Images Alive

IMPA – Image Processing and Computational Photography (2006)

Mário de Sá Vera
mvera@impa.br

Images Alive

- abstraction
- proposal description

Images Alive

- how to make an image "*alive*" ?

Images Alive

- one possibility is to think on movement ...
- and why not consider a natural event simulation ?

Images Alive

- Work description :
  i. description of a photorealistic rendering model to simulate rain on images or videos
  ii.implementation of an application to bring an image alive by simulating the rain rendering model
  iii.improvements for the user interface with controls
- Reference :
  - ACM SIGGRAPH 2006 official presentation

Photorealistic rendering of rain streaks

- motivation
- concepts
- model description
- architecture
- implementation details

Motivation

- give photorealistic rain effect to images , games and videos without the costs and limitations of current techniques
  - the strategy is to use image-based rendering !

other models limitations...

- ray tracing is computationally expensive
- hand draw is not robust enough (Matrix)
- static models do not consider viewing parameters dynamics (Maya, 3D-Studio and Inferno)
- mostly useful for long distance camera positioning

Concepts

- at close distances , raindrops become large streaks revealing a complex brightness intensity pattern
- raindrops shape varies :
  - particle oscillations due to aerodynamics and surface forces produces random raindrops shape modification
- lighting and viewpoint direction parameters also matters :
  - distance from the source and camera
  - size of the drop
  - camera exposure time

Model description

- first photorealistic oriented model for rain rendering :
  - based on atmospheric science particle dynamics
  - captured the interaction between Lighting + Viewpoint + Oscillations (LVO) parameters
- two steps model :
  - preprocessed images database creation
  - rendering of simple viewing parameters at real time

