

Modelo de Objetos Para Um Sistema de Pintura Digital

MARIA CÉLIA PAIVA DE FREITAS¹

¹Laboratório de Computação Gráfica - COPPE / UFRJ
Caixa Postal 68511
21945-970 Rio de Janeiro, RJ, Brasil
celia@lcg.ufrj.br

Abstract. This paper describes an electronic paint system based on the "canvas-ink-brushes" paradigm. The system model is object-oriented and simulates some kinds of strokes using a hierarchical representation among the paint objects. The pigments used by the brushes have all the material properties normally associated with standard illumination models. The small-scale geometry of the canvas can be interpreted as a texture which could be revealed by bump-mapping. The brushes can also be used to alter the height values of the canvas. This textured paint adds an extra physical dimension to the image and expands its expressive possibilities.

1 Introdução

Ao contrário de áreas da computação gráfica que estudam a síntese de cenas com um realismo fotográfico, o sistema de pintura digital descrito neste artigo tem como objetivo a geração de imagens com abstrações artísticas e a busca de novas formas gráficas expressivas, únicas ao computador. Imagens obtidas por meio da pintura, da gravura ou da fotografia, por exemplo, possuem uma inacreditável variedade e complexidade de formas, cores, texturas e contrastes, ao contrário de muitas imagens produzidas eletronicamente. Os "pincéis" normalmente usados nos sistemas de pintura convencionais constroem imagens apenas pelo posicionamento sucessivo de cópias de algum padrão estático. Em muitos casos, a aparência fria e estéril das imagens computadorizadas deve-se à ausência de texturas interessantes e dos contrastes propiciados pela sobreposição e combinação das mesmas.

Este trabalho descreve um sistema de pintura digital que busca criar de maneira simples e estruturada imagens com um nível maior de interesse e complexidade. O sistema é baseado em um modelo orientado a objetos, onde as principais unidades de representação são: *tela*, *tinta*, *paleta*, *pincel* e *pincelada*. Os objetos podem ser divididos em classes e subclasses e uma hierarquia de representação entre estas classes de objetos (agrupamentos) foi usada para modelar as pinceladas.

Uma modelagem da textura da pintura foi incorporada ao sistema, mas sem compromissos com a física e com as leis da dinâmica. Neste caso interessa que os resultados estéticos sejam satisfatórios e que as respostas estejam consistentes com a expectativa do

artista após a compreensão do modelo proposto. As tintas, pincéis e pinceladas possuem propriedades materiais e texturas diversas (brilho, cor, volume, etc.), sendo possível pintar com tintas de diferentes materiais. A micro-geometria da tela é alterada através da inserção das partículas de pintura ou através de pincéis especiais. Os atributos da pintura gerada podem ser utilizados como entrada para modelos de iluminação comumente usados.

Um dos objetivos alcançados pelo presente trabalho foi a unificação de propostas de trabalhos anteriores em um modelo orientado a objetos ([Bleser et al (1988)], [Cockshott et al (1992)], [Hanrahan et al (1990)], [Haerberli (1990)], [Smith (1982)], [Strassman (1986)]). Foram incorporadas em um único contexto técnicas novas de pintura digital, além das já conhecidas, mas com a vantagem de se facilitar a manipulação de um grande número de recursos pelo conhecimento da estrutura lógica do modelo. A modularização do sistema permite também experimentações em diversos níveis durante o processo de criação, através da seleção e combinação entre os diferentes tipos de objetos. Além disso, foi modelado um novo tipo de pincelada (*pincelada Minkowski*, seção 3.5) capaz de produzir efeitos complexos e interessantes de uma maneira bastante simples e rápida.

2 Visão Geral do Modelo de Objetos

A pintura será feita sobre uma *tela* plana composta de *células*. As células da tela armazenarão pigmentos materiais, as *partículas de pintura*. Cada partícula possui não apenas cor mas também propriedades materiais (difusão e especularidade, brilho, transparência, volume, etc). A tela possui uma

micro-geometria, ou "relevo", definida através das alturas das células, que são usadas para calcular as suas normais. O relevo da tela e as propriedades materiais dos pigmentos são revelados pela iluminação da tela por uma ou mais fontes de luz.

Um *pincel* é um conjunto de pontos do plano, sendo que cada ponto deste conjunto está associado a um ou mais parâmetros, como cor, volume, etc. (*ponto material*). Estes parâmetros são determinados pelo mapeamento de texturas sobre a *forma* geométrica do pincel, o que é feito pela *paleta*. O pincel usa o ponto material, após processamento pela *pincelada*, para alterar o estado das células. Cada subclasse especializada do pincel altera as células de maneira diferente, seja gerando novas partículas de pintura, ou alterando propriedades das partículas já existentes, ou alterando a micro-geometria das células.

A *pincelada* será a responsável pelo processamento do conjunto de pontos materiais dos pincéis. Associada à *pincelada* está uma sequência de pincéis, representando o movimento de um único pincel que vai se alterando durante a sua trajetória. Os atributos dos pontos destes pincéis podem ser interpolados, combinados ou simplesmente copiados pela *pincelada*, gerando um novo conjunto de pontos materiais. Estes pontos são retornados aos pincéis componentes da *pincelada* e usados pelos mesmos para alterar as células. Cada subclasse especializada da *pincelada* possui diferentes métodos para o processamento dos pontos materiais dos seus pincéis.

Uma pintura pode ser pensada como uma coleção ordenada de *pinceladas*.

3 As Classes de Objetos

O esquema conceitual do modelo está baseado nos objetos de pintura. Estes objetos podem ser agrupados, formando uma hierarquia (figura 1) e, além disso, as classes de objetos dividem-se em subclasses especializadas (figura 2). Uma interface comum é usada para diferentes subclasses derivadas de uma mesma classe (polimorfismo) e os detalhes de implementação são determinados por cada classe específica [Kim et al (1989)].

