites (the lattice)
- Some state-variables defined at each site (several real variables for LBM)
- An update rule, based on local and neighbor information

To summarize, the solution algorithm is simplified as follows:

The LBM solution algorithm is

1. for each time step

2. for each lattice (mesh) point

3. Stream()

4. Collide()

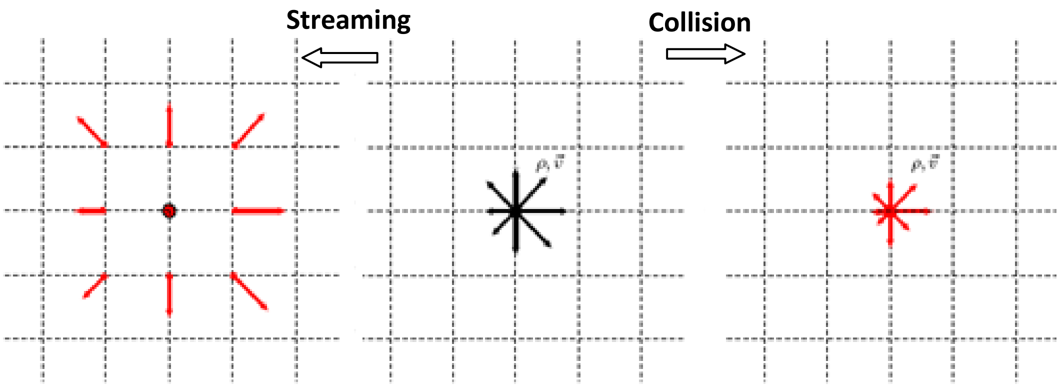


fig.38 Illustration of the streaming process and collision process on a D2Q9 lattice.

Streaming: f_i is travelling to the neighboring cells along the vector e_i .

Collision: f_i : f_i that reaches at the same cell are collided and they are forced to disperse to finally attain their equilibrium function $f_i^{(eq)}$

(3.19) $$\boldsymbol{\rho} = \sum_{i=0}^{\beta-1} \boldsymbol{f_i}$$ (Macroscopic fluid density)

(3.20) $$\vec{\boldsymbol{u}} = \frac{1}{\rho} \sum_{i=0}^{\beta-1} \boldsymbol{f_i}\, \vec{\boldsymbol{e_i}}$$ (Macroscopic fluid velocity)

The DFs at each lattice point is updated by using the equation:

$$\boldsymbol{f_i}(\vec{\boldsymbol{x}} + \vec{\boldsymbol{e_i}}\Delta \boldsymbol{t}, \boldsymbol{t} + \Delta \boldsymbol{t}) = \boldsymbol{f_i}(\vec{\boldsymbol{x}}, \boldsymbol{t}) - \frac{[\boldsymbol{f_i}(\vec{\boldsymbol{x}},\boldsymbol{t}) + \boldsymbol{f_i^{eq}}(\vec{\boldsymbol{x}},\boldsymbol{t})]}{\boldsymbol{\tau}} \qquad (3.20)$$

When the droplet falls to the canvas, it creates a spray droplet footprint, according to the *Point droplet* process described in the above section. The resultant droplet is then down sampled and converted into a composition of the water and pigment in the surface layer. At this time LBL simulation process comes into action. Fluid flows or evaporates within this layer under the LBM process. Pigment redistributed to adjacent cells based on $\mathbf{f_i}$

$$Pr^*(x) = \frac{1}{\rho}\sum_{i=1}^{8} f_i \mathrm{Pr}(x - e_i) \qquad (3.21)$$

Where p_r and p_r^* is the amount of pigment in the flow layer in the current and the next timestep respectively. For each iteration, a certain amount of pigment of the flow layer is transferred to the fixture layer, where they no longer move along with the fluid.

LBM has many advantages; one of them that is particularly relevant for my current purpose is the simplicity of coding. Application inspired by LBM only has several hundreds of lines, yet it can be easily be worked out to simulate some non-trivial systems such as the Rayleigh-Benard convection or the von Karman vortex street r.

