Quantum Image Dynamics — an entertainment application of separated quantum dynamics

Ulrich Mutze *

A bijective mapping is established between the set of pure qubit states and the set $[0,1]^3$. This latter set corresponds naturally to the RGB data which most digital image formats associate with the color of pixels. This allows to associate with a digital color image (considered as a $[0,1]^3$-valued matrix) a rectangular lattice of qubits, where for each pixel there is a qubit, the state of which is determined by the pixel's color data according to the correspondence mentioned above. We thus associate with a digital color image an idealized physical system. We define a law of dynamical evolution for this system in a manner that not only the initial state but also each evolved state can be represented as a color image. This will be done in two steps: 1. A Hamiltonian is specified which represents interaction of the pixel-based qubits with a homogeneous magnetic field together with a Heisenberg spin interaction between adjacent qubits. 2. Evolution is defined not as the exact quantum dynamics defined by the specified Hamiltonian, but as the approximate quantum dynamics which treats this interaction via the time-dependent Hartree equations and thus leaves each qubit in a pure state, to which there corresponds a well-defined color. This approximate dynamics is computationally very cheap with a computational complexity proportional to the number of pixels, whereas the complexity of exact dynamics is well known to grow exponentially with that number. This method allows to evolve a digital color image, thus producing a 'movie' from it. In such a movie the image undergoes changes which may be considered as interesting graphical effects in a first phase. In the course of further evolution, the larger and obvious structures of the image fade away and what remains is a dull, grainy, uniformity of seemingly random origin. Since, however, the evolution scheme is reversible, the initial image can be recovered from the final image by reversed evolution. Two examples of such evolving images are presented, one together with the reversed evolution.

*www.ulrichmutze.de

1 Introduction

It is common practice to represent Schrödinger wave functions on a 2-dimensional domain as color images by representing the complex values of the wave function as colors. Here the absolute value of a complex number determines the *brightness* of the corresponding color and the phase determines the *hue*. The third independent quality of colors, the *saturation*, can not naturally (*together* with brightness and hue) be represented by a single complex number. Therefore, a natural color image (in which colors make use of their three degrees of freedom) cannot be used directly as an areal Schrödinger wave function on a rectangular domain. If, however, the image is deprived of its saturation degree of freedom (e.g. by setting saturation always to its maximum representable level [1]) it can be used as a Schrödinger wave function, but the natural color image cannot be reconstructed from it (since information was deleted).

A quantum analog to natural colors occurs with qubits. As is well-known, the pure states of qubits (i.e. quantum systems with a two-dimensional Hilbert space) can be represented in a natural manner as points which form the surface of a sphere: *Bloch's sphere*. If we don't ignore the overall phase (which is irrelevant when the task is to characterize the quantum state of an isolated qubit, but which matters if such states are subject to manipulations such as interference with other qubit states) we have in addition to the two *Bloch angles* θ and ϕ an overall phase angle and thus three parameters for which it is only a technical matter to map them onto the space of natural colors [2]. This suggests that one can associate with each pixel of a digital color image a pure qubit state. We give a concrete version of this connection in the next section. It actually makes no direct use of Bloch's angles but employs related quantities. A following section describes the corresponding quantum lattice system, and a final section presents pictorial examples.

2 Correspondence between colors and pure qubit states

Any normal computer system allows to set the color of screen pixels for given integer-valued screen coordinates i and j and integer-valued 'RGB-values'. Although for most systems the ranges of R, G, B are restricted to $\mathcal{C} := \{0, 1, \ldots, 255\}$ [3], it is convenient to abstract from this limitation by using the continuous range $\mathcal{R} := [0, 1]$ for these quantities (in actual computations one then replaces 8 bits for representing colors by 64 bits for representing a real number). By transformations provided by *color management systems* one may let these data triplets play a role different from the direct control of the

[1] Saturation is maximal for monochromatic light. Normal tri-chromatic display systems (e.g. LCD) are unable to reproduce highly saturated colors.

[2] Of course, also impure states (which form the interior of Bloch's sphere) could be chosen to correspond to natural colors. Since in the dynamical scheme to be used here, no impure states arise, we do not follow this path.

