

SciVis 2020 – Project Overview

Konrad Polthier, Henriette Lipschütz

Freie Universität Berlin

May 12, 2020

Expectations/To Dos

- Send your three favorite projects until May, 22nd, 8:15 pm, via email to Henriette.Lipschuetz@fu-berlin.de.
- On May, 22nd, you will be told with whom you are going to work on which project.
- In the end of June, there will be a short presentation (appr. 10 min.) of the state of the art of your project. Date: tbn.
- code submission: July, 17th, 2020, 6 am, via email
- submission of documentation (pdf): July, 17th, 2020, 6 am, via email

01: Seam Carving for Content-Aware Image Resizing



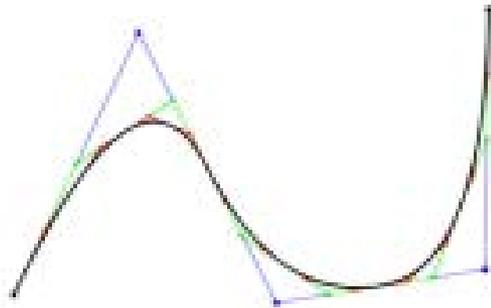
Abstract: Effective resizing of images should not only use geometric constraints but consider the image content as well. We present a simple image operator called seam carving that supports content-aware image resizing for both reduction and expansion. A seam is an optimal 8-connected path of pixels on a single image from top to bottom, or left to right, where optimality is defined by an image energy function. By repeatedly carving out or inserting seams in one direction we can change the aspect ratio of an image.

Literature: Shai Avidan & Ariel Shamir, Seam Carving for Content-Aware Image Resizing, Siggraph.
See: [Project 01](#)

Expectation:

- Implement the image resizing algorithm.
- Evaluate on three representative images of your choice.
- Discuss the strengths and weaknesses of the algorithm.

02: Spline Editor



Abstract: Implementation of a spline editor where a user can generate a control polygon and then a tangent continuous spline curve is generated. The spline consists of segments of Bezier curves with are connected with tangent continuity.

Literature: Gerald Farin, Curves and Surfaces for CAGD. A practical guide, 5th edition, Academic Press, San Diego, 2002
See: [Project 02](#)

Expectation:

- (a) Implement the creation and editing of splines in the plane.
- (b) The control polygon should be editable by the user.
- (c) A project panel provides controls for adding and removing spline control points or enforce continuity.

03: Image Deformation

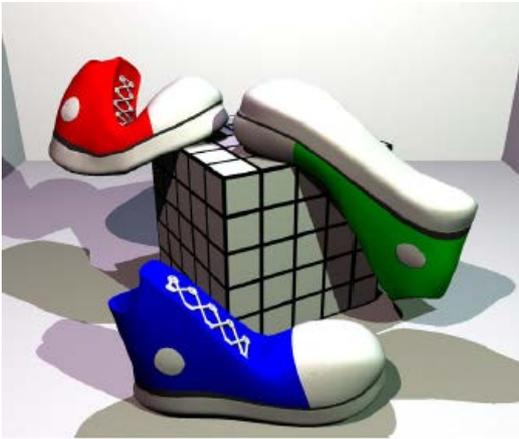


Abstract: We provide an image deformation method based on Moving Least Squares using various classes of linear functions including affine, similarity and rigid transformations. These deformations are realistic and give the user the impression of manipulating real world objects. For each of these techniques, we provide simple closed form solutions that yield fast deformations, which can be performed in real time.

Literature: S. Schaefer, T. McPhail, and J. Warren, Image Deformation using moving least squares, ACM Transactions on Graphics, 2006.
See: [Project 03](#)

Expectation: (a) Implement the algorithm presented in the paper.
(b) Evaluate its pros and cons.

04: Meshless Deformation based on Shape Matching



Abstract: We present a new approach for simulating deformable objects. The underlying model is geometrically motivated. It handles point-based objects and does not need connectivity information. The approach does not require any preprocessing, is simple to compute, and provides unconditionally stable dynamic simulations.

