# Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion

Paper by: Greg Turk

Richie Wales

CAP 6675 – Fall 2017

# Motivation

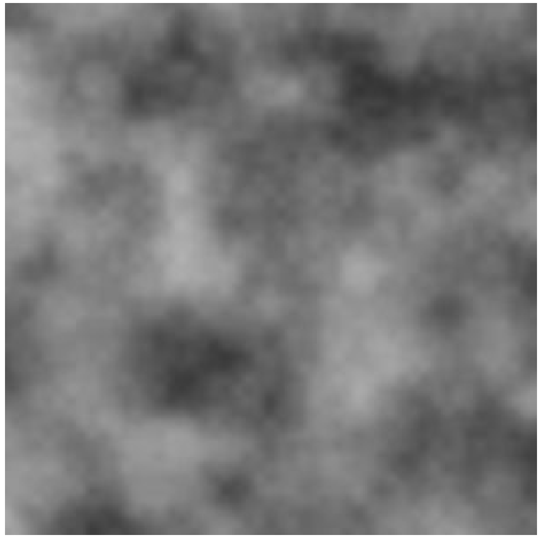
- Biological processes for creating animal coats unknown
- Reaction-Diffusion possible candidate
- How can we mimic this process for more effective texture generation and mapping to an object from 2D to 3D space?

# Goals

- Two main goals
- Pattern generation with reaction-diffusion
  - Simple patterns
  - Complex with cascading R-D multiple times
- Generate mesh and synthesize a texture directly on a mesh
  - Removes need to map texture coordinates to the polygon’s space

# Background: Texture Generation

- Gardner – Summing sine waves to create a texture [2]
  - Similar to a Fourier series
- Perlin – “Noise” [3] and Lewis – Isotropic noise functions [4]
  - Isotropic: No orientation
- Witkin and Kass – Beginnings of Reaction-Diffusion [5]



Public Domain,
https://en.wikipedia.org/w/index.php?curid=10409833

# Background: Texture Mapping

- “Assigning texture coordinates from a rectangle to the vertices of the polygons in a model”[1]
- Mapping depends on the model shape
  - Should match the object’s coordinate system
- Projecting texture onto an object’s surface
- Solid textures
  - Function defining color of the object at specific point in space

# Reaction-Diffusion

- Chemicals diffuse and interact with each other, promoting or discouraging growth of other chemicals

$$\frac{\partial a}{\partial t} = F(a,b) + D_a \nabla^2 a$$

**[**General form of a two-chemical reaction-diffusion**]**

$$\frac{\partial b}{\partial t} = G(a,b) + D_b \nabla^2 b$$ [1]

rations of *a* and *b* and the diffusion of *a* from places nearby" [1]

- $D_a$: Constant, speed of *a*'s diffusion
- $\nabla^2 a$: Concentration of *a* compared to neighbor concentrations of *a*
    - Decides where *a* will diffuse to/away from

# One-Dimensional R-D Example

$$\Delta a_i = s\,(16 - a_i b_i) + D_a\,(a_{i+1} + a_{i-1} - 2a_i)$$

$$\underbrace{\Delta b_i}_{} = \underbrace{s\,(a_i b_i - b_i - \beta_i)}_{} + D_b\,\underbrace{(b_{i+1} + b_{i-1} - 2b_i)}_{}$$

Change in chemical concentration over time

Concentrations of *a* and *b* and a scaling/shape regularity constant (Reaction of *a* and *b*)

Concentrations of chemical at a point compared to its neighbors (Diffusion)

