Seeing with the Mind: from the Matrix into the Cloud

Pedro Campos and Ana Isabel Portugal
Center for Mathematical Sciences
University of Madeira
Campus Universitário da Penteada
9000-390 Funchal, Portugal
E-mail: {aic, pcampos}@uma.pt

Abstract

In this paper, we aim at showing how to use a mathematical abstraction as a way of supporting a visual representation. A *matrix* is an abstract concept that is usable as a support to a data structure, which can represent objects from the real world. Its essential functionality is grounded on the order it provides to all its elements. This connection can be strengthened by another mathematical abstraction (e.g. a law of movement) which limits the way each point can move in relation to the others.
A *cloud* is a set of connected water drops, which are therefore close to each other. If we have a set of drops loosely coupled, the cloud will vanish.
We used the matrix concept and a law which represents the strength of connection between drops as a way to represent the cloud set through a restricted programming language, Java, using Processing as a subset of that language in order to visualize graphically that object from the real world. The visualization of the screen cloud is therefore made through its decomposition in points stored in elements / ellipses / drops inside a matrix. Also, a mathematical model of the relative movement of the points is used to visualize the cloud in the place and with the shape the artist wants. So, the abstract conceptualization imagined in the artist's mind is then visually represented in the computer screen and can be appreciated by others.

Introduction

This paper is based on promoting creativity using mathematics as a way to support visual representations. In fact, the work of a computer engineer consists of observing the real world and modeling the reality so that is can be translated into a restricted language, the programming language, and consequently implemented in a given computer software. To achieve that, the engineer uses specific tools. Afterwards, the target users validate the operational model. This process is analogous to the creation of an artistic work. An artist has specific tools to model the reality and translate that model into any given support media. The spectator adheres to its representation by validating it, reinterpreting it in a sensorial way. And this is because he or she enjoys it. There is always an appropriation of reality according to a personal view, which is made available, and followed by a validation. This is, in essence, the link between the mathematical aspects of computer science and the creativity of the artistic creation.

In order to be understood, the part of the reality that engineers model is represented using formal structures which allow its representation but also its implementation control. We can observe the same in the artistic representation. Also in this process, there are rules used, such as perspective mathematical rules, geometrical primitives, composition rules and pattern generation rules.

In systems development theory [1] the type-S problems are those which can be validated by using its own specification, and they are completely static, since its real world applicability is independent of its own conception. For instance, the implementation to a matrix inversion problem is always the same and

outside of the real world, but can be used in any given domain of discourse and be the solution to a problem of that domain.

Can the artist manipulate this kind of conceptual solutions and use them in the creation of his or her artworks? Can he see with the mind? The answer is yes, and in fact many artists have been doing it since the 50's. In 2007 Casey Reas and Ben Fry, students of John Maeda at the Aesthetics and Computation Group of the MIT created an application aimed at facilitating the usage of the computer as an artistic expression means, and with their software an important step was given towards this goal of linking artistic creativity and mathematics [2]. This link is the focus of this paper and we used Reas and Fry's software, called Processing, as a practical tool, which allows digital painting. In the following section, we will demonstrate the materials and methods used for that purpose, in particular the example of using a matrix [3] as a way to artistically represent a cloud.

Materials and Methods

We use the concepts of a mathematical matrix and a law of movement to illustrate the link from the mathematical abstraction to creative visual expression in Processing. A *matrix* is an abstract concept that is usable as a support to a data structure, which can represent objects from the real world. Its essential functionality is grounded on the order it provides to all its elements. A mathematical model is another abstract concept of the relative movement of the points, which sets the connection between them. This connection can be strengthened by this mathematical abstraction (e.g. a law of movement) which limits the way each point can move in relation to the others.

A cloud is a set of tiny, connected water particles, which are therefore close to each other and made of water or ice, suspended and in movement in the atmosphere [4].

Using the matrix concept as a way of representing that set is an interesting, mathematical visualization exercise. We use that concept as a way to represent the cloud set through a restricted programming language, Java, using Processing [2] as a subset of that language in order to visualize graphically that object from the real world. The visualization of the screen cloud is therefore made through its decomposition in points stored in elements / circles / droplets inside a matrix.

We can think of a cloud as a set of circles with some connection between them so that they don't become dispersed in the atmosphere and keep visible. We want them to move along the skyline and dynamically change their shapes. As a tutorial example, we consider 2000 water particles set in a given position around the main water particle according to the following mathematical rules:

px[i] = dropX+sin(radians(angle[i]))*radiiX[i]
py[i] = dropY+cos(radians(angle[i]))*radiiY[i];

which are valid for the coordinates of the remaining points to be drawn, always maintaining a relation between them. With these coordinates, we obtain a matrix containing all the points of the cloud. Using that matrix, we can draw each of the ellipses, which represent the water droplets:

ellipse(px[i], py[i], cellRadius[i], cellRadius[i]);

This is as long as we also save the cell radius of the corresponding ellipse in another matrix. After that, we introduce some randomness in the size, position and connection of each droplet and redraw it taking into account not letting the drawing slip out of the screen drawing window.

In this sense, a cloud is also, in this line of thought, two matrixes that keep their drawing coordinates, some other matrixes containing the values of their radiuses and movement velocities, and a mathematical rule which describes the sinusoidal movement of its points.