Image database creation

- Oscillations
  - transverse
  - oblate-prolate

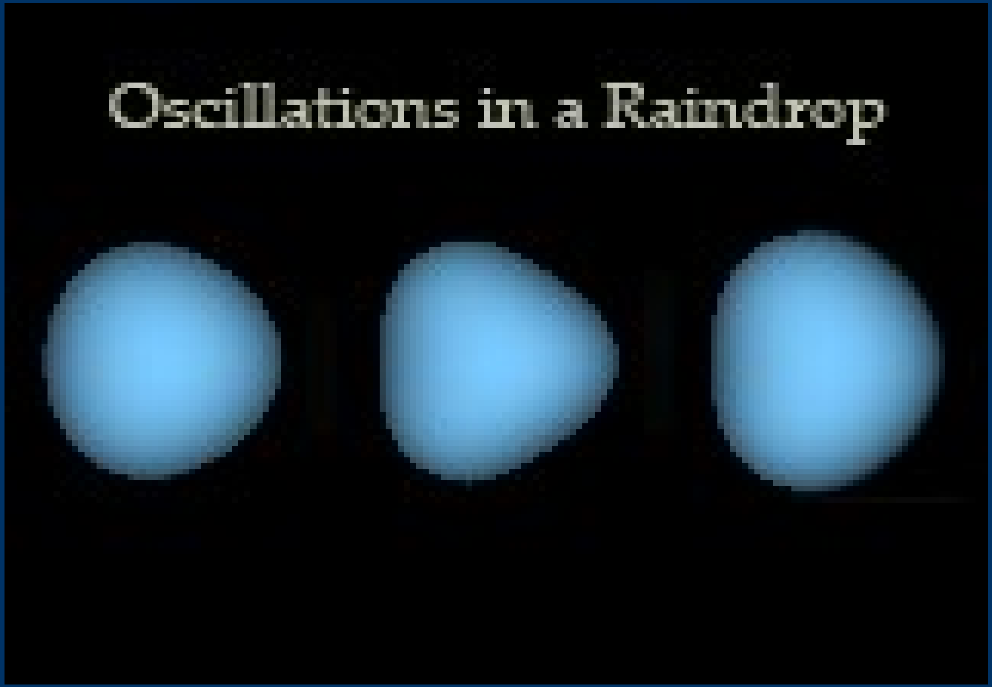


Image database creation

- Coordinate System

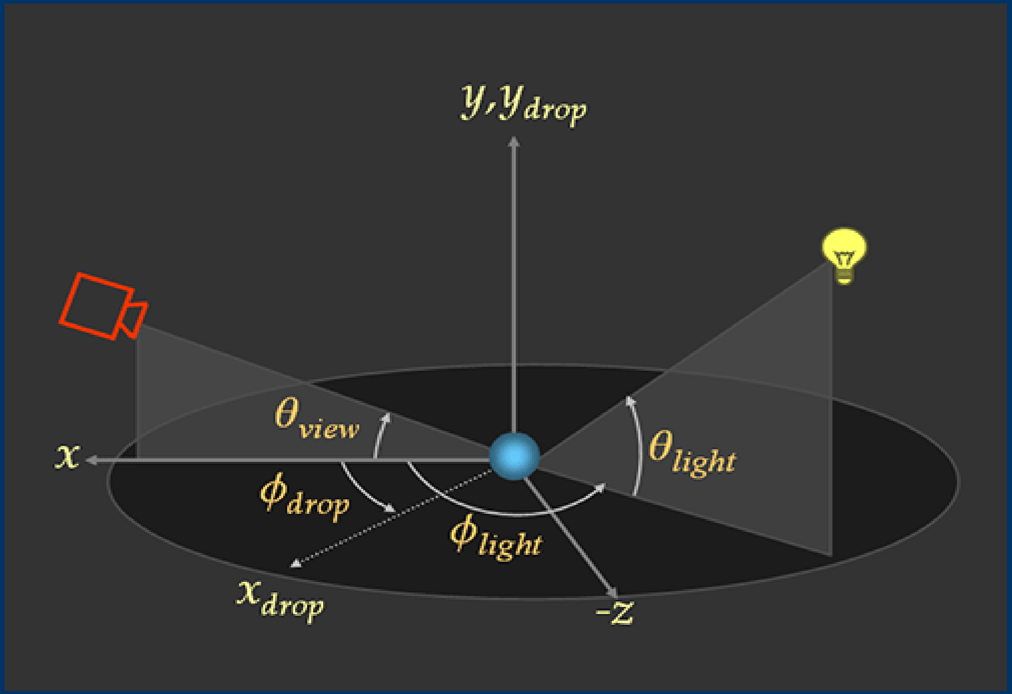


Image database creation

- the Particle shape calculation
  - A : amplitude of all spherical harmonic modes
  - P : Legendre function for theta dependency

$$r[t,\theta,\phi] = r_0\left(1+\sum_{n,m} A_{n,m}\sin(\omega_n t)P_{n,m}(\theta)\cos(m\phi)\right)$$

Image database creation

- Rendering
  - number of images = number of pixels
  - PBRT package
  - direct light surface shading
  - 16200 vertices triangle mesh
  - 16bits monochrome
  - 32 x 1024 pixels (16,8,4)
  - 6300 per resolution