LBM has several disadvantages, the most serious of them being the relative "stiffness" of the approach relative to the equations it eventually integrates (which refer to the problem that some additional attempts is required to solve anything different from the Navier-Stokes equations; however, once this effort is done, the algorithm simplicity appears once again)

3.6.4.1 INK DIFFUSION RENDERING

The above described fiber mess structure of the canvas hold the information about how each point on the canvas is connected to its neighboring points. The schema to simulate the point to point flow of ink through the fiber mesh is controlled by the LBM.

Bounding speed up time is my main motive At any particular moment, points at the periphery of the fallen droplet called "circulation border", being the initial boundary point for the diffusion. The diffusion process initiated by the prorogation of particles from initial *circulation border* to successive connected points [27]. Diffusion process is a function of time represented by step counter represented by counter from zero to n.

Process follows following steps

1. Color Pigment in point C at the current circulation front can flow to point C' only if C' is connected to C and is dried.
2. A portion of the fluid associated with the color pigment is absorbed by the point C' before it is flowing away to some other points.
3. The ink absorbed at C' evaporates after unit time ***t***. ***t*** is assumed according to stylization.

In my schema, the color pigment density at *circulation border* is evaluated on the basis of sized of the droplet. The points covered by the circulation border will not be included in the next evaluation of the circulation boundary for a short period of time. Hence computation become easier because of a linear increase in the number of points over time. Diffusion process does not propagate particles in reverse direction. Addition to these properties, overlapping of the droplet is allowed on prior droplet imprint as it was evaporated after a set time elapses.

3.6.4.1 PIGMENT INTENSITY CONTROLLING FRAMEWORK

Refereeing work on real watercolor painting, it is found density of the pigment in ink reduces in the progression of the diffusion process due to the deposition of colloidal color pigments onto the fibers. However, ink diffusion is a micro level physical phenomenon of hydrodynamics, it is difficult to formulate the ink dynamics.