Os objetos do modelo possuem os seus próprios métodos e atributos (*ação de pincelar, regras de espalhamento da tinta, atributos materiais da tinta, regras de combinação entre materiais, etc*), possuem diferentes graus de especialização (*diferentes tipos de pincéis e pinceladas*) e "trocam" mensagens entre si (*pincel insere tinta na tela*) (figura 3). A idéia é que conforme sejam melhoradas ou criadas novas regras, estas possam ser incorporadas com facilidade ao modelo proposto sem alterações drásticas na arquitetura

do sistema. Técnicas bastante diversas podem ser inseridas neste contexto comum, facilitando a sua compreensão e também a implementação e manutenção do sistema.

O processo de pintura é baseado em um conjunto de efeitos organizados modular e hierarquicamente. Aspectos da simulação podem ser substituídos ou aumentados através de algoritmos mais sofisticados. O usuário familiariza-se com um pequeno repertório de ferramentas simples, manipuláveis de maneira estruturada.

Neste tipo de ambiente uma mesma *pincelada*, por exemplo, pode ser desenhada em vários níveis de complexidade, desde desenhos rápidos até a visualização final, já iluminada.

Partindo destas idéias básicas foi construída uma representação cujas unidades básicas são as classes de objetos (figuras 1, 2 e 3). Os principais métodos e atributos de cada uma destas classes são descritos a seguir.

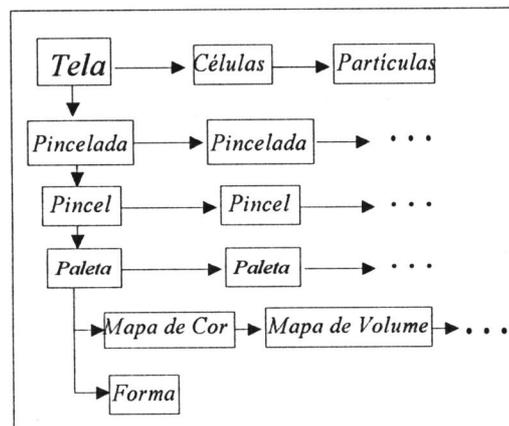


figura 1: Hierarquia dos objetos

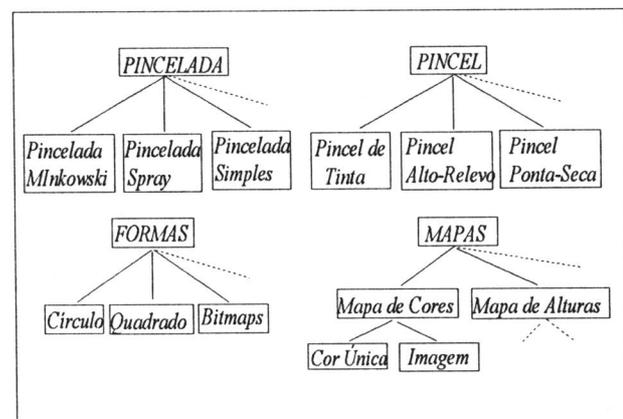


figura 2: Algumas Classes e Subclasses do Modelo

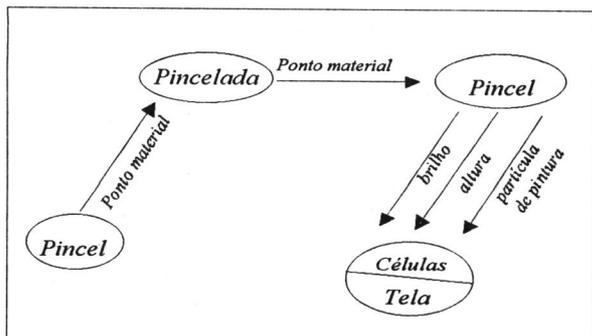


figura 3: Troca de mensagens entre objetos do modelo

3.1 Tela e Células

A tela τ é uma coleção de células c_{ij} dispostas sobre uma malha regular, interconectadas e associadas a um conjunto de regras de comportamento da pintura, podendo ser pensada como um autômato celular [Cockshott et al (1992)]:

$$\tau = \cup c_{ij}, 0 < i < m, 0 < j < n,$$

$m = \text{largura da tela}; n = \text{altura da tela}$

As células são reservatórios de partículas de pintura com determinada capacidade de armazenamento. Cada célula c_{ij} possui uma altura h_{ij} que, em conjunto, definem a textura da tela ou da pintura. Os volumes das partículas de pintura inseridas nas células vão alterando a sua altura. Cada célula possui uma coordenada (x,y) definida por sua indexação na tela:

Coordenadas da célula $c_{ij} = (i,j)$

A classe célula possui os seguintes atributos:

- Partícula de Pintura
- Altura h
- Norma
- Volume armazenado

As células possuem métodos para exibir o seu conteúdo e para receber os atributos que alteram o seu estado.

3.1.1 Cálculo da Normal das Células

As alturas das células formam um mapa de alturas que pode ser interpretado como uma textura a ser revelada por "bump-mapping". Esta textura é criada pela deposição de partículas de pintura na célula ou através da utilização, pelo pincel, de mapas de alturas pré-definidos (como será visto na seção 3.3).

A chave para o "bump-mapping" é o cálculo da perturbação da normal. Como a superfície da tela é um plano, a normal N possui sempre a direção $N = [0, 0, 1]$. O cálculo da normal perturbada N' em c_{ij} é feito através da fórmula de Blinn [Blinn (1978)], considerando a pirâmide ao redor de cada vértice, onde

h_{ij} = altura de c_{ij} (figura 4). A normal perturbada é uma média das normais das faces do tetraedro considerado.