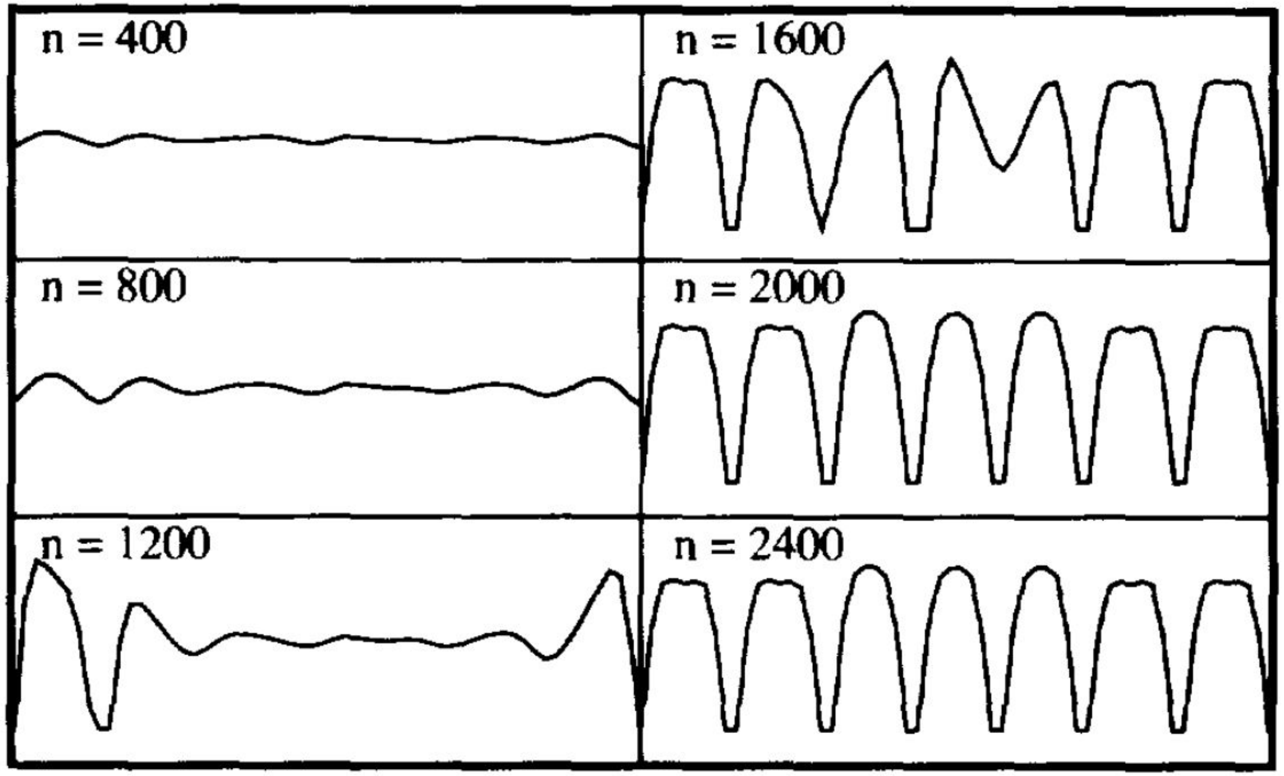


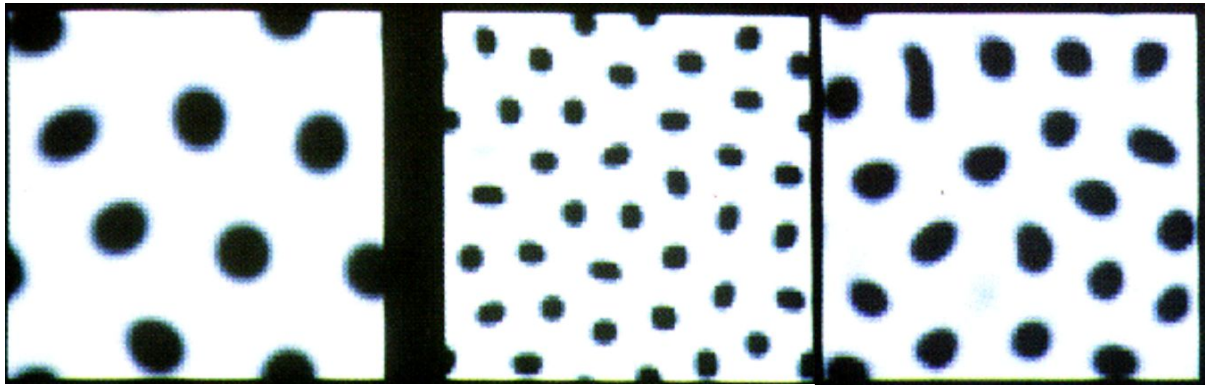

**Figure 1:** One-dimensional example of reaction-diffusion. Chemical concentration is shown in intervals of 400 time steps. [1]