[3] This is a reasonable technical compromise as the success of the 24-bit RGB ('true-color') image data formats in digital photography, image processing, and printing shows.

display color channels. They thus may control brightness, hue, and saturation directly. The pictorial examples in this article make no use of non-trivial color management.

A natural correspondence between these two sets is given by the two maps

$$\alpha : \mathcal{C} \to \mathcal{R}\,, \quad i \;\mapsto\; \frac{i+\frac{1}{2}}{256} \tag{1}$$

and

$$\beta : \mathcal{R} \to \mathcal{C}\,, \quad x \;\mapsto\; \text{if } x<1 \text{ then } \lfloor 256\cdot x\rfloor \text{ else } 255 \tag{2}$$

which satisfy $\beta\circ\alpha = \mathbf{1}$ so that converting an image from conventionally digitized color to 'continuous' color does not change the information: converting back to conventionally digitized color does not induce changes of the original digital image.

All colors that a normal computer systems can show on screen are obtained if the modified RGB-values vary over $\mathcal{R}^3$. A substantial part of color science [1] deals with the equivalence relation on $\mathcal{R}^3$ defined as 'indiscernibility by a typical human visual observer'. In conclusion, we may consider $\mathcal{R}^3$ as a mathematical model for the space of colors, and a $\mathcal{R}^3$-valued matrix as a model for a ('pixelized') color image.

On the other hand, as we will see immediately, $\mathcal{R}^3$ is related to the set $\mathbb{C}^2_1 := \{(z_1,z_2)\in \mathbb{C}^2 \,:\, |z_1|^2+|z_2|^2=1\}$ of 'normalized qubit wave functions'. A reasonably simple connection is

$$\gamma : \mathbb{C}^2_1 \to \mathcal{R}^3\,, \quad (z_1,z_2) \;\mapsto\; \left(\frac{1+\sin\phi_1}{2}, \frac{1+\sin\phi_2}{2}, |z_2|\right)\,, \tag{3}$$

where

$$\phi_1 := \arg(z_1)\,, \quad \phi_2 := \arg(z_2)\,, \quad \text{where} \quad \arg(0) := 0\,. \tag{4}$$

This function is continuous except on the subsets $\mathbb{C}^2_{1\uparrow} := \{(z_1,0)\in\mathbb{C}^2_1\}$ and $\mathbb{C}^2_{1\downarrow} := \{(0,z_2)\in\mathbb{C}^2_1\}$ of its domain. The continuity is clear from the representation $\sin\phi_i = \Im z_i/|z_i|, i=1,2$ which is free of singularities outside $\mathbb{C}^2_{1\uparrow}\cup\mathbb{C}^2_{1\downarrow}$. For showing the discontinuity on $\mathbb{C}^2_{1\uparrow}$ we consider the sequence $s : n\mapsto \mathrm{e}^{in}/n$ and an arbitrary point $(w,0)\in\mathbb{C}^2_{1\uparrow}$. The $\mathbb{C}^2_1$-valued sequence $n\mapsto (w,s_n)/\sqrt{1-n^{-2}}$ converges to $(w,0)\in\mathbb{C}^2_{1\uparrow}$, whereas its image $n\mapsto (\frac{|z_1|+\Im z_1}{2|z_1|\sqrt{1-n^{-2}}}, \frac{1+\sin n}{2}, \frac{1}{n\sqrt{1-n^{-2}}})$ does not converge (due to the oscillatory behavior of its second component). Discontinuity on $\mathbb{C}^2_{1\downarrow}$ follows from a similar argument. An implementation of γ is function `R3 CpmImaging::Color::toR3()const` in [5] and [6].

Due to this lack of continuity, states near to $\mathbb{C}^2_{1\uparrow}$ get represented by colors (via γ) which depend sensitively on the state. The same is true for states near $\mathbb{C}^2_{1\downarrow}$. We have

$$\gamma(\mathbb{C}^2_{1\uparrow}) = \mathcal{R}\times\{\frac{1}{2}\}\times\{0\}\,, \quad \gamma(\mathbb{C}^2_{1\downarrow}) = \{\frac{1}{2}\}\times\mathcal{R}\times\{1\}\,. \tag{5}$$