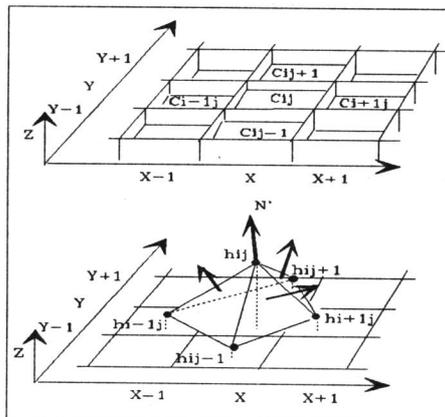


figura 4: Micro-geometria das células

Pode-se mostrar que [Cockshott et al (1992)]:

$$N' \propto [h_{i-1,j} - h_{i+1,j}, h_{i,j-1} - h_{i,j+1}, 4]$$

3.1.2 Iluminação da Tela

O modelo de iluminação descrito determina a luz refletida para o observador em cada célula da tela. A luz incidente de uma ou mais fontes, a orientação da tela e as propriedades materiais das partículas de pintura são usados pelo modelo para determinar a intensidade de luz refletida, de acordo com a seguinte equação de iluminação [Blinn (1977)]:

$$I = aI_{amb} + dI_{diff} + sI_{spec},$$

$$I_{diff} = \sum_j I_{L_j} (\max(0, \langle N, L_j \rangle)),$$

$$I_{spec} = \sum_j I_{L_j} \langle N, H_j \rangle^c$$

onde:

- I = Intensidade percebida
- I_{amb} = luminosidade do ambiente
- I_{diff} = contribuição difusa
- I_{spec} = contribuição especular
- I_{L_j} = intensidade da j -ésima fonte de luz
- N = vetor normal à célula
- L_j = vetor da direção da j -ésima fonte de luz
- V = vetor da direção de observação $[x,y,-d]$, onde x,y são as coordenadas da célula e d é a distância do observador à tela.
- H_j = direção ideal de reflexão, onde $H_j = (L_j + V) / \|L_j + V\|$
- a = proporção de reflexão ambiente
- d = proporção da reflexão difusa
- s = proporção da reflexão especular
- c = constante de Phong (brilho do material)

A equação é usada para cada canal de cor. Outros modelos de iluminação poderiam ser usados, permitindo, por exemplo, a simulação de materiais com características não encontradas na natureza.

3.2 Partículas de Pintura

A tinta neste modelo é representada por um conjunto de *partículas de pintura*. Estas partículas são depositadas nas células pelos pincéis durante as pinceladas (ou apenas alteradas) e possuem as seguintes propriedades materiais:

- *Cor RGB (difusa e especular)*
- *Volume, transparência, coeficiente de difusão, coeficiente de especularidade, brilho, diluição, etc...*

As propriedades materiais influem nas regras de combinação e espalhamento e no processo de iluminação.

3.2.1 Combinação entre Partículas

Ao ser depositada em uma célula pelo pincel, a partícula de pintura é combinada com as partículas previamente armazenadas nas células. Alguns dos coeficientes da partícula que está sendo depositada são interpolados com os coeficientes da partícula já existente na célula.

Efeitos simples de transparência podem ser incorporados conforme descrito em [Rogers (1985)]. Quando a partícula inserida em uma célula é transparente, a sua cor pode ser combinada linearmente com a da partícula armazenada na célula:

$$C = tC_1 + (1-t)C_2$$

onde C_1 é a cor da partícula inserida e C_2 é a cor da partícula anterior. Se $t = 0$ a partícula é totalmente transparente e se $t = 1$ a partícula é opaca.

Esta aproximação linear não é um modelo adequado para superfícies rugosas, como no caso da tela. No caso em que as normais da célula são perpendiculares à direção de observação (células "laterais"), a espessura do material reduz a sua transparência. Para representar este efeito, pode-se usar no cálculo do fator de transparência uma aproximação não-linear simples baseada na componente z da normal à célula:

$$t = t_{min} + (t_{max} - t_{min})[1 - (1 - |n_z|)^P]$$

onde t_{min} e t_{max} são as transparências máxima e mínima para a partícula, n_z é a componente z da normal unitária à célula e P é um fator entre 0 e 1.

Uma outra alternativa é considerar o fator de transparência t como sendo inversamente proporcional ao volume da partícula.

Após ser inserida nas células, a tinta pode espalhar-se, combinar-se novamente com o conteúdo de outras células, depositar-se (alterando a altura da célula) e também secar, conforme o tempo vai passando. Este comportamento vai depender do tipo de material de que é feita a tinta, da geometria da tela e das regras de "espalhamento" das células. Assim, a tela pode ser pensada como sendo um autômato celular¹.

Os vários coeficientes das propriedades materiais das partículas são ajustados através de um editor de materiais, usando-se "sliders". Este tipo de editor normalmente ilumina uma esfera de acordo com os coeficientes selecionados e altera apenas uma tabela de cores indexada pelas normais, o que permite uma interação em tempo real. Para economizar memória no armazenamento de coeficientes materiais das partículas, pode-se criar uma tabela de materiais pré-definidos.

3.3 Paleta

A *paleta* é um conjunto de pontos, definidos por uma *forma*, cujos pontos possuem um ou mais parâmetros como coordenadas, cor, volume, altura (*ponto material*). Os atributos da paleta são:

- *uma forma geométrica simples*
- *um conjunto de mapas de texturas (mapas de cores, de alturas, de volumes, etc...) que são aplicados sobre esta forma.*

A paleta pode ser interpretada como sendo uma função que associa a uma forma um conjunto de parâmetros (o material):

$$\begin{aligned} \text{paleta: } \mathcal{R}^2 &\rightarrow \mathcal{R}^n \\ \text{paleta}(x,y) &= \\ \{ (x,y, \text{cor}(x,y), \text{altura}(x,y), \text{volume}(x,y), \dots) \mid \\ (x,y) \in \text{forma}, \langle \text{parâmetro} \rangle(x,y) \text{ é dado pelo mapa de} \\ &\text{texturas correspondente} \} \end{aligned}$$

$$\begin{aligned} \text{e} \\ \text{Paleta} &= \cup \text{paleta}(x,y), \text{ onde} \\ \text{paleta}(x,y) &\text{ é um ponto-material da forma} \\ &(x,y, \text{cor}, \text{altura}, \text{volume}, \dots) \end{aligned}$$

A partir da classe paleta são definidas subclasses e cada uma delas associa de maneira diferente os atributos dos mapas sobre os pontos da forma. Por exemplo, uma paleta "impressionista" mapeia sempre uma mesma cor do mapa de cores, qualquer que seja o ponto (x,y) da forma, reproduzindo efeitos propostos

¹Não implementado na presente versão

em [Haerberli (1990)]. As figuras 9 e 10 foram feitas usando-se paletas impressionistas e um mapa de cor comum a todas elas (foto pequena). As pinceladas usadas serão descritas depois.