# Two-Dimensional R-D

- Inhibitor: High values of the inhibitor prevent other chemicals from concentrating
- Chemical *a* → Spots → Prevent other spots from forming nearby
- Spots form in low concentrations of *b*
- *s* parameter affects spot size
- β affects consistency in shape

$$\Delta a_{i,j} = s\,(16 - a_{i,j}\,b_{i,j}) + D_a\,(a_{i+1,j} + a_{i-1,j} + a_{i,j+1} + a_{i,j-1} - 4a_{i,j})$$

$$\Delta b_{i,j} = s\,(a_{i,j}\,b_{i,j} - b_{i,j} - \beta_{i,j}) + D_b\,(b_{i+1,j} + b_{i-1,j} + b_{i,j+1} + b_{i,j-1} - 4b_{i,j})$$

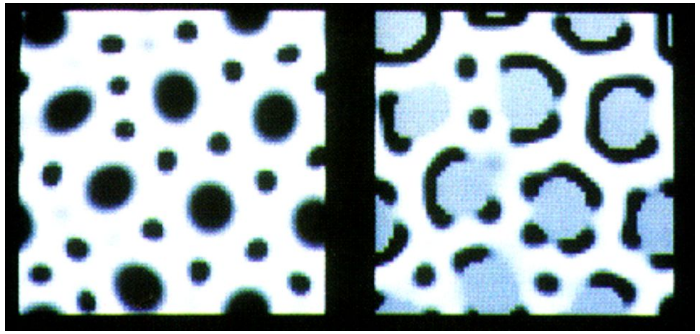


[1]

# Complex Patterns

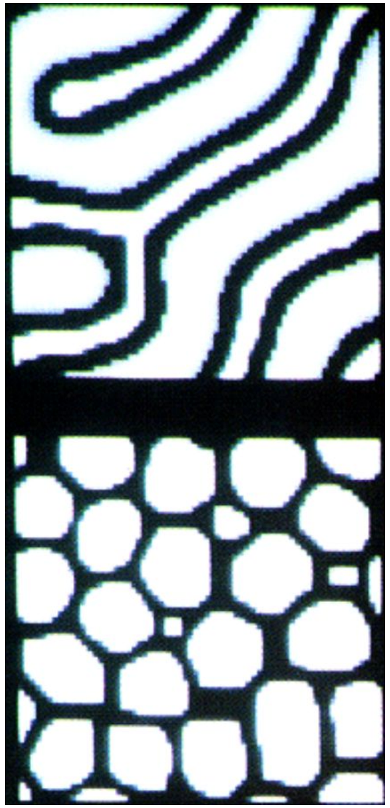
- Cascade Process
  - First system lays out a pattern
  - Second system modifies the pattern further
- Left figure (Spots)
  - Large spots first
  - Cells are “frozen”
  - Small spots synthesized in the non-frozen region
- Right figure (Leopard)
  - Large spots first
  - Cells are frozen and *a* and *b* concentrations -> 4
  - Smaller spots tend to form AROUND large spots due to concentration values

[1]



# Complex Patterns (Continued)

- Additional examples include Lionfish and Giraffe textures
- Lionfish
  - Large stripes formed first
  - Small stripes form in-between “frozen” large stripes
- Giraffe
  - Irregularly shaped spots initially created
  - Stripes form between the spots to create the “web-like pattern called reticulation” [1]



[1]

[1]

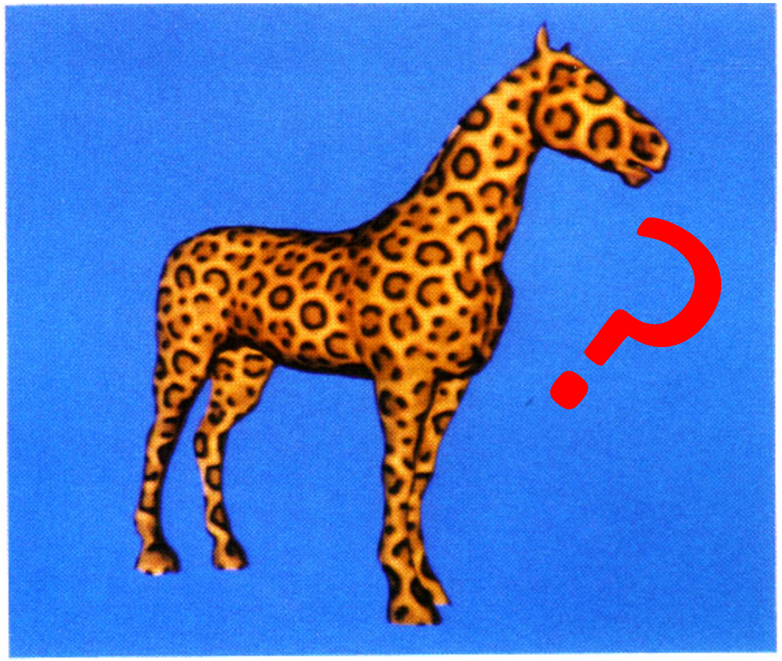


Figure 4: Leopard-Horse

[1]