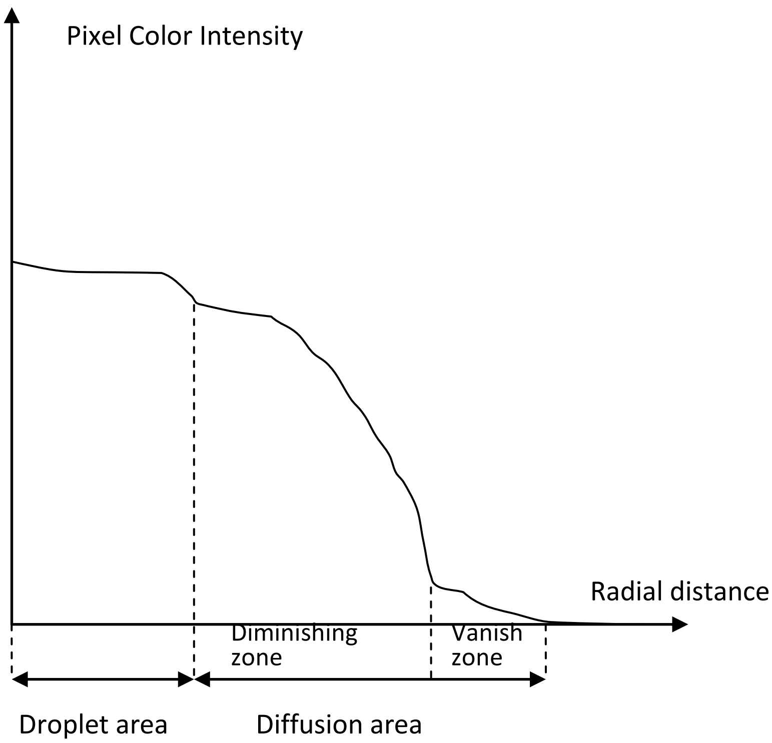


fig.39 Droplet area representation

To actually observe changes in ink density, I have performed an experiment on real circular diffusion images generated in two different types of papers. Briefly, after an image is divided like a pie into four equal parts, all the cells of one part are plotted, given the color density as a function of radial distance from the center of the image. Then, based on the geometric feature of the skeleton line, the diffusion area is divided into two zones [27]: (1) the *diminishing zone* where the ink density gradually decreases, and (2) the ensuing *vanish zone* where the ink density is weak and fluctuates. The whole rendered area can now be divided into the stroke area, where ink was directly applied on the paper and the diffusion area that appears as the result of diffusion. The diffusion area is divided again into the *diminishing zone* and *vanish zone* (Fig. 36).

Chapter 4

IMPLEMENTATION ISSUES AND RESULTS

- **Texture Updates**
- **Hardware Precision**
- **GPU Processing**

4. IMPLEMENTATION ISSUES AND RESULTS

All results in my research were generated using Sony Xperia Z 1.5 GHz Qualcomm APQ8064+MDM9215M Quad Core and with a desktop computer with an Intel Core i5 CPU, an NVIDIA GeForce GTX 680 GPU. For further details refer appendix A. I have also used threading to implement the LBM fluid simulator, and OpenGL and for the final rendering. In examples figure, I used a canvas with a resolution of 40962, and the LBM spray droplet and Canvas simulation were performed on a resolution of 5122. Each brush stroke is simulated with 200 to 400 particles. I have an overall system frame rate of 40 to 70 frames per second, depending on the number of particles currently being simulated in canvas.

I have simulated my approach in Processing language. Processing is a programming language, development environment. It was initially developed as the software sketchbook to embody computer graphic essentials to within a visual context. For further details refer appendix A. Specification of the Processing is following

- Interactive programs with 2D , 3D or PDF output
- Integrated to OpenGL and OpenGL ES for accelerated 3D.
- For GNU/Linux, Mac OS X, and Windows and Android
- Over 100 libraries extend the core software

	Texture	Contents
1.	**VelDen**	[**u, v, wf, leak**]
2.	**Misc**	[**blk, f0, lwf, ws**]
3.	**Texture1**	f[**N, E, W, S**]
4.	**Texture2**	f[**NE, SE, NW, SW**]

Table 1: Texture packaging data

Symbol	**Porpose**
u, v	Fluid velocity
Wf	fluid density in flow layer
Lwf	Fluid density inflow layer in the last iteration
Ws	Water amount on surface layer
Leak	Amount of water leaking from surface layer to flow layer
Blk(Visc)	Blocking factor
f0	Distribution function for stationary particles
f[N, E, W, S]	Dist. Functions towards nearest neighbors
f[NE, SE, NW, SW]	Dist. Functions towards next nearest neighbors

Table 2: Texture packaging data

4.1 TEXTURE UPDATES

Each of the above simulation textures is updated by rendering the paper geometry to a pixel buffer using a fragment program that performs the needed LBM operations. The content of the pixel buffer is then copied to the destination texture. The six texture updates for the LBM simulation are listed in Table

Texture update	Output texture	Operation
1.	Misc	Derive **f0**, **blk** Deposit or update **ws** Save **wf** to **lwf**
2.	Texture1	Collide f[N, E, W, S]
3.	Texture2	Collide f[**NE, SE, NW, SW**]
4.	Texture1	Stream f[**N, E, W, S**]
5.	Texture2	Stream f[**NE, SE, NW, SW**]
6.	VelDen	Derive **u, v, wf, leak** Receive water from surface

Table 3: Texture Update Operations

For color pigment deposition, I apply fragment programs that perform the necessary operations on the geometry formed by the movement of the *Sprayer Handle* over the canvas. Spray pattern stretched as shown figure to give continuity in the spray process. Both the penetrating parts of the stretched tuft and the proper geometry are rendered to the pixel buffer simultaneously, so that deposition does not need extra rendering passes.