A classe *mapa* é uma superclasse de todos os tipos de mapas de texturas (*mapas de cores, de alturas,...*). Cada subclasse possui o seu próprio método para atribuir a uma das coordenadas do ponto material o parâmetro correspondente. A subclasse mapa de cores, apenas como exemplo, também possui outras subclasses, correspondentes a diferentes maneiras de se gerar e armazenar mapas de cores (imagens, uma única cor, rampa de cores). Por sua versatilidade os mapas de textura são componentes chaves de um sistema de pintura.

A classe *forma* possui classes derivadas como *circulos, quadrados, poligonos e bitmaps*. Cada subclasse tem os seus próprios métodos de varredura, sorteio de pontos e cálculo da área, que são usados pelas pinceladas.

3.4 Pincel

A alteração direta do estado das células é feita através do *pincel*, seja através da introdução de partículas de pintura, pela mudança de seus atributos ou pela alteração da sua altura. O pincel fornece para a pincelada o seu conjunto de pontos materiais, que então retorna ao pincel um novo conjunto de pontos que serão usados para alterar as células. O estado do pincel em um determinado momento é dado por uma *lista de paletas* (figura 5):

$$Pincel = \{ \bigcup_i paleta_i \}$$

$$Pincel = \{ \text{ponto-material } p=(x,y,cor, \dots): p \in paleta_i, \text{ para algum } i \}$$

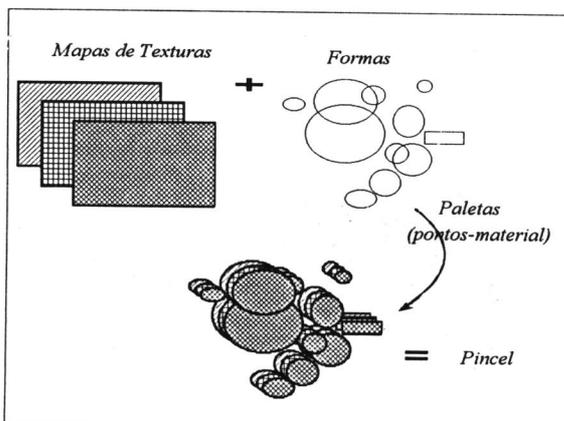


figura 5: Definição do material dos pontos do pincel

Cada subclasse do pincel usa um ou mais parâmetros do ponto material e seus métodos para alterar as células são implementados diferenciadamente. É descrito a seguir como algumas das subclasses do pincel usam um ponto material $p = (x,y,cor,brilho,volume,altura,\dots)$ para alterar a célula de coordenada $(p.x, p.y)$:

- *Pincel de tinta (carimbo)*
Cria partícula de pintura com atributos de p e insere a partícula na célula de coordenada $(p.x, p.y)$.
- *Pincel de tinta em alto-relevo (baixo-relevo)*
Cria partícula de pintura que só é aplicada se a altura da célula $(p.x, p.y)$ é maior (menor) que determinada constante. Pode ser usado para criar efeito de pintura com pastel sobre papel [Smith (1982)].
- *Pincel polidor*
Altera apenas brilho e coeficiente especular, mantendo a cor da partícula.
- *Pincel de laterais*
Cria partícula de pintura que só é inserida dependendo da inclinação da normal da células, permitindo pintar apenas laterais ou apenas horizontais.
- *Pincel alterador de relevo*
Seja h_p a altura do ponto p , a ser mapeada sobre a célula $(p.x, p.y)$ e h_c a altura desta célula. Então a nova altura da célula será:

$$h_c = \max(h_p, h_c) \text{ (pincel união)}$$

$$\text{ou}$$

$$h_c = \min(h_p, h_c) \text{ (pincel diferença)}$$
- *Pincel ponta seca*
Altura da célula é substituída pela altura de p .
- *Pincel raspador*
Altura da célula $(p.x, p.y)$ é diminuída por um fator de escala h_p .

É possível criar muitos outros tipos de pincéis, de acordo com as propriedades da célula, das partículas de pintura e dos pontos materiais. Pode-se definir um pincel-borracha, que remova a partícula da célula ou a substitua por uma partícula "de fundo" pré-definida, além de zerar a altura da célula. Pode-se também usar pincéis que alterem apenas a saturação da cor das partículas para simular um "branqueamento" das células. Enfim, um pincel pode "pintar" os atributos das células seletivamente, possibilitando a criação de uma ampla gama de efeitos.

3.5 Pincelada

A pincelada é definida por uma sequência de pincéis do mesmo tipo. A pincelada cria um conjunto de pontos materiais a partir dos pontos destes pincéis, seja por cópia, interpolação ou combinação, determinando também o conjunto de células que serão alteradas pelos pincéis. Um único ponto material de um pincel pode originar vários outros pontos durante a pincelada.

Cada tipo de pincelada possui os seus próprios métodos de edição, tratando também diferenciadamente os eventos associados ao mouse. A edição deve garantir a consistência e validade dos atributos. Cada pincelada possui tipos de respostas diferentes.

As subclasses processam os atributos dos pontos dos pincéis de maneira diferente, levando em conta apenas os atributos relevantes dos pincéis. Alguns tipos de pinceladas criadas usando este modelo hierárquico são descritas a seguir.