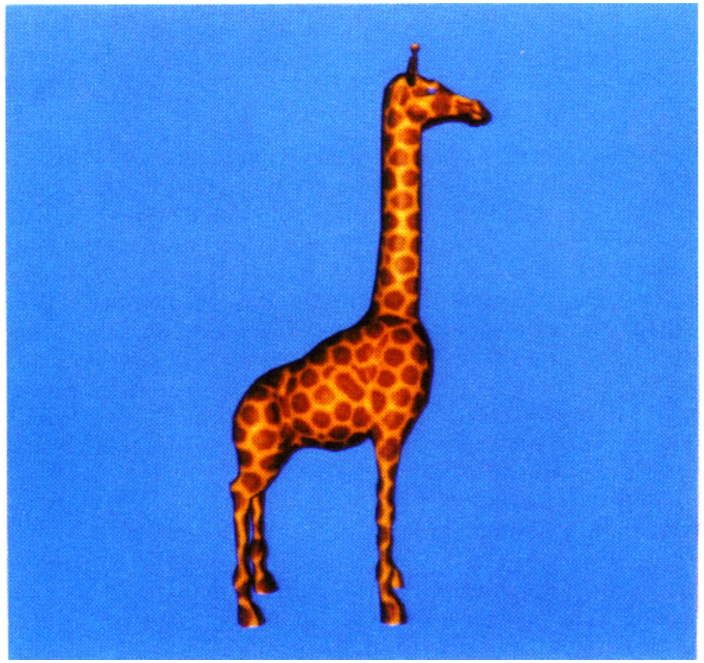


Figure 5: Giraffe

[1]

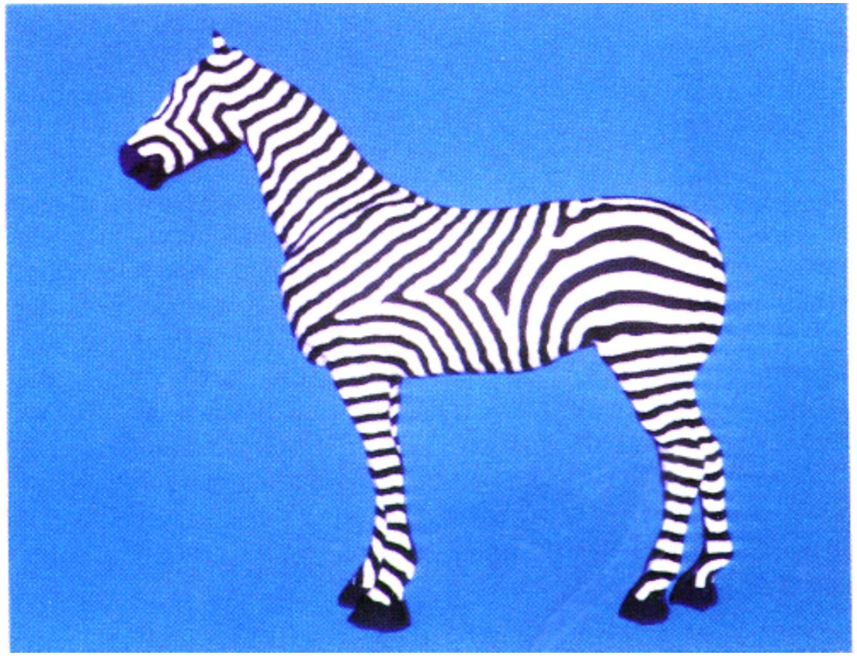
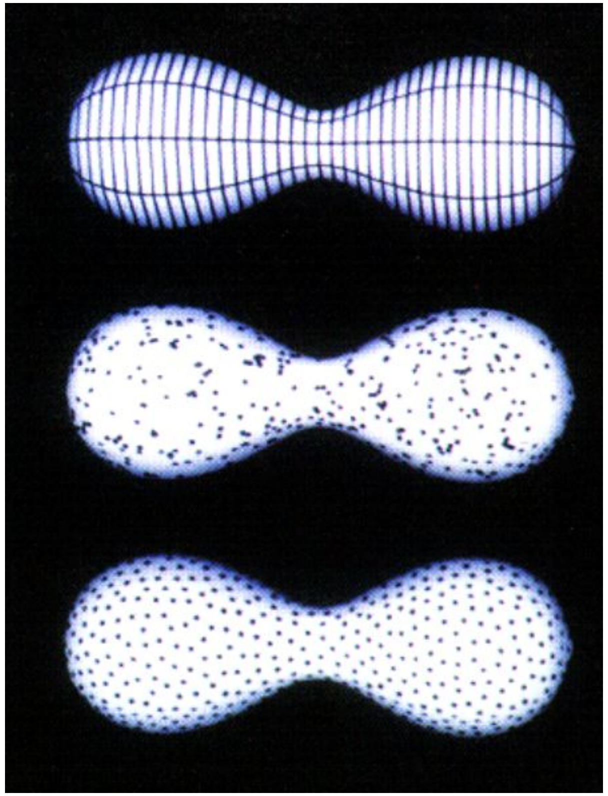