Therefore, colors for which the blue component is close to 0 or close to 1 should be expected to show strong variations (between red and green in the first case and between cyan and magenta in the latter case) if the state (represented by this color) undergoes

small changes. This was indeed observed in several of my experiments. By obvious modifications of (3) one may have red or green play the role that blue plays here. The mapping γ is easily seen to be surjective. It has, therefore, a right-inverse mapping. This can be chosen as the following function

$$\delta : \mathcal{R}^3 \to \mathbb{C}_1^2 , \quad (x_1, x_2, x_3) \;\mapsto\; (r_1 \, e^{i\phi_1}, r_2 \, e^{i\phi_2}) , \tag{6}$$

where

$$r_1 := \sqrt{1 - {x_3}^2} , \quad r_2 := x_3 , \quad , \quad \phi_1 := \arcsin(2x_1 - 1) , \quad \phi_2 := \arcsin(2x_2 - 1) . \tag{7}$$

An implementation of δ is function `CpmImaging::Color::Color(R3 const&)` in [5] and [6]. This mapping allows us to associate with a digital color image a state of a rectangular array of qubits, for which in the next section we define a dynamical evolution.

3 A quantum lattice system

The quantum system to be considered will be defined by specializing the framework in [2], Section 2: Each of the Hilbert spaces $\mathcal{H}_j$ is simply $\mathbb{C}^2$, the number of particles (here qubits) n is $h \cdot w$ where the image to be associated with a system state is h pixels in height and w pixels wide. Despite their intended rectangular arrangement it is convenient to index the qubits linearly by numbers 1 to n. For any of these indexes it is easy to compute the indexes of the four neighbors in the rectangular lattice. For a boundary point, the missing neighbor is understood to be replaced by the point on the opposite boundary on the same row (if the boundary is a column) or column (if the boundary is a row). According to this rule, each point has four different neighbors. For each lattice corner, two of the neighbors are also corners. Let us denote the four neighbors of i simply as i_1, i_2, i_3, i_4.

One could define neighborship of boundary points differently and thereby change our presently described toroidal topology into that of a simple rectangle (by giving corners only two neighbors and other boundary points three) or into a Möbius strip (by leaving two opposite boundaries without substitutes for neighbors, and defining the neighbors for the other pair of boundaries cross-wise). In images there is normally no important content on the boundary and the treatment of the boundaries makes no large difference. The case selected here is the one which can be coded with the least number of conditional statements.

In the Hilbert space $\mathbb{C}^2$, all Hermitian operators which we will need are the three Pauli matrices $\sigma_1, \sigma_2, \sigma_3$. The Hamiltonian H for each of the isolated qubits we choose

$$H = h_1 \cdot \sigma_1 + h_2 \cdot \sigma_2 + h_3 \cdot \sigma_3 , \quad (h_1, h_2, h_3) \in \mathbb{R}^3 \tag{8}$$

so that, for the selection $h_1 = h_2 = 0$, the conventional *up* and *down* states $(1,0)$ and $(0,1)$ are eigenstates with eigenvalues h_3 and $-h_3$.

The pair interactions V_ρ, $\rho := \{i,j\}$, have to be defined as Hermitian operators on the Hilbert space $\mathcal{H}_i \otimes \mathcal{H}_j = \mathbb{C}^2 \otimes \mathbb{C}^2$, where the two tensorial factors are identical only by the 'accident' that we have identified the isomorphic state spaces of different qubits. Independent of which concrete data structure we choose for this two-qubit space (the obvious choices are $\mathbb{C}$-valued functions on $\{1,2,3,4\}$ and $\mathbb{C}$-valued functions on $\{1,2\} \times \{1,2\}$, only the second choice is canonical) it implies a corresponding tensor product of Hermitian operators on $\mathbb{C}^2$. Understanding this tensor product each of the terms $\sigma_i \otimes \sigma_j$, $i,j \in \{1,2,3\}$, defines an Hermitian operator in $\mathbb{C}^2 \otimes \mathbb{C}^2$. Further, for each $\psi \in \mathbb{C}^2$ and each $\phi \in \mathbb{C}^2 \otimes \mathbb{C}^2$, the element $\langle \psi | \phi \rangle$ is a well-defined element of $\mathbb{C}^2$ and this contraction function $\langle \cdot | \cdot \rangle$ has the property

$$\langle \psi | \psi' \otimes \psi'' \rangle = \langle \psi | \psi' \rangle \cdot \psi'' \tag{9}$$

for all $\psi, \psi', \psi'' \in \mathbb{C}^2$.