- *Pincelada Minkowski*

É uma classe na qual os atributos dos diferentes pincéis são interpolados para gerar um novo conjunto de pontos materiais. Esta pincelada é baseada na Soma de Minkowski [Kaul et al (1991)], como explicado adiante.

Sejam A e B conjuntos do \mathfrak{R}^n , conexos ou não. A Soma de Minkowski \oplus entre A e B é definida como sendo:

$$A \oplus B = \{a+b: a \in A, b \in B\}$$

Analogamente, dois conjuntos são interpolados através da combinação linear de Minkowski, onde o parâmetro u é variado continuamente (figura 6):

$$u * A \oplus (1-u) * B = \{u * a + (1-u) * b: a \in A, b \in B, u \in [0, 1]\}$$

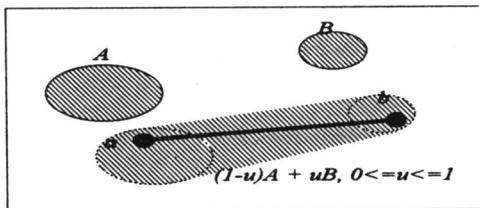


figura 6: Combinação linear de Minkowski

A combinação polinomial de Minkowski entre dois conjuntos é feita de maneira semelhante:

$$P_1(u) * A \oplus P_2(u) * B = \{P_1(u) * a + P_2(u) * b : a \in A, b \in B, u \in [0, 1], P_i(u) \text{ polinômio em } u\}$$

Assim, pode-se pensar em "Curvas" Bezier de Minkowski onde os pontos de controle, ao invés de pontos do \mathfrak{R}^n , são conjuntos do \mathfrak{R}^n .

Estes conjuntos são interpolados continuamente através dos polinômios de Bernstein [Farin (1988)]:

$$P_1(u)A_1 \oplus P_2(u)A_2 \oplus \dots \oplus P_k(u)A_k = \{P_1(u)a_1 + \dots + P_k(u)a_k: a_i \in A_i \subset \mathfrak{R}^n, P_i(u) \text{ polinômio de Bernstein de grau } k\}$$

$$P_i(u) = \binom{n}{i} (u)^i (1-u)^{n-i}$$

A Combinação de Minkowski descrita acima é usada para a geração de pinceladas da seguinte maneira: cada A_i da equação acima representa um pincel da pincelada e os pontos $a_i \in A_i$ são os pontos materiais dos pincéis.

Para implementar esta combinação, sorteia-se um ponto de cada pincel e estes pontos são interpolados, gerando uma curva de Bezier. O número de curvas de Bezier geradas é proporcional à área do maior pincel (soma das áreas das suas formas), levando-se em conta também a densidade da pincelada desejada pelo usuário. O formato da pincelada acompanha o formato da sequência de pincéis e interpola, neste caso, os pontos dos pincéis inicial e final (figura 7).

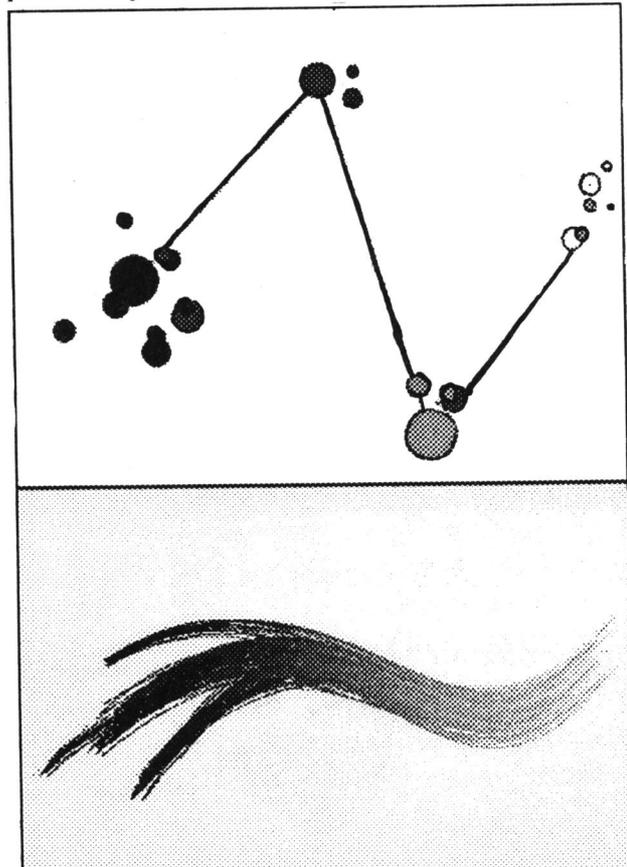


figura 7: Exemplo de sequência de pincéis com formas circulares e a pincelada Minkowski resultante.

```

PARA n=1 a n=(densidade*área do maior pincel)
{
  PARA cada pinceli da pincelada
  PARA alguma paleta do pinceli
  {
    sorteia ponto (x,y) da forma;
    pi = paleta(x,y);
  }
  interpola os pontos materiais pi usando
  curvas de Bezier;
  pontos interpolados são usados pelos
  pincéis para alterar células;
}

```

Este método "força-bruta" para a interpolação de conjuntos mostra-se adequado neste caso, pois uma textura é gerada automaticamente pela inserção de várias partículas de pintura numa mesma célula (sobreposição de várias curvas de Bezier). A pincelada amplia os efeitos obtidos com as pinceladas Sumi-E² (descritas em [Strassman (1986)]), porém de uma maneira muito mais rápida e simplificada e sem simular o fenômeno físico que o originou.

Variações da pincelada acima são obtidas usando-se, na combinação de Minkowski, qualquer outro tipo de polinomial, como splines, curvas interpoladoras lagrangeanas ou poligonais. No caso de polinomiais interpoladoras, a pincelada interpola o conjunto de pontos de todos os pincéis, o que pode ser desejável para o usuário.