Figure 6: Pseudo-Zebra 293

Varied R-D parameters across the model so that the stripes matched its animal counterpart at every location of the body

# Mesh Generation

- Mesh must be created that matches model's shape
  - Mesh is made for texture synthesis
- Points created across mesh
- Relaxation sets these points to be evenly spaced
- Points pushed away at distance of repulsion radius
- Points can be pushed off of a polygon
  - Rotated about polygon's edge it crossed until it lies on the surface of a polygon



$r = 2\sqrt{a/n}$

$a$ = area of surface

$n$ = number of points on surface

[1]

# Voronoi Regions

- "Given a set of points *S* in a plane, the Voronoi region of a particular point *P* is that region on the plane where *P* is the closest point of all points in S" [1]
- Helps decide what cells are adjacent for R-D
- Voronoi Regions can be found in O(n) time
  - Points within distance *2r* of a point are in the Voronoi Region
- R-D is then run on the Voronoi Regions of the mesh
- Chemical concentrations found by R-D must be rendered into texture

# Rendering

- A function is used to convert chemical concentrations to texture color
- When mapping colors, repulsion radius and scaling can cause texture aliasing
  - Blurring texture colors reduces effect efficiently
  - Supersampling (rendering texture with higher resolution and scaling down) not efficient
- Bump mapping possible by using the gradient of the cell colors
  - Perturbation vector added to “surface normal” to give lighting effect of bumpy or rough texture [3]
  - Normal used “to determine a surface's orientation toward a light source” [8]