The pair interaction we will be using is $V_{\{i,j\}} = 0$ if i and j are not neighbors and is the interaction operator of the Heisenberg models

$$\mathcal{V} := q \cdot (\sigma_1 \otimes \sigma_1 + \sigma_2 \otimes \sigma_2 + \sigma_3 \otimes \sigma_3) \,, \quad q \in \mathbb{R} \tag{10}$$

for neighbors.

The state ψ of the n-qubit system is (in the separated interaction approximation) given as a $\mathbb{C}_1^2$-valued list $(\psi_1, \ldots, \psi_n)$ which depends on time. The dynamics of the system is governed by equation (20) of [2]. The present notation gives this the form

$$\mathrm{i}\,\dot{\psi}_i = H\psi_i + \langle \psi_{i_1} | \mathcal{V} \psi_{i_1} \otimes \psi_i \rangle + \langle \psi_{i_2} | \mathcal{V} \psi_{i_2} \otimes \psi_i \rangle + \langle \psi_{i_3} | \mathcal{V} \psi_{i_3} \otimes \psi_i \rangle + \langle \psi_{i_4} | \mathcal{V} \psi_{i_4} \otimes \psi_i \rangle \tag{11}$$

where the operations $\otimes$ and $\langle \cdot | \cdot \rangle$ are to be understood as explained above. Recall the definition of i_1, i_2, i_3, i_4 at the end of the first paragraph of this section.

Efficient numerical methods are available which allow us to follow timestep by timestep the evolution of arbitrary initial states. The present work is exclusively concerned with initial states which were created from digital color images according to the method of section 2.

With this definition, the conservation properties proved in [2], Section 2 are as follows: For all $i \in \{1, \ldots, n\}$ the scalar product $\langle \psi_i | \psi_i \rangle$ is constant in time and the same is true for the expectation value

$$\begin{aligned} \langle H_{\mathrm{tot}} \rangle = \sum_{i=1}^{n} \Big(& \langle \psi_i | H \psi_i \rangle \\ & + \frac{1}{2} \langle \psi_{i_1} \otimes \psi_i | \mathcal{V} \psi_{i_1} \otimes \psi_i \rangle \\ & + \frac{1}{2} \langle \psi_{i_2} \otimes \psi_i | \mathcal{V} \psi_{i_2} \otimes \psi_i \rangle \\ & + \frac{1}{2} \langle \psi_{i_3} \otimes \psi_i | \mathcal{V} \psi_{i_3} \otimes \psi_i \rangle \\ & + \frac{1}{2} \langle \psi_{i_4} \otimes \psi_i | \mathcal{V} \psi_{i_4} \otimes \psi_i \rangle \Big) \end{aligned} \tag{12}$$

of the total Hamiltonian. Conserved quantities are very useful as indicators for the quality of a computed implementation of the dynamics. Only if the computed discrete system trajectory shows only small deviations of these quantities from constancy the timestep is probably not chosen too large. How these deviations depend on the time step size of the discrete trajectory is a useful indicator of the order of the integration method. An implementation of all this is in `class CpmQM::SpinN` in [5] and [6].