- *Pincelada Mistura*

É um outro tipo de pincelada interpoladora, na qual o material de um pincel é alterado pela combinação com o material do pincel anterior da lista. Este novo conjunto de pontos é então inserido nas células. Todos os pincéis devem possuir formas iguais.

- *Pincelada Simples*

Esta pincelada copia todo o material do pincel para as células, que são alteradas de acordo com o tipo de pincel. Serve para aplicar diretamente mapas de textura sobre a tela.

```

PARA cada pincel da pincelada
PARA cada paleta deste pincel
PARA cada ponto (x,y) da forma
{
  p = paleta(x,y);
  pincel altera célula usando ponto
  p;
}

```

- *Pincelada Spray*

A pincelada sorteia pontos dos pincéis de acordo com a densidade da pincelada e o pincel usa estes pontos para alterar as células.

```

PARA cada pincel da pincelada
PARA cada paleta do pincel
PARA i=0, i<(densidade)*(área _forma)
{
  sorteia ponto (x,y) da forma;
  pi = paleta(x,y);
  pincel altera célula usando ponto
  pi;
}

```

- *Pincelada DripPainting*

É uma variação da pincelada Spray. Simula gotas de tinta (ou água) sobre a tela, através da alteração das células ao redor de $(p_i.x, p_i.y)$. O número de células alteradas ao redor de p_i varia randomicamente e a pincelada cria volumes esféricos para simular gotas. Efeitos de refração podem ser obtidos. Outra variação possível é aquela em que células inclinadas recebem menor quantidade de tinta do que células planas.

4 Implementação - Interface

O sistema roda numa estação gráfica Sun SPARCstation2, equipada com placa de vídeo Parallax, permitindo o uso de imagens de 24 bits (True Colors). O suporte gráfico, desenho na tela, manipulação e aquisição de eventos foi provido pelo pacote gráfico XWindow. Todo o código foi escrito em C++, usando os conceitos da programação orientada a objetos ([Stroustrup (1987)], [Kim et al (1989)]).

O protótipo do sistema não pretende ser um programa de pintura de propósito geral. Como tal, faltam-lhe muitas ferramentas funcionais e uma sofisticação maior na interface. A idéia básica, no entanto, é que a interface fique subordinada ao modelo e não o contrário. A partir do conhecimento do modelo que a originou, torna-se fácil entender e modificar a interface ou ir sofisticando-a aos poucos. Cada objeto possui a sua própria janela de edição e os seus próprios métodos de edição, que dependem dos recursos disponíveis. O usuário entra com parâmetros que são armazenados e utilizados durante a definição dos objetos e de seus atributos (figura 8).

Diversos tipos de facilidades de edição podem ser usados. Por exemplo, ao criar-se uma nova pincelada, é possível editar apenas os mapas inicial e final dos seus pincéis, sendo os mapas intermediários automaticamente interpolados. O mesmo pode ser feito para as formas. Se houver disponibilidade de dispositivos gráficos mais sofisticados que um simples mouse, pode-se melhorar a interação com o usuário durante a edição dos objetos. Caso haja uma "light pen" sensível à pressão, o tamanho das formas ou o volume de tinta usado é editado de acordo com a pressão usada. Este dispositivo também pode ser usado para selecionar automaticamente uma determinada forma a partir de

²Sumi-E (Sumi: tinta preta usada na caligrafia - E: figura): Arte tradicional japonesa

uma biblioteca de formas pré-definidas, simulando efeitos de "crayon", por exemplo [Bleser (1988)].

5 Resultados - Figuras

As figuras 9 e 10 foram feitas posicionando os pincéis sobre um mesmo mapa de cores e usando paletas impressionistas (foto pequena). Na figura 9 foram usadas pinceladas simples, com pincéis de formas circulares e, na figura 10, pinceladas Minkowski. Estas duas telas não foram iluminadas. A figura 11 foi gerada usando uma mapa de alturas inicial feito a partir da foto pequena e também usando pinceladas simples com pincéis de alto e baixo-relevo. A figura 12 foi feita usando-se pinceladas "drip painting" e, a seguir, pinceladas simples mas usando pincéis de alto-relevo, de modo que as alturas das células fossem pintadas seletivamente com cores diferentes. A figura 13 exhibe um conjunto de pinceladas Minkowski aplicadas sobre uma tela onde inicialmente foram usadas pinceladas "drip painting" com a mesma cor de fundo da tela. As partículas de pintura usadas são opacas e o volume delas serviu para modelar a micro-geometria das células. A tela foi iluminada por uma única fonte de luz, de acordo com o modelo de iluminação apresentado.

6 Extensões

Uma aplicação interessante é o mapeamento da tela para superfícies de sólidos em sistemas tradicionais

(Ray-Tracing, B-Rep, Splines), pois a tela já armazena os parâmetros do modelo de iluminação mais comumente usado nestes sistemas. A princípio a pintura poderia ser feita diretamente sobre uma cena tridimensional mas haveria necessidade de equipamento mais robusto e veloz para que a interação fosse em tempo real. Além disso, surgem problemas para a localização dos objetos e para o posicionamento do pincel sobre estas superfícies [Hanrahan et al (1990)].

Uma pintura pode ser pensada como uma tela e uma sequência de pinceladas. Estas pinceladas são reproduzíveis a partir do conhecimento dos seus atributos. Assim, é possível criar uma animação entre estas pinceladas a partir da interpolação do seus parâmetros. As pinceladas, no entanto, devem ser compatíveis (mesmo número e tipos de atributos, etc.).

Um grau maior de realismo é obtido ao simular-se o espalhamento da tinta e ao usar-se modelos mais acurados para a física da mistura entre materiais. Novos materiais de pintura podem ser criados e pré-definidos, de modo que o usuário tenha à disposição uma biblioteca de diferentes materiais de pintura.

Inúmeras outras técnicas de pinceladas podem ser desenvolvidas usando este mesmo modelo de objetos.