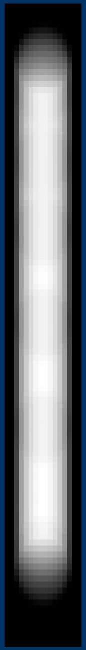
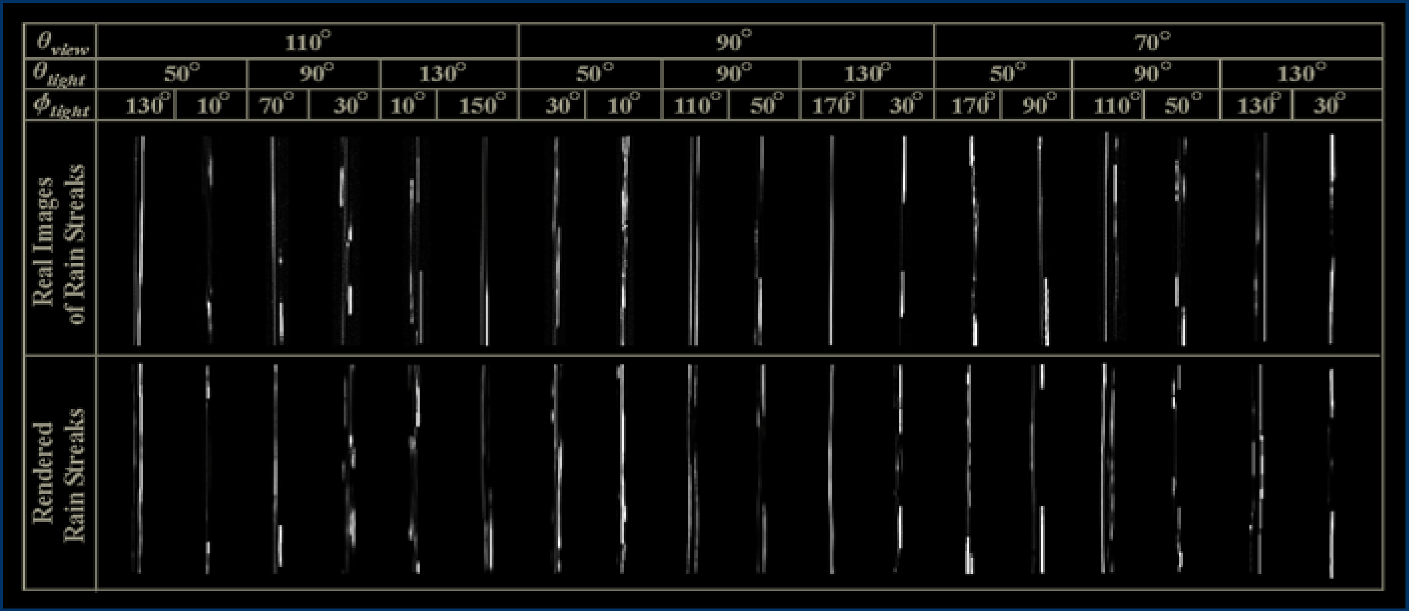
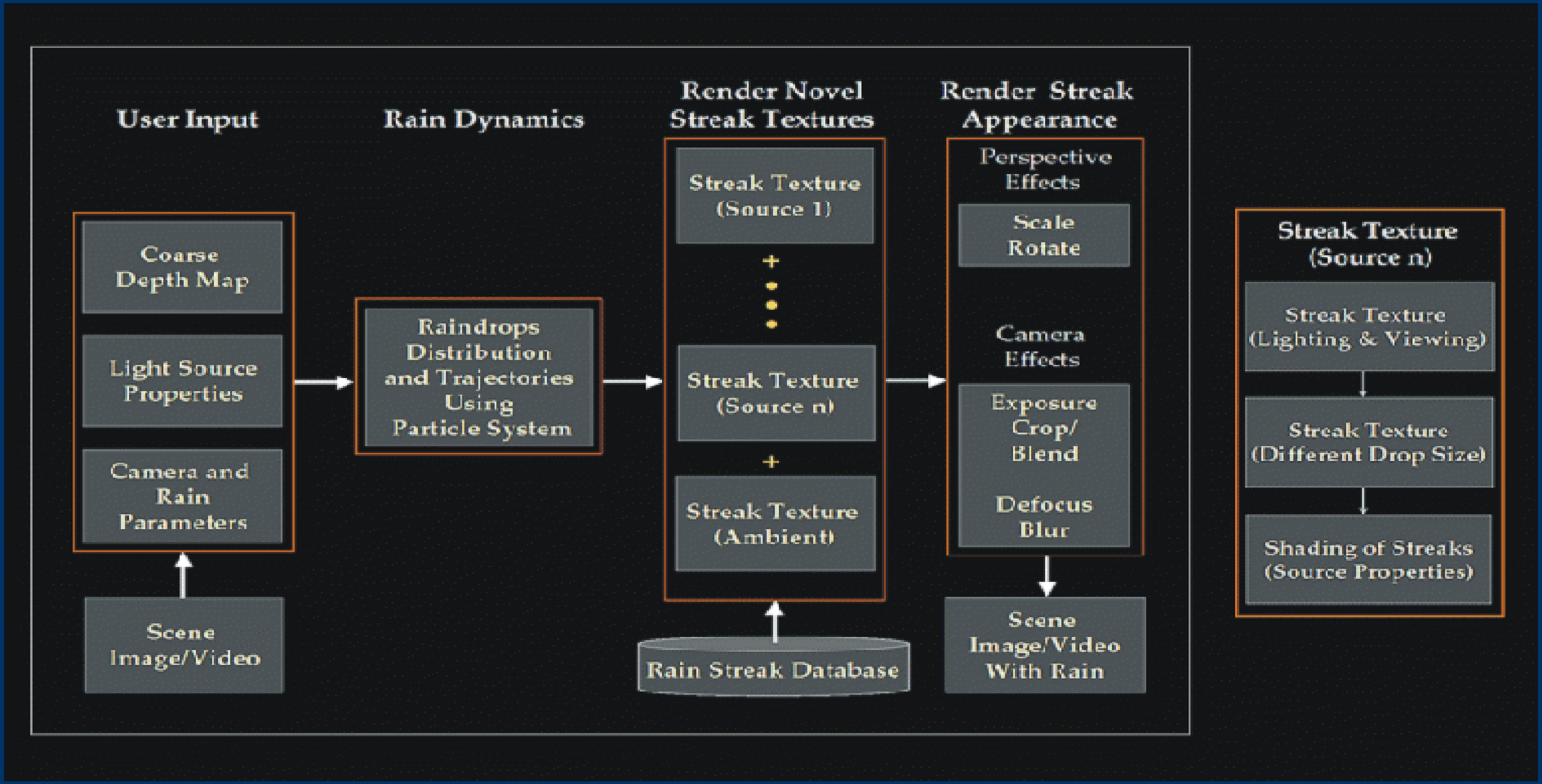