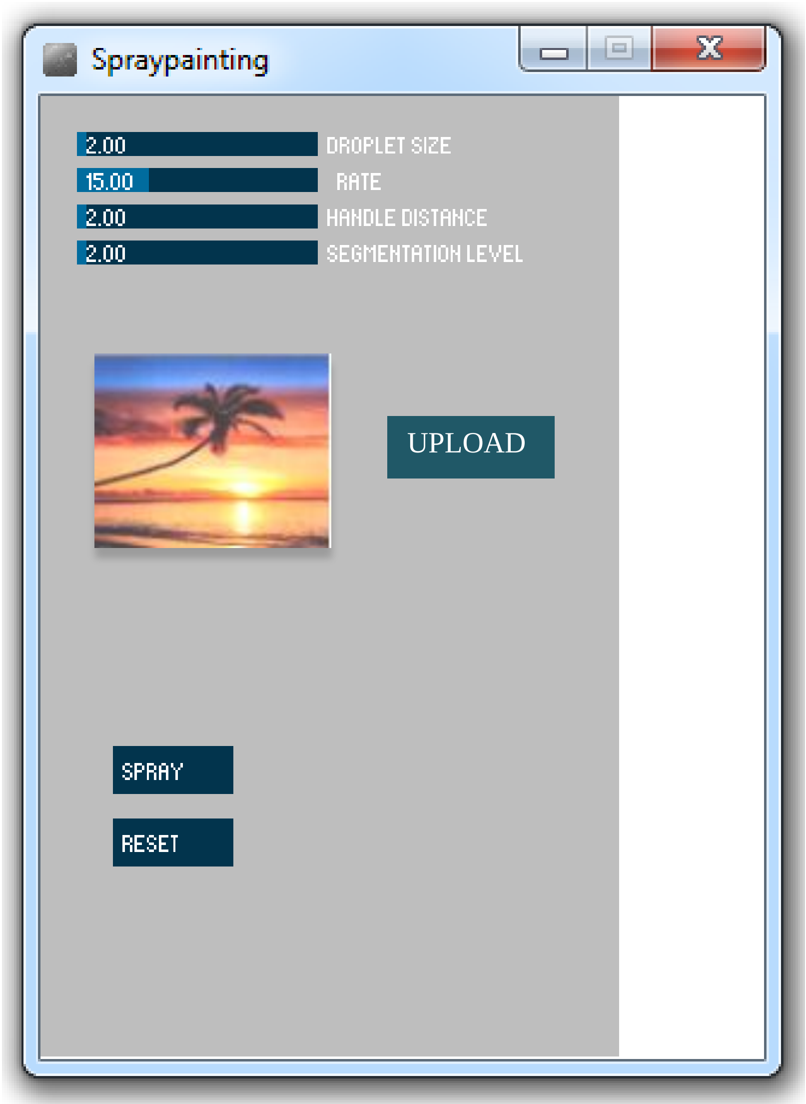


fig.40 Interface of Spray application

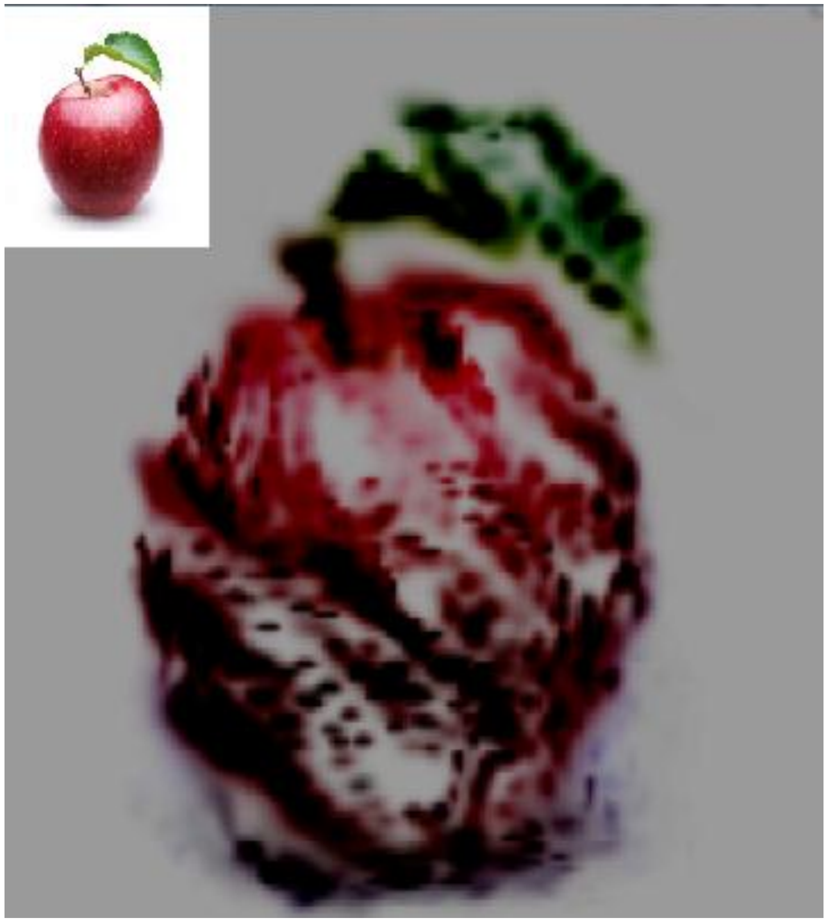


fig.41 Simulated Painting 1

Parameter to draw above Spray Painting

Fluid_Visc = 0.0011f;

Fluid _Diffusionspeed = .0004;

Fluid _Solver_Iterations = 10;

Pigment_Mult = 10;

Pigment_velocity_Mult = 45.0f;

Droplet size = 5;

Segmentation level = 165;

fig.42 Simulated painting 2

Parameter to draw above Spray Painting

Fluid_Visc = 0.0021f;

Fluid _Diffusion_speed = .0001;

Fluid _Solver_Iterations = 10;

Pigment_Mult = 5;

Pigment_velocity_Mult = 25.0f;

Droplet size = 4;

Segmentation level = 240;