7 Conclusões

As técnicas de pintura discutidas neste trabalho são apenas algumas dentre muitas possibilidades. Através

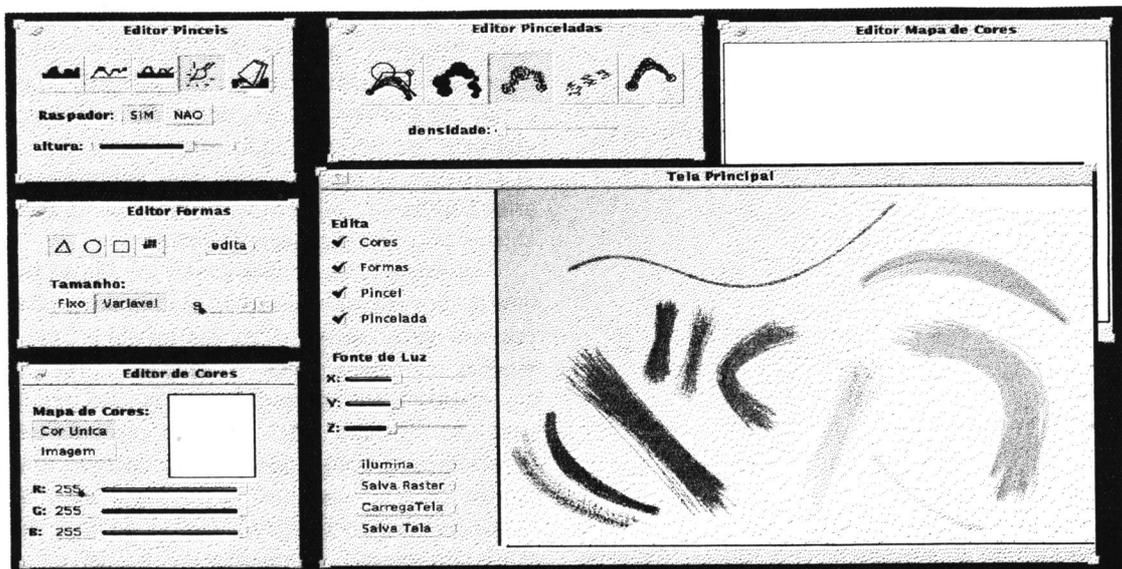


figura 8: Interface do Sistema

das diversas técnicas da computação gráfica usadas, imagens e texturas geradas algorítmicamente ou matematicamente são combinadas e incorporadas à uma composição de maneira sensível, criando novas possibilidades de expressão visual. Apesar do paradigma ser baseado na realidade do artista pode-se gerar novos tipos de resposta, conforme seja desejável ou necessário, através da criação de novas classes de objetos ou da especialização das já existentes e sem alterações drásticas na arquitetura do sistema. A interação com o artista é feita em uma linguagem que não lhe é totalmente estranha, permitindo um aprendizado rápido.

No sistema desenvolvido o tempo de resposta durante a geração das pinceladas ainda é bastante lento, sendo a etapa da iluminação a que demanda mais tempo de processamento. Outra limitação diz respeito à memória, pois o armazenamento de todos os atributos das células é bastante dispendioso, principalmente no caso de telas com uma resolução maior. Além disso, qualquer tipo de pintura digital possui limitações que são intrínsecas à sua estrutura. O processo de construção da imagem é bastante construtivista e cerebral e incorpora pouco o gestual do artista (só temos o "mouse" à disposição). Também existe o problema da escala física, que está limitada ao tamanho do monitor de vídeo. Outra questão que se coloca é a da ausência de "matéria", pois não se pode sentir a textura da imagem, apenas enxergar a sua representação visual.

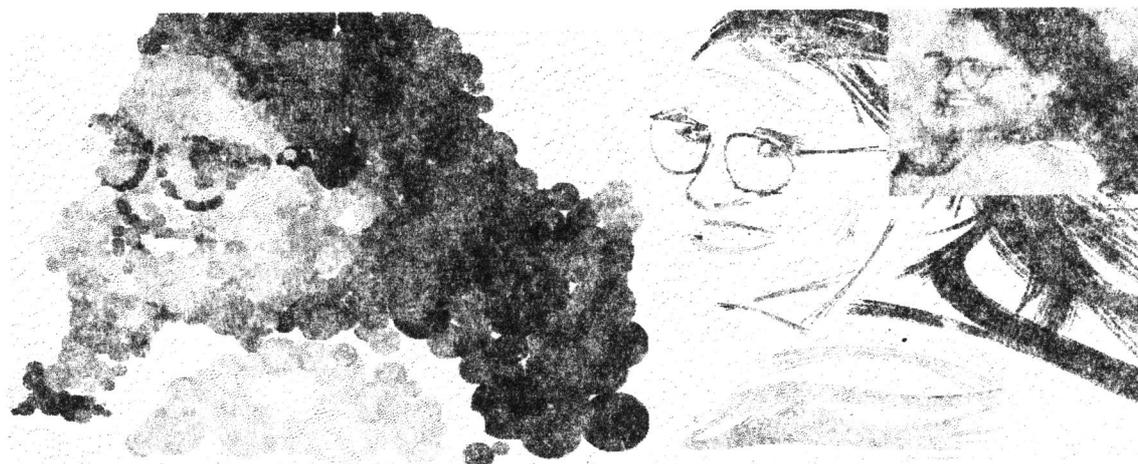
O objetivo ao se propor algumas técnicas para a pintura digital, é o de ampliar o leque de opções de artistas que usem a computação gráfica como instrumento de trabalho. Em nenhum momento pretendeu-se gerar pinturas automaticamente, pois entendo o processo de criação humana como sendo insubstituível e inimitável. A computação gráfica deve ser vista pelos artistas como mais um recurso de criação, com suas vantagens e limitações inerentes. Devem-se descobrir as linguagens específicas a este meio e torná-las cada vez mais expressivas. Cabe aqui uma última observação [Ostrower (1990)]: simplesmente chamar as imagens produzidas pelo computador de "computer art", ainda não as transforma em imagens com qualidades artísticas, isto é, formas significativas que incorporam conteúdos expressivos das vivências humanas. O mais avançado sistema de computação continua sendo um instrumento de trabalho e não um ente criador. Numa obra de arte, seja ela uma pintura, gravura ou fotografia, as técnicas devem ser invisíveis e inteiramente absorvidas pela expressividade, a própria técnica transformando-se em forma expressiva.