4 Numerical integration

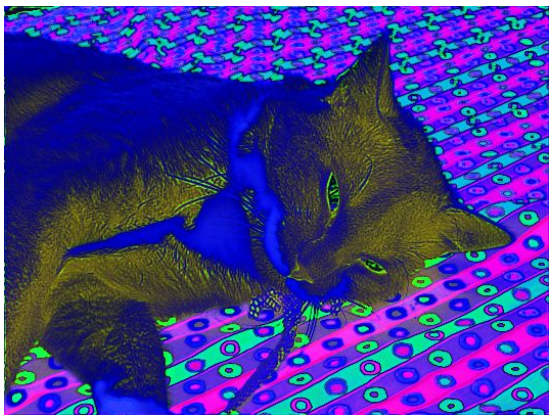
As in [2] we use the asynchronous leap-frog method [3] for a computational solution of the initial value problem. The initial state is obtained from a digital image of my cat Oscar [4]. The original is an image of 2048×1536 pixels, coded as a JPG-file of size 1944 KB. This image was converted to a 533×400 image coded as a PPM-file of size 625 KB. This then was converted into a state of $n = 533 \cdot 400 = 213200$ qubits according to the method of section 2. The Hamiltonian is defined by giving its four parameters the values $h_1 = h_2 = 0$, $h_3 = 1$, and $q = 10$. Then the expectation value of the Hamiltonian with respect to the initial state is $\langle H \rangle = 4.497 \cdot 10^6$. A semi-empirical rule says that $\Delta t = \langle H \rangle / \sqrt{n} = 1.0267 \cdot 10^{-4}$ is a good value for the time step of the densified version of the asynchronous leap-frog integrator; this is the choice which is made for the run which created the following evolved images. It created 8001 images (including the image of the initial state) with 10 (densified) integration steps between images. Computation time for these images was 59 minutes on a state of the art laptop computer (2.39 GHz, 3 GB RAM). What is shown here, is each 50th image from this collection. We see that the integration does not diverge (explode) during these 80000 integration steps. It does so very soon, if we multiply the time step by 150; then the cat is well visible in the first picture which contains divergent pixel values (see Picture 6 for the pictorial effect of divergences as 'black holes'). If we use 4 times the stable step we get divergences only after the cat's image is no longer visible. After the performance of 4000 steps, (and thus after completion of image 401), the sign of Δt is changed so that the evolution — due to the reversibility of the integrator, see [3] — goes all the evolution backward, as can be seen from the paired figures.

Finally a similar image series is shown which starts from a computer generated image [4] which contains large noiseless uniform areas, which typically are missing in camera captures. This series does not include reversal. Instead it shows in an obvious manner the appearance of divergences. These appear since the time step was chosen to be 150 times the recommended value. An implementation of the program run is `CpmVQM2::App18::doTheWork()` for the control path `sel==-2` (see [5],[6]).

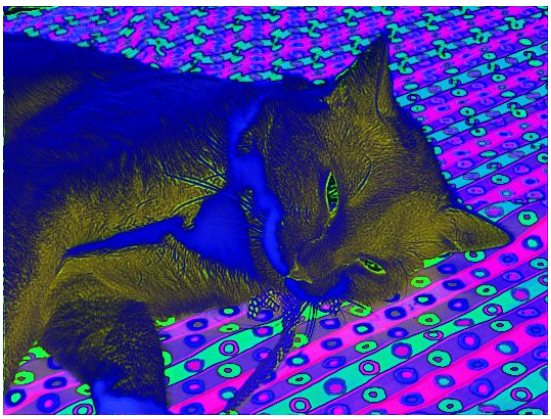
[4] it needs to be a cat in recognition of the heroic role that cats play in quantum theory: Schrödinger's cat and in dynamical systems theory: Arnold's cat transformation