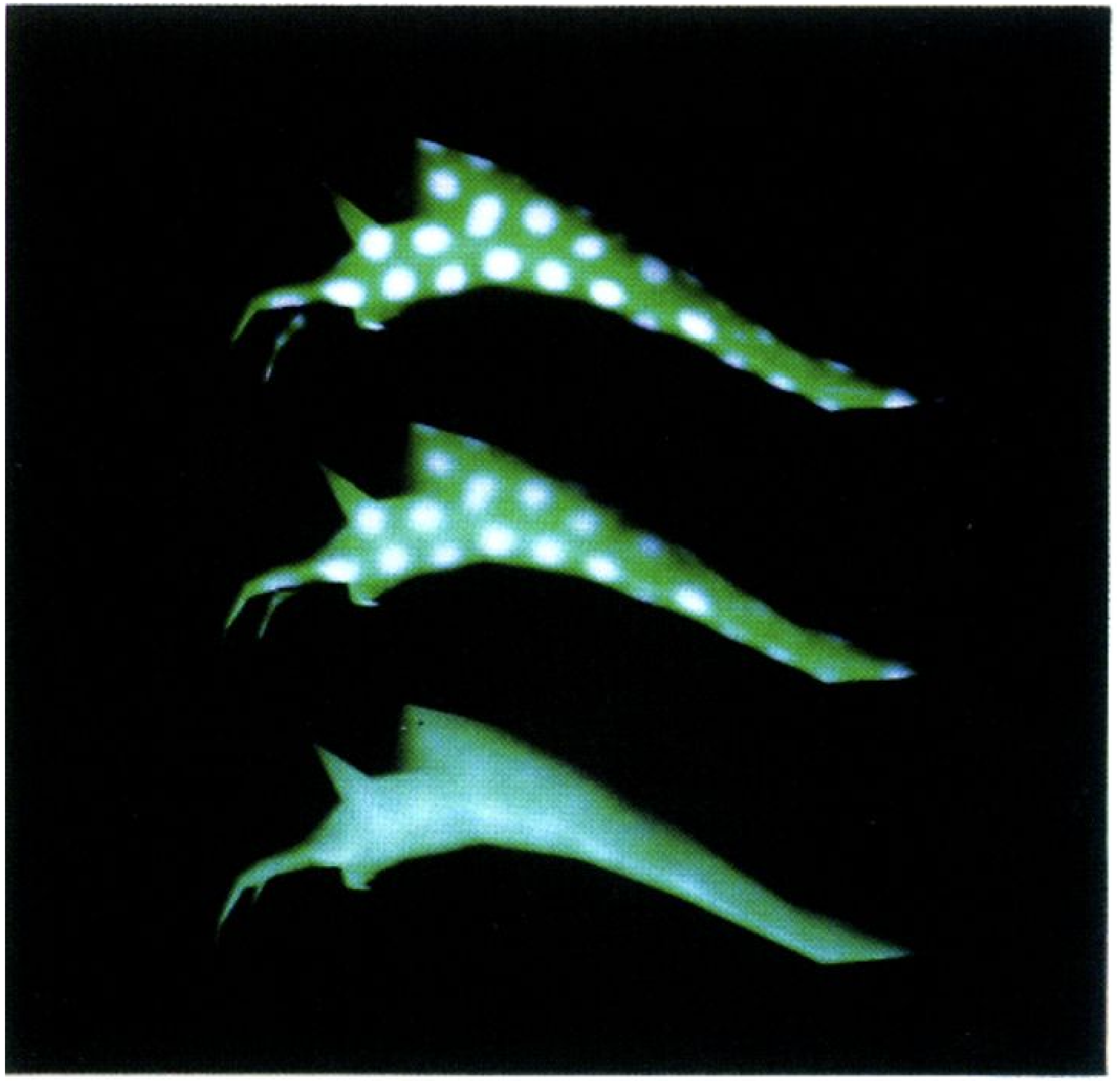
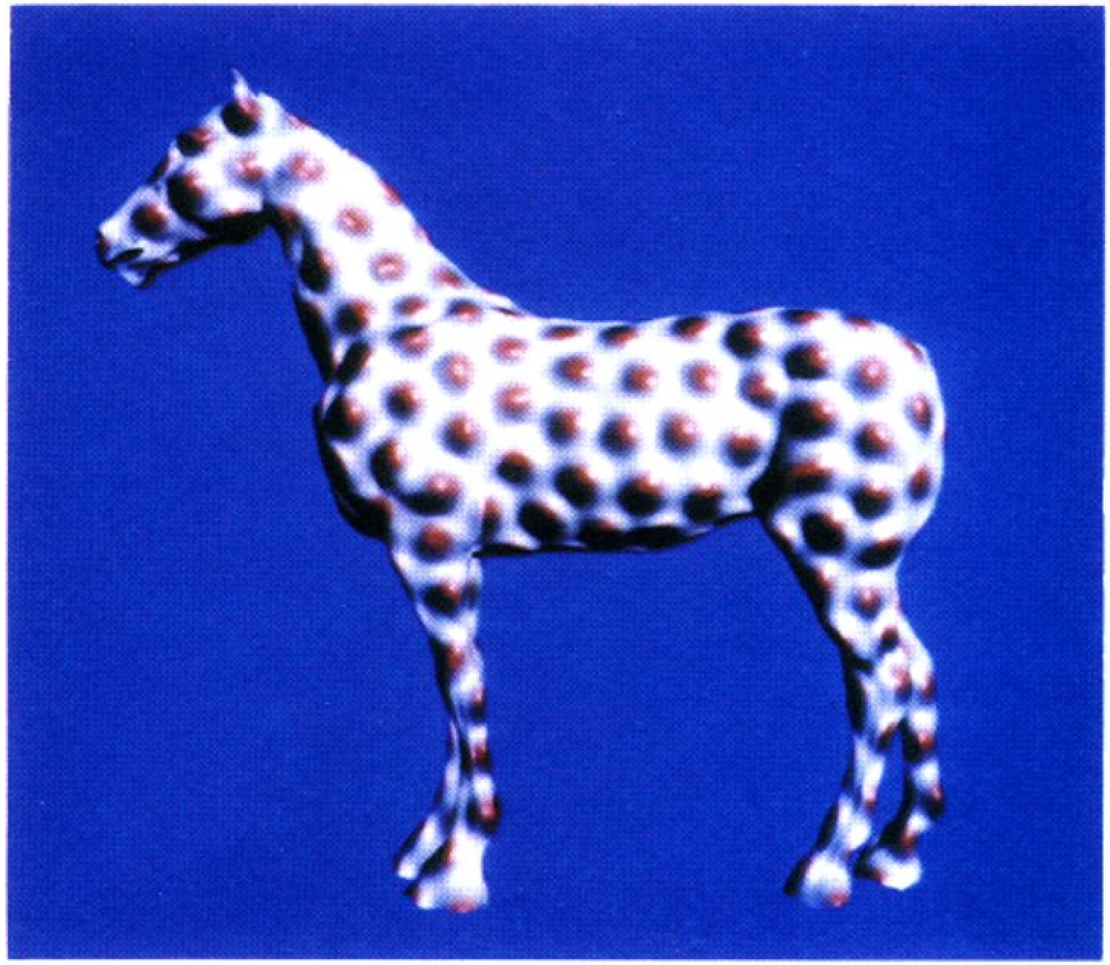