8 Agradecimentos

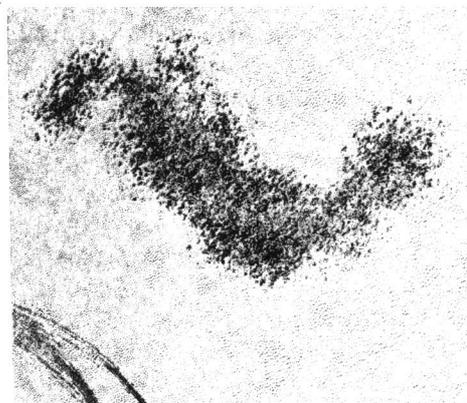
Este trabalho foi possível graças ao auxílio financeiro do CNPq e da CAPES, através da concessão de bolsas de pesquisa. A autora agradece a todos os amigos do LCG pelo apoio e sugestões, ao J. H. Gandra pelas fotos e pela zen paciência, à Carla Ferreira pela utilização da sua imagem, à Valéria Barbalho, ao André Guedes, ao José Maria Neves e ao Bebeto Tornaghi pelas leituras e, em especial, ao professor Ronaldo Marinho Persiano, orientador deste trabalho, por estes anos de constante aprendizado.

9 Referências

- Bleser, T., John Sibert e J. Patrick McGee (1988). "Charcoal Sketching: Returning Control to the Artist", *ACM Transactions on Graphics*, 7(1) pp 76-81.
- Blinn, J.F (1977). "Models of Light Reflection for Computer Synthesized Pictures", *Computer Graphics*, 11(2) pp 192-198.
- Blinn, J.F (1978). "Simulation of Wrinkled Surfaces", *Computer Graphics*, 12(2) pp 286-292.
- Cockshott, T., John Patterson e David England (1992). "Modelling the Texture of Paint", *proc. EUROGRAPHICS'92* 11(3).
- Farin, G. (1988). *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press.
- Hanrahan, P. e Paul Haeberli (1990). "Direct WYSIWYG Painting and Texturing on 3D Shapes", *Computer Graphics*, 24(4) pp 215-223.
- Haeberli, Paul E. (1990). "Paint By Numbers: Abstract Image Representations", *Computer Graphics* 24(4) pp 207-214.
- Kaul, Anil e Jarek Rossignac (1991). "Solid-interpolating Deformations: Construction and Animation of PIPs." *proc. EUROGRAPHICS'91* pp 493-505.
- Kim, Won e Frederick Lochovsky (1989). *Object-Oriented Concepts, Databases, and Applications*, Addison-Wesley.
- Ostrower, Fayga (1990). *Acasos e Criação Artística*, ed. Campus.
- Rogers, David F. (1985). *Procedural Elements For Computer Graphics*, McGraw-Hill.
- Smith, Alvy Ray (1982). "Paint", *Tutorial: Computer Graphics*, ed. K.S. Booth, IEEE Press.
- Strassman, S. (1986). "Hairy Brushes", *Computer Graphics* 20(4) pp 225-232.
- Stroustrup, Bjarne (1987). *The C++ Programming Language*, Addison-Wesley.



figuras 9 e 10



figuras 11 e 12

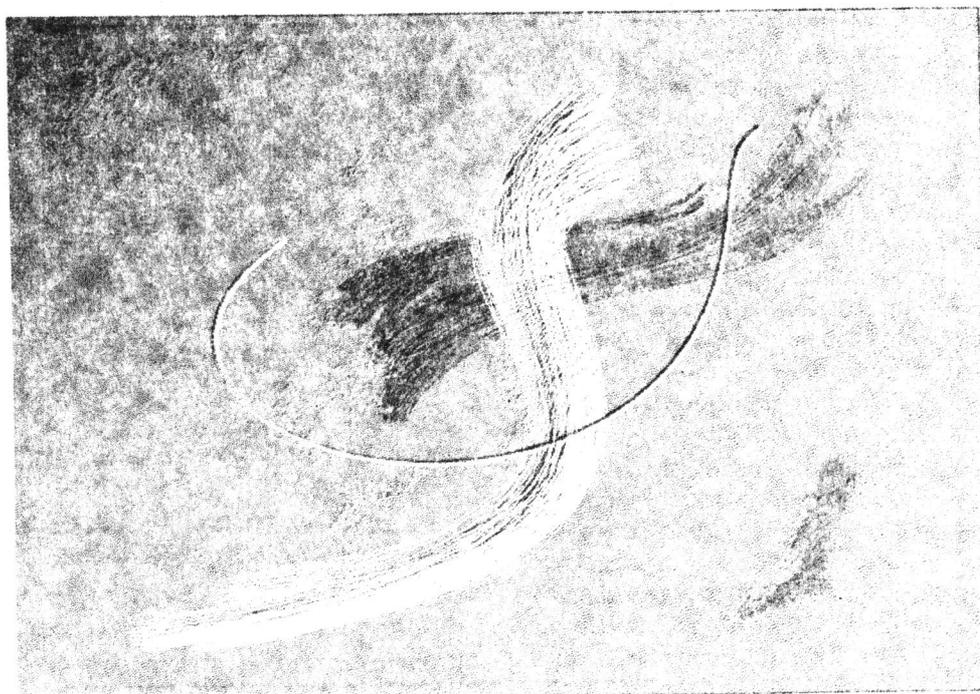


figura 13