Visually the cloud exists in the screen until the program is stopped, with the following appearance:
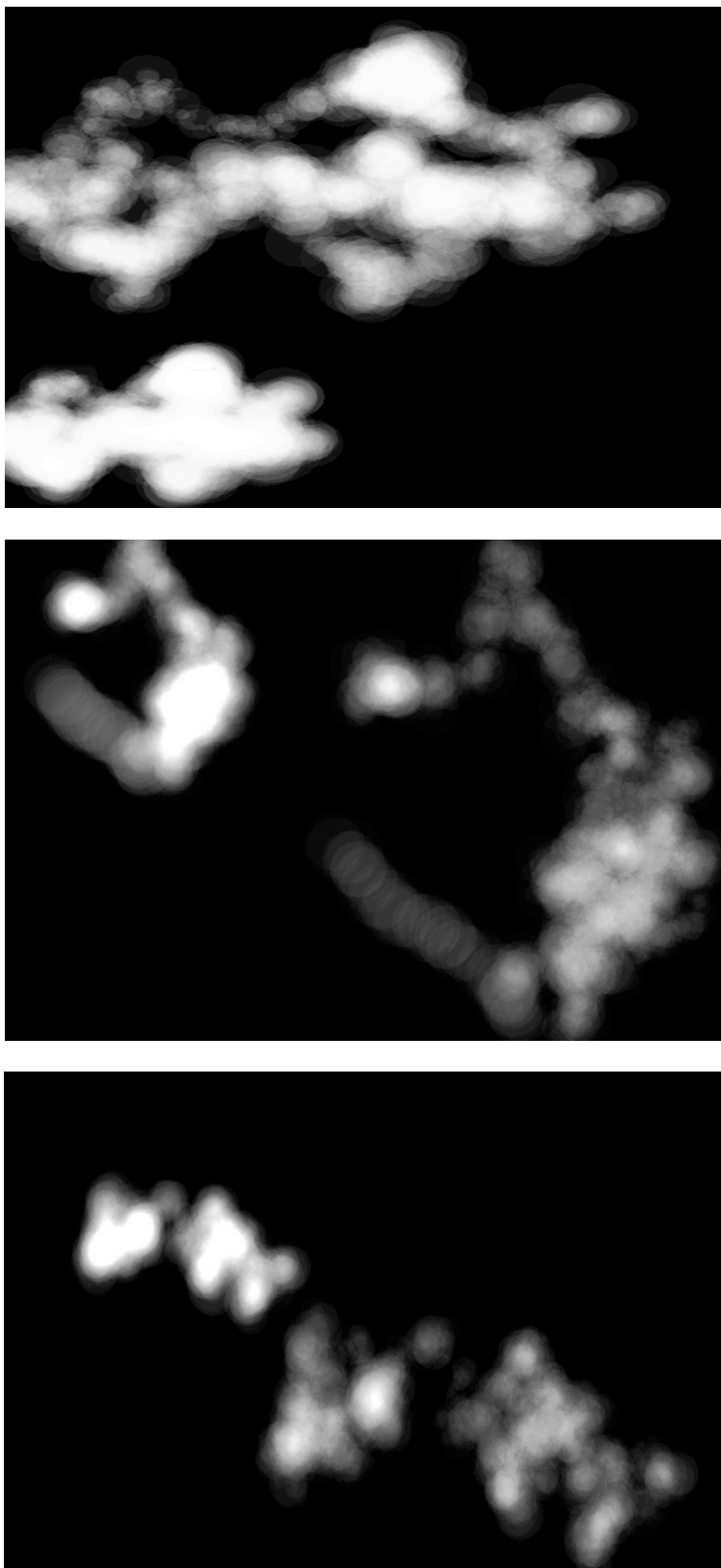
Figure 1: *Some snapshots of the animated cloud in Processing.*

Conclusions

"*Freedom of expression is the matrix, the indispensable condition, of nearly every form of freedom*" Benjamin N. Cardozo, Palko v. Connecticut, 302 U.S. 319 (1937)

A matrix is a rectangular array of numeric or algebraic quantities subject to mathematical operations. If the model is correct, what happens in the computer screen is much closer to the reality than a non-digital drawing. This kind of abstract reasoning, linking mathematical concepts to reality modeling and artistic conceptualizations can be used to model other visualizations, for instance a flock of birds or a dancer skirt. The Processing language is a new artistic means of expression, whichever concepts the artists wish to convey in their artworks.

This paper's title is "Seeing with the mind" because our approach is based on this artistic conceptualization materialized by the use of the Processing language. We started out with the description of the cloud, i.e. what a real world cloud is, how we see and perceive it, how we understand it and how we can finally describe it in a mathematical structure so that it can be modeled, so really we can see with our mind.

References

[1] M M Lehman and L A Belady. *Program Evolution - Processes of Software Change*, Academic Press, New York, 1985, 552 pps. (ISBN 0-12-442441-4).

[2] Reas, Casey and Fry, Ben. *Processing: a programming handbook for visual designers and artists*. The MIT Press Cambridge, Massachusetts, London, England, 2007.

[3] Shores, Thomas S. *Applied Linear Algebra and Matrix Analysis*. Springer Science+Business Media, LLC, 2007.

[4] Vianello, R. L., Alves. *Atlas Internacional de Nuvens*, 2010.
http://www.fisica.ufc.br/lfnm/html/nuvens.html