(a) Image 1

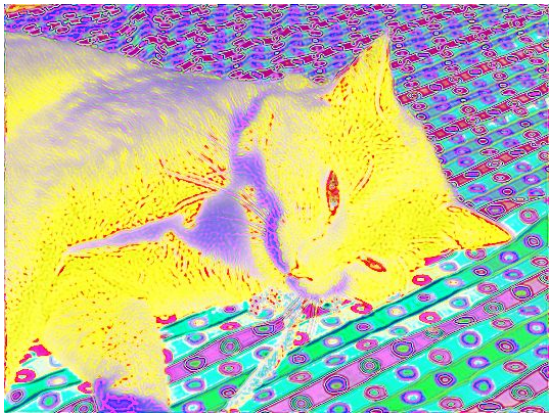
(b) Image 801



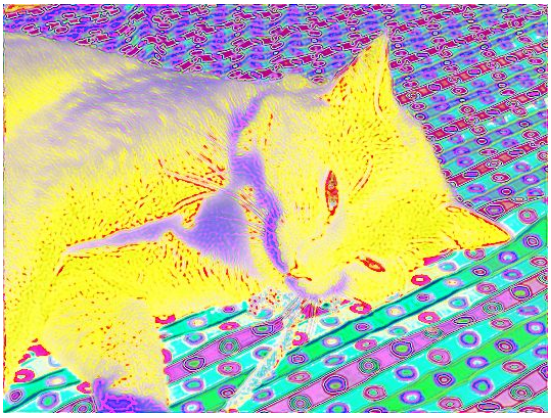
(c) Image 51



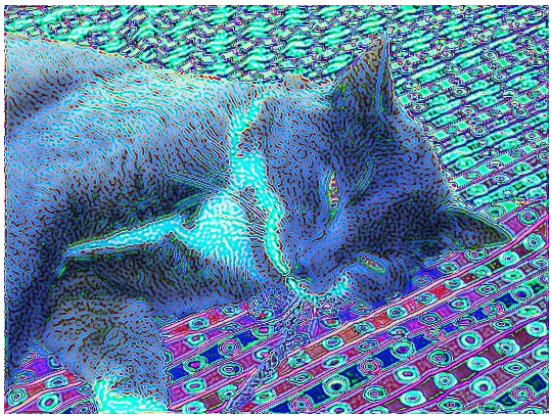
(d) Image 751



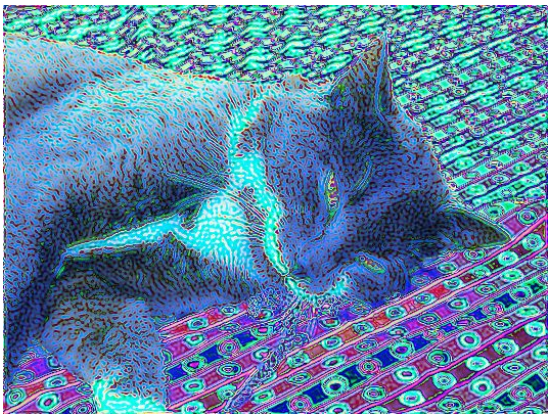
(e) Image 101



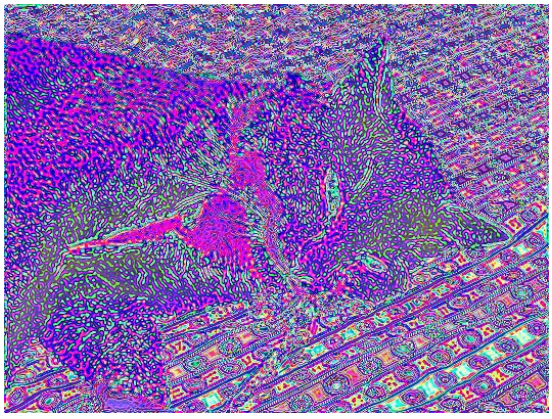
(f) Image 701



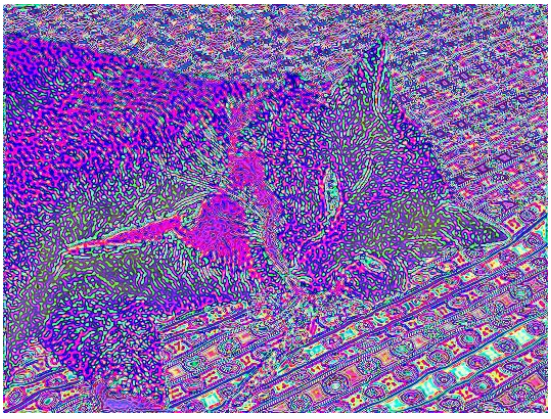
(g) Image 151



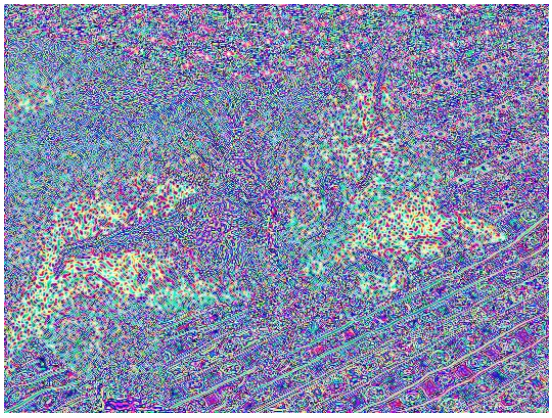