**Figure 9:** Blur levels for anti-aliasing

[1]



'96

**Figure 10:** Bump mapping

[1]

# Conclusion

- Took several hours to render each texture with reaction-diffusion
- Rendering models was much quicker
- Leopard Horse took 70 seconds to render to the model at 512x512
- Model without texture took 16 seconds
- "Such texture synthesis times would seem to prohibit much experimenting with reaction-diffusion textures" [1]

# Strength 1 – In-Depth Background on Alternative Methods

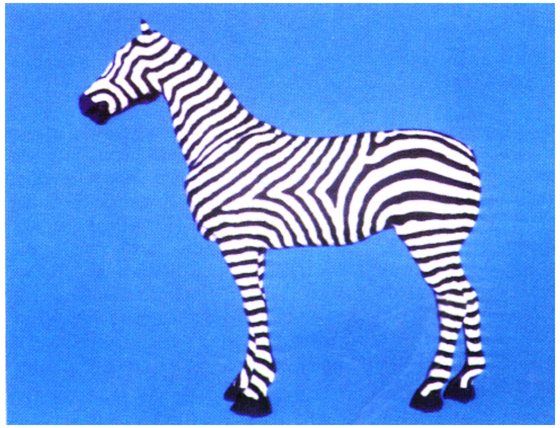
- Provides the reader with a solid background on texture synthesis and rendering methods
- Shows how the problem domain is important
- Validates their work as an effective alternative with performant results
- Their research addresses a growing area (developmental biology) with texture generation

# Strength 2 – Slowly Developed Complexity

- Initially show small reaction-diffusion demos affecting size/shape
- Add cascading process to create more complex dualities between chemicals
- Add multiple (5) chemicals to system for additional intricacy
- Include mesh generation for practical applications
- Shows each step individually to see how they all contribute to a complex system

# Strength 3 – Many Applications for Research

- Inherent applications in developmental biology
  - Used reaction-diffusion to accurately generate popular animal coats
  - Validates the process as possible candidate in embryo development
- Computer graphics/Videogames, Movies, Etc.
  - Procedural content generation, especially with dynamic meshes
  - Creating unique textures time consuming. Mapping them to dynamic shapes even more difficult.



[1]

# Weakness 1 – Relatively Simple Models Used

- Good introduction into texture/mesh generation
- For practical uses, 3D models in simulations and games are much more complex
  - Polygons continuously added to modern models
- On a workstation at the time of publishing, each texture “took several hours to generate” [1]
- Might not be as useful in real world
- Does the runtime linearly scale with model detail? (Voronei region detection is O(n))

# Weakness 2 – Liberties Taken in Code

- Paper is focused on biological systems
- Made algorithm choices that would not mimic biology
- To get appealing stripe patterns…
  - Run specific “Meinhardt” stripe algorithm [1]
  - Freeze cells that belong to a white stripe region
  - Run another system with altered parameters to create thinner stripes
- Too many specific changes so that their desired behavior would emerge
- Perhaps explain why this liberty was taken
  - Example of this “freezing” occurring in nature