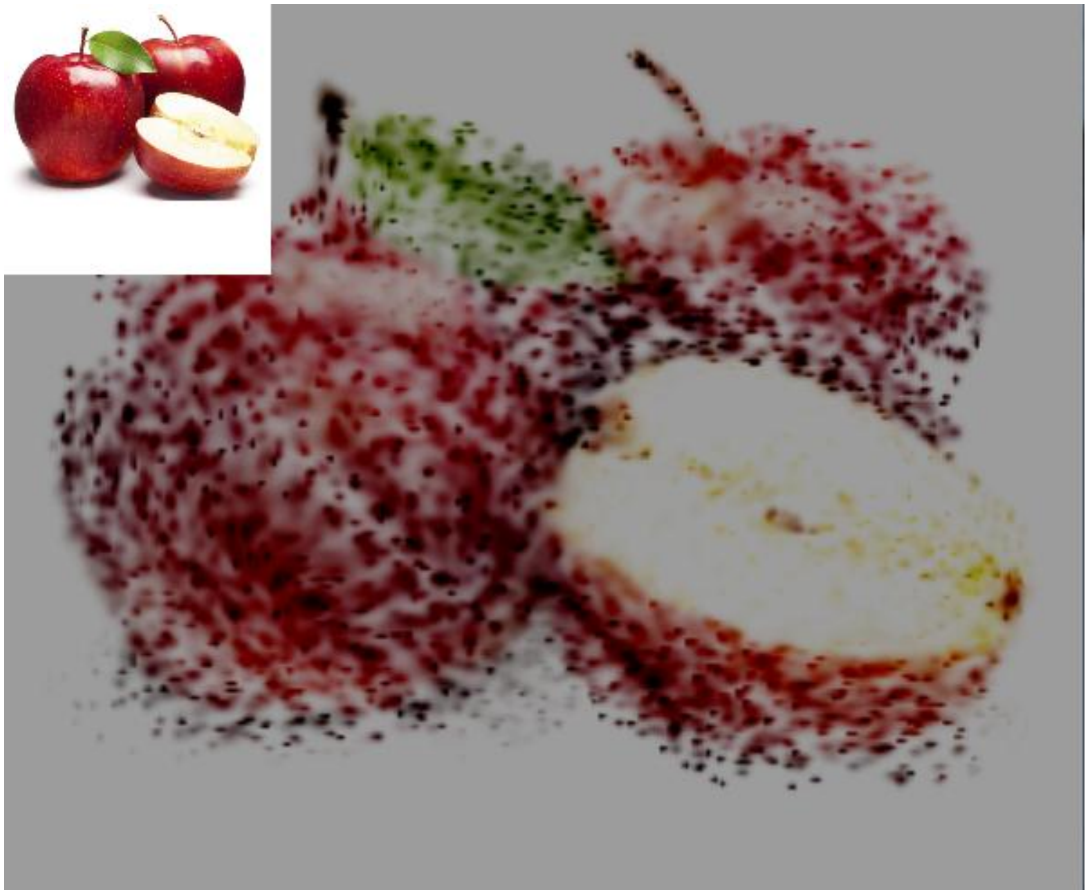


fig.43 Simulated painting 3

Parameter to draw above Spray Painting

Fluid_Visc = 0.0051f;

Fluid _Diffusion_speed = .0001;

Fluid _Solver_Iterations = 15;

Pigment_Mult = 5;

Pigment_velocity_Mult = 25.0f;

Droplet size = 2;

Segmentation level = 250;

4.2 HARDWARE PRECISION

I use the 16-bit float data type for my simulation spray rendering (Texture formation), but perform the computation in the fragment programs with 32-bit float precision. There are two reasons for using 16-bit float texture instead of 32-bit: (1) it is currently the only float type with hardware bilinear texture interpolation (OPENGL), and (2) it lowers the memory consumption and the overheads in copying data.

A few precision issues came out when I used the 16-bit float textures. A problem was encountered when pigments were transported from the flow layer to the fixture layer (Section 3.6.4). I found that pigments were vanishing at some stage in the relocation, with the target surface not getting the correct amount. The hardware round-off procedure for the 16-bit float texture to write was the one reason. I solved the issue by manually quantizing the transferred pigment amount with a quantization step of 1/2048.

Another issue is in the hardware interpolation of 16-bit float textures as needed in the generation of high quality output (Section 4.1). Some artifacts appear in the generated higher resolution output I′ : the pixels in I′ that are supposed to be interpolated from the pixels in I were lighter than what they should be. We suspect this is caused by the hardware interpolation giving values a bit smaller than they should be. These small errors would not be noticeable if the interpolation is done only once, but become visible when accumulated. We solve this problem by reducing the number of updates to I′. In our implementation, we only update I′ every 200-400 time steps.

4.3 GPU PROCESSING:

Currently, I limit our dispersion simulation to three pigments (colors) at a time so that we can fit pigments and glue concentrations into one RGBA texture. A large gamut of colors, however, is still available by mixing the three pigments (e.g. Red, green, yellow). During each time step, I perform six texture updates for the LBE flow simulation, and another six texture update for moving the pigments and glue. The boundary trimming requires one more texture update for I′. Thanks to the simplicity of the method, the six fragment programs for the LBM operations have an average assembly instruction count of only.

5 CONCLUSIONS AND FUTURE PLAN

I present an interactive system for spray art. The ink-dripping interaction mode imitates the real-world artistic skills; artists can extend their real-world experience and adapt to my model easily. The core 2-stage spray framework simulates the interaction of ink-spray handle outlet holes, and spray droplet-Canvas. I show that my system could allow users to generate more realistic organic patterns by comparing with previous works and real-world splatters and sprayer techniques. In the future, I will explore the possibility of using the 1D fluid simulation of [9] to more accurately model paint movement within the brush hair, in order to create a better initial particle distribution when paint dislodges from the brush.