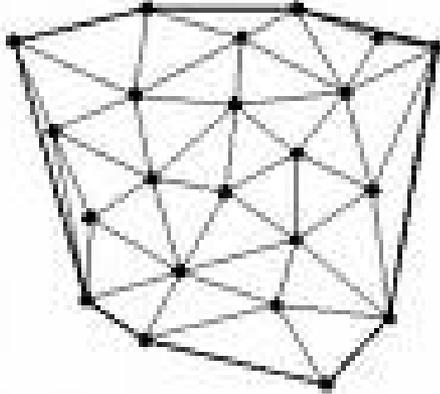
Literature: M. Müller, B. Heidelberger, et al., *Meshless Deformation Based On Shape Matching*, ACM Transaction on Graphics, 2005.

See: [Project 04](#)

Expectation:

- Implement the algorithm presented in the paper.
- Extend the method to linear and quadratic deformation as proposed in the paper.
- Demonstrate the method for a falling cube and a classical minimal surface hitting a plane.

05: Delaunay Triangulation in the Plane



Abstract: Given n points in the plane in general position, the aim is to find a triangulation that has these points as vertices and satisfies the following condition: The circumcircle of each triangle does not contain any vertex in its interior. The algorithm proceeds by inserting new points. The Delaunay triangle is determined that contains the new point and a subdivision of this triangle is performed. The resulting triangulation may not be a Delaunay triangulation. To repair this defect, edges are flipped.

Literature: M. de Berg, M. van Kreveld, M. Overmars, and O. Cheong, Computational Geometry, Section 9, Springer Verlag, 3rd edition, 2008.

See: [Project 05](#)

Expectation:

- Implement the algorithm presented in the paper.
- Allow the user to change the number and position of the points.
- Visualize how the algorithm works.

06: Simulation of Crowd Motion



Abstract: We present a real time crowds models based on continuum dynamics. In our model, a dynamic potential field simultaneously integrates global navigation with moving obstacles such as other people, efficiently solving for the motion of large crowds without the need for explicit collision avoidance. Simulations created with our system run at interactive rates, demonstrate smooth flow under a variety of conditions, and naturally exhibit emergent phenomena that have been observed in real crowds.

Literature: A. Treuille, S. Cooper, and Z. Popevic, Continuum Crowds, Siggraph, 2006.
See: [Project 06](#)

Expectation: (a) Implement the algorithm presented in the paper.
(b) Evaluate its pros and cons.

07: Particle Fluids

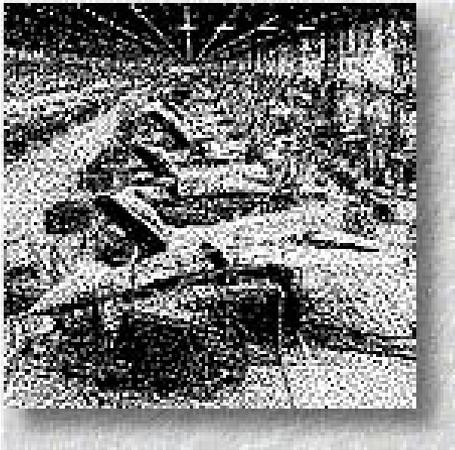


Abstract: We present novel adaptive sampling algorithms for particle based fluid simulation. We introduce a sampling condition based on geometric local feature size that allows focusing computational resources in geometrically complex regions, while reducing the number of particles deep inside the fluid or near thick flat surfaces. Further performance gains are achieved by varying the sampling density according to visual importance.

Literature: B. Adams, M. Pauly, et al., Adaptively Sampled Particle Fluids, ACM Transaction on Graphics, 2007.
See: [Project 07](#)

Expectation: (a) Implement the algorithm presented in the paper.
(b) Evaluate its pros and cons.

08: Digital Halftoning with Space Filling Curves

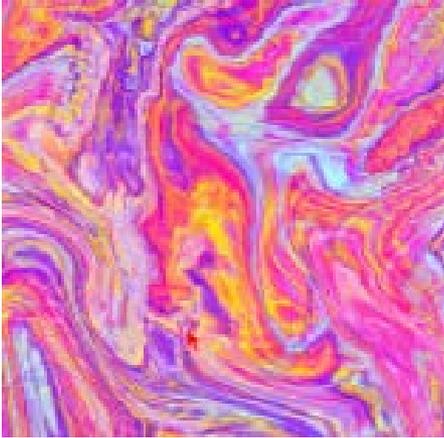


Abstract: This method is a new digital halftoning technique that uses space filling curves to generate aperiodic patterns of clustered dots. It allows the parameterization of the size of pixel clusters which can vary in one pixel steps. The algorithm unifies in this way, the dispersed and clustered-dot dithering techniques.