# Weakness 3 – Too Focused Biologically

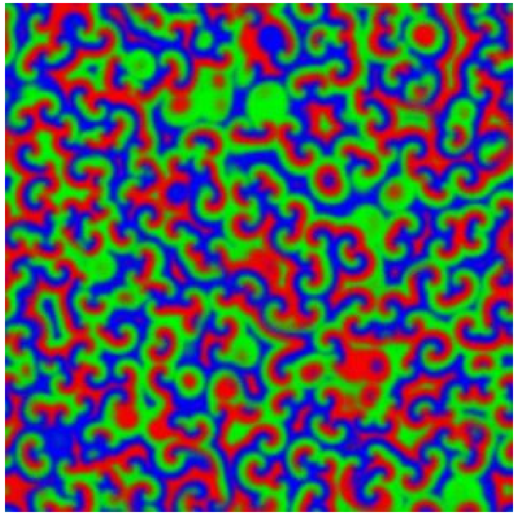
- Every pattern they discussed was deeply rooted in animal coloring
- Might have suppressed other aesthetically pleasing results by focusing on biology
- Texture generation could be important in other areas
    - Games as discussed previously
    - Design (Graphics or abstract interior design)
- What NEW patterns can we find, rather than replicate?
- If this paper focused more on the texture synthesis than developmental biology, more fields could benefit from their work
    - Put themselves into a ”niche”

# Extension 1– Branching Out of Biology

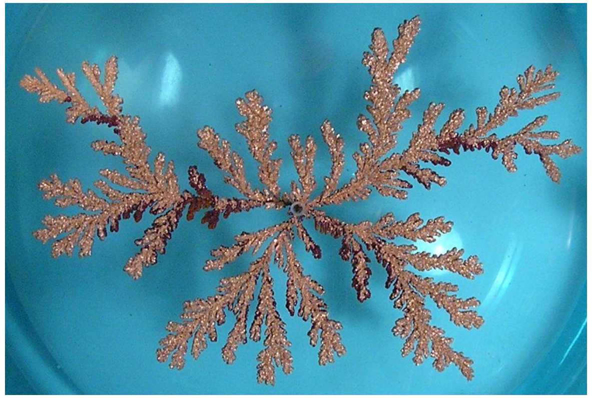
- This study was a good introduction in reaction-diffusion
- Recreating animal textures can validate the system’s performance
- Create a more exploratory study in the future
- What texture synthesis results are possible that are unique and can be applied in other fields or systems?
- Is reaction-diffusion’s performance acceptable for computer graphics work? Could it change the field of procedurally generated content (PCG)

# Extension 2 – Additional Texture Generation Methods

[6]



- Borrow from Cellular Automata and reality
- Belousov–Zhabotinsky (BZ) reaction
  - Theoretical Chemistry reaction without equilibrium
  - Evolve “chaotically” [6]
- Diffusion-Limited Aggregation
  - Seen in some organic systems, like mineral deposits
  - Studied through computer simulations [7]



[7]

# References


[1] Turk, Greg. "Generating textures on arbitrary surfaces using reaction-diffusion." *ACM SIGGRAPH Computer Graphics*. Vol. 25. No. 4. ACM, 1991.

[2] Gardner, Geoffrey Y. "Visual simulation of clouds." *Acm Siggraph Computer Graphics*. Vol. 19. No. 3. ACM, 1985.

[3] Perlin, Ken. "An image synthesizer." *ACM Siggraph Computer Graphics* 19.3 (1985): 287-296.

[4] Lewis, John-Peter. "Algorithms for solid noise synthesis." *ACM SIGGRAPH Computer Graphics*. Vol. 23. No. 3. ACM, 1989.

[5] Witkin, Andrew, and Michael Kass. "Reaction-diffusion textures." *ACM Siggraph Computer Graphics* 25.4 (1991): 299-308.

[6] Wikipedia contributors. "Belousov–Zhabotinsky reaction." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 11 Sep. 2017. Web.18 Sep. 2017

[7] Wikipedia contributors. "Diffusion-limited aggregation." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 12 Sep. 2017. Web.18 Sep. 2017

[8] Wikipedia contributors. "Normal (geometry)." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 12 Jan. 2017. Web.20 Sep. 2017