APPENDIX A

A.1 SONY XPERIA Z

The Sony Xperia Z is a rectangular slab that marks a departure from Sony's 2012 design. The design is "Omni-Balance", according to Sony, which is focused on creating balance and symmetry in all directions.

HARDWARE

The phone's dimensions are stated to be 139 x 71 x 7.9 mm and its weight is 146 grams (including battery). It comes with a Lithium-ion battery, rated at 2330 **mAh**.

The Xperia Z features a 13.1-megapixel rear camera with **an Exmor RS sensor**, flash, image stabilization, HDR, face detection (with red-eye reduction), Sweep Panorama, scene recognition and 16x digital zoom, and a 2.2-megapixel front camera with an Exmor R sensor. Both cameras can record video at 1080p.

The Smartphone has a capacitive touchscreen with a 5" size, and its resolution is 1920x1080 pixels (1080p) at 443 ppi, with 16 million colors. The screen uses Sony's Mobile Bravia Engine 2 to enhance the picture.

In addition to this, the Xperia Z has a Qualcomm Snapdragon S4 Pro chipset, 2 GB RAM memory, 16 GB internal storage flash memory (with approx. 11 GB available to the user) and a MicroSD card slot which accepts cards up to 64 GB.

Specification of handset (Xperia Z)

- 5(12.7 cm), 1920x1080 pixels, 16,777,216 color TFT
- Sony Mobile BRAVIA Engine 2
- Dynamic normalize
- Google Android 4.3(Jelly Bean)
- 1.5 GHz Qualcomm APQ8064+

A.2 PROCESSING

Processing relates software concepts to the principles of visual form, motion, and interaction. It integrates a programming language, development environment, and teaching methodology into a unified system. Processing was created to teach fundamentals of computer programming within a visual context, to serve as a software sketchbook, and to be used as a production tool for specific contexts. Students, artists, design professionals, and researchers use it for learning, prototyping, and production. The Processing language is a text programming language specifically designed to generate and modify images. Processing strives to achieve a balance between clarity and advanced features. Beginners can write their own programs after only a few minutes of instruction, but more advanced users can employ and write libraries with additional functions. Many computer graphics and interaction techniques can be discussed, including vector/raster drawing, image processing, color models, mouse and keyboard events, network communication, and object-oriented programming. Libraries easily extend Processing's ability to generate sound, send/receive data in diverse formats, and to import/export 2D and 3D file formats.

8. REFERENCES

1. Kyprianidis, J., John Collomosse, Tinghuai Wang, and Tobias Isenberg. "State of the'Art': A Taxonomy of Artistic Stylization Techniques for Images and Video." (2012): 1-1.
2. Hertzmann, Aaron. "Painterly rendering with curved brush strokes of multiple sizes." *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998.
3. Gooch, Bruce, Greg Coombe, and Peter Shirley. "Artistic vision: painterly rendering using computer vision techniques." *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*. ACM, 2002.
4. Huang, Hua, Yu Zang, and Chen-Feng Li. "Example-based painting guided by color features." *The Visual Computer* 26, no. 6-8 (2010): 933-942.
5. Kindlmann, Gordon, Ross Whitaker, Tolga Tasdizen, and Torsten Moller. "Curvature-based transfer functions for direct volume rendering: Methods and applications." In *Visualization, 2003. VIS 2003. IEEE*, pp. 513-520. IEEE, 2003.
6. Huang, Shen-Wen, De-Lor Way, and Zen-Chung Shih. "Physical-based model of ink diffusion in Chinese paintings." (2003).
7. Huang, Zhiyong. "A GPU-based method for real-time simulation of Eastern painting." In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pp. 111-118. ACM, 2007.
8. You, Mi, Taekwon Jang, Seunghoon Cha, Jihwan Kim, and Junyong Noh. "Realistic paint simulation based on fluidity, diffusion, and absorption."*Computer Animation and Virtual Worlds* 24, no. 3-4 (2013): 297-306.
9. Lee, Sangwon, Sven C. Olsen, and Bruce Gooch. "Interactive 3D fluid jet painting." In *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pp. 97-104. ACM, 2006.
10. Wang, Chung-Ming, and Ren-Jie Wang. "Image-based color ink diffusion rendering." *Visualization and Computer Graphics, IEEE Transactions on* 13.2 (2007): 235-246.