Image database creation

- Matching with real rain streak images
  - 810 HDR images
  - 3m distance using Canon EOS-20D
  - varying viewing and lighting parameters



The algorithm


User Input
Rain Dynamics
Render Novel Streak Textures
Render Streak Appearance
Coarse Depth Map
Light Source Properties
Camera and Rain Parameters
Scene Image/Video
Raindrops Distribution and Trajectories Using Particle System
Streak Texture (Source 1)
+
Streak Texture (Source n)
+
Streak Texture (Ambient)
Rain Streak Database
Perspective Effects
Scale Rotate
Camera Effects
Exposure Crop/ Blend
Defocus Blur
Scene Image/Video With Rain
Streak Texture (Source n)
Streak Texture (Lighting & Viewing)
Streak Texture (Different Drop Size)
Shading of Streaks (Source Properties)

The algorithm

- user input
  - light properties
  - coarse depth map
- spatio-temporal distribution
  - particle API
  - random Oscillation parameters
- rendering transformations
  - lighting & viewing dependencies
  - drop size
  - light source

Implementation details

- a public domain rain streaks database
- a public domain particle systems library
- imaging processing package
- openGL

Conclusions

- not a technical improvement but a strategy innovation in this particular field
- *openGL is awkward..*

References

- Photorealistic Rendering of Rain Streaks
  - http://www1.cs.columbia.edu/CAVE/projects/rain_ren/
- Imaging Processing Package
  - http://www.tecgraf.puc-rio.br/im
- Particle System
  - http://www.cs.unc.edu/~davemc/Particle/