Literature: Luiz Velho, Digital Halftoning, Siggraph, 1991.
See: [Project 08](#)

Expectation: (a) Implement the algorithm presented in the paper.
(b) Create a space filling curve mimicing a gray level distribution.

09: Stable Fluids

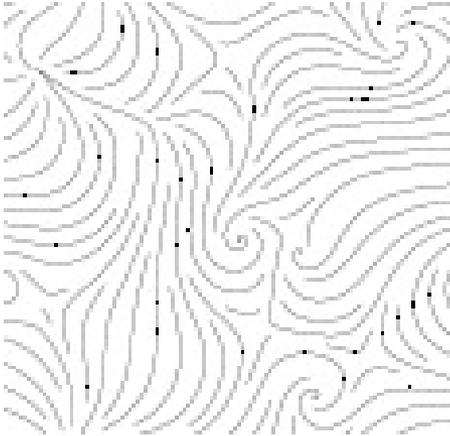


Abstract: Creating realistic fluids remains a challenging and interesting problem in computer graphics. Given the ever increasing demand for convincing physical simulations, the intent of this project is to implement a fluid solver based on the semi Lagrangian method first introduced by Jos Stam in Stable Fluids.

Literature: J. Stam, Stable Fluids, Siggraph 1999.
See: [Project 09](#)

Expectation: (a) Implement Stam's algorithm.
(b) Evaluate its pros and cons on sample flows.

10: Evenly Streamlines



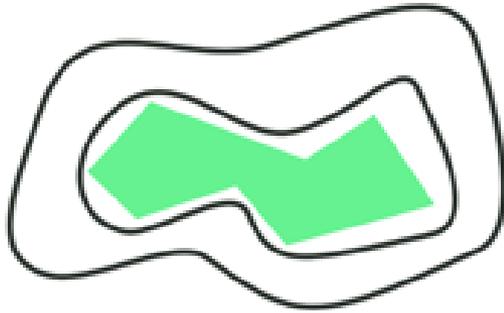
Abstract: We propose a novel algorithm for placement of streamlines from two dimensional steady vector or direction fields. Our method consists of placing one streamline at a time by numerical integration starting at the furthest away from all previously placed streamlines. Such a farthest point seeding strategy leads to high quality placements by favoring long streamlines, while retaining uniformity with the increasing density.

Literature: A. Mebarki, P. Alliez, and O. Devillers, Farthest Point Seeding for Placement of Streamlines, INRIA, 2006.
See: [Project 10](#)

Expectation:

- Implement the propose streamline visualization technique.
- Provide an interface to evaluate a user given vector field.
- Test the algorithm for various input vector fields.

11: Snakes in 2d Images

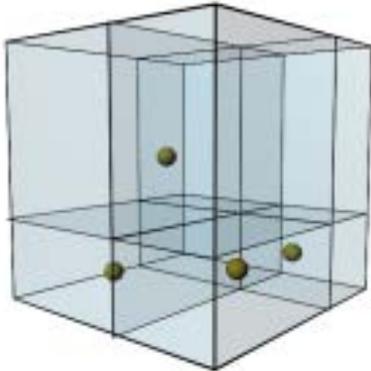


Abstract: A snake is an energy minimizing spline guided by external constraint forces and influenced by image forces that pull it toward feature such as lines and edges. Snakes are active contour models: they lock onto nearby edges, localizing them accurately.

Literature: M. Kaas, A. Witkin, and D. Terzopoulos, Snakes: Active Contour Models, International Journal of Computer Vision, 1988.
See: [Project 11](#)

Expectation: (a) Implement the 2d snakes algorithm for shape detection.
(b) Start with an image and a user defined curve, propagate this curve according to the snakes energy from the paper.

12: Visualization of kd trees



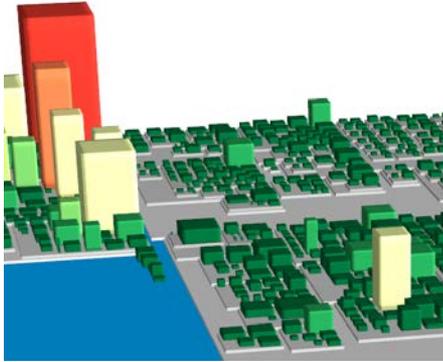
Abstract: A kd tree is a data structure to accelerate point location queries in d dimensional space.

Literature: M. de Berg, M. van Kreveld, et al., Computational Geometry, Section 9, Springer Verlag, 3rd edition, 2008.
See: [Project 12](#)

Expectation:

- Implement a kd tree structure as described in the reference.
- Create a kd tree for an own point set.
- Visualize the created partition similar to the teaser image.

13: Tree Maps



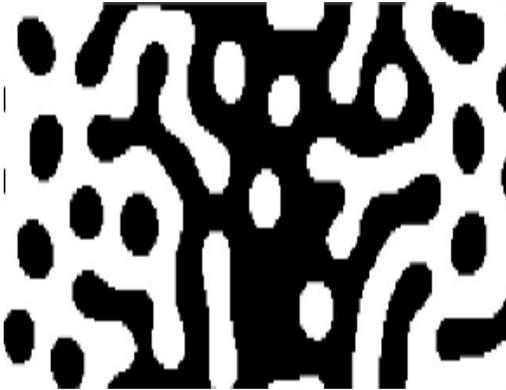
Abstract: Tree maps present hierarchically ordered data in form of a tree. Every branch is depicted as a rectangle separated into smaller rectangles which represent underlying branches. A leaf has an area proportional to some quantity all data has in common. A color code enables the user to read off further properties.

Literature: B. Johnson, B. Shneiderman, Tree visualization with Tree-maps: A 2d space-filling approach, IEEE, 1991.
See: [Project 13](#)

Expectation:

- (a) Implement 3D tree maps.
- (b) Visualize an additional quantity through height.
- (c) Discuss the strengths and weaknesses of the algorithm.

14: Turing Pattern On Cellular Automata



Abstract: Turing, Young, and McCabe considered the formation of pattern in fur of animals. In 1984, Young used cellular automata to model such patterns. Some parameters of the cellular automaton can be varied.

Literature: G. Greenfield, Turing-like Patterns from Cellular Automata, Bridges, 2016.
See: [Project 14](#)

Expectation:

- (a) Implement the cellular automaton.
- (b) Implement different neighborhoods.
- (c) Discuss specific properties of this automaton.

15: City Generation



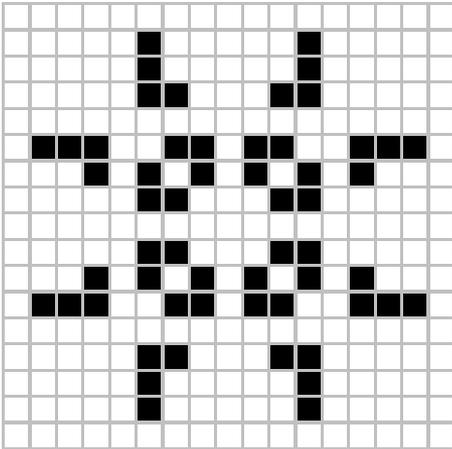
Abstract: The creation of virtual realities gets more cost- and time-intensive due to increasing demands and complexity. This plays an important role in the video game industry. An interesting task in this area is the generation of cities which creates random surroundings under given assumptions.

Literature: G. Kelly and H. McCabe, A Survey of Procedural Techniques for City Generation, ITBJ, 2006.
See: [Project 15](#)

Expectation:

- Implement a 2d city generator containing three components: street, house, and nature.
- Provide a random generation and a user definable input size.
- Formulate five useful rules for the generation algorithm.

16: Game of Life



Abstract: The Game of Life by J. Conway is a cellular automaton simulating life following some simple rules. On a regular grid, the rules are applied iterative to a given starting configuration.

Then the state of living of the cells is determined. The default rules are set in a way such that it is difficult to foretell the state of each cell in the next iteration.

Literature: M. Gardner, The fantastic combinations of John Conway's new solitaire game of life, *Scientific American*, 1970.
See: [Project 16](#)

Expectation:

- Implement a cellular automaton with random and editable starting configurations.
- Discuss the effect of choosing different neighborhoods.
- Evaluate other regular grids (triangles, hexagons, etc.) and discuss the differences.