11. Dev, Kapil. "Mobile Expressive Renderings: The State of the Art." *IEEE Computer Graphics and Applications* 33, no. 3 (2013): 22-31.
12. Rakkolainen, Ismo, Stephen DiVerdi, Alex Olwal, Nicola Candussi, Tobias Hüllerer, Markku Laitinen, Mika Piirto, and Karri Palovuori. "The interactive fogscreen." In *ACM SIGGRAPH 2005 Emerging technologies*, p. 8. ACM, 2005.
13. Bao, Yuanxun Bill, and Justin Meskas. "Lattice Boltzmann Method for Fluid Simulations." (2011).
14. Lee, Jintae. "Diffusion rendering of black ink paintings using new paper and ink models." *Computers & Graphics* 25, no. 2 (2001): 295-308.
15. Chen, Chang Wen, Jiebo Luo, and Kevin J. Parker. "Image segmentation via adaptive K-mean clustering and knowledge-based morphological operations with biomedical applications." *Image Processing, IEEE Transactions on* 7, no. 12 (1998): 1673-1683.
16. Ye, Zhou, Shun-Nan Liou, and Ying-Ko Lu. "Method of controlling mobile device with touch-sensitive display and motion sensor, and mobile device." U.S. Patent Application 13/492,918, filed June 10, 2012.
17. Zhao, Mingtian, and Song-Chun Zhu. "Customizing painterly rendering styles using stroke processes." *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2011.
18. Strassmann, Steve. "Hairy brushes." *ACM SIGGRAPH Computer Graphics*. Vol. 20. No. 4. ACM, 1986.
19. Haeberli, Paul. "Paint by numbers: Abstract image representations." *ACM SIGGRAPH Computer Graphics*. Vol. 24. No. 4. ACM, 1990.
20. Hertzmann, Aaron. "Non-photorealistic rendering and the science of art."*Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2010.
21. Mukundan, R., and C. Han. "A Fast Algorithm for Painterly Rendering on Mobile Devices." (2008).
22. Plesa, Malisa Ana, and William Cartwright. "Evaluating the effectiveness of non-realistic 3D maps for navigation with mobile devices." *Map-based mobile services*. Springer Berlin Heidelberg, 2008. 80-104.
23. Hines, R. L. "Electrostatic atomization and spray painting." *Journal of Applied Physics* 37, no. 7 (2004): 2730-2736.

24. Mi, Xiao-Feng, Min Tang, and Jin-Xiang Dong. "Droplet: a virtual brush model to simulate Chinese calligraphy and painting." *Journal of Computer Science and Technology* 19, no. 3 (2004): 393-404.
25. Xu, Shibiao, Xing Mei, Weiming Dong, Zhiyi Zhang, and Xiaopeng Zhang. "Interactive visual simulation of dynamic ink diffusion effects." In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pp. 109-116. ACM, 2011.
26. Hund, Jerry P. "Spray application processes." *Metal finishing* 98, no. 6 (2000): 181-197.
27. Geng, Weidong. "Simulating Artistic Media for Digitized Creation of Artworks." In *The Algorithms and Principles of Non-photorealistic Graphics*, pp. 11-89. Springer Berlin Heidelberg, 2010.
28. Ekman, Axel, Arttu Miettinen, Tuomas Turpeinen, Kaj Backfolk, and Jussi Timonen. "The number of contacts in random fiber networks." *Nordic Pulp and Paper Research Journal* 27, no. 2 (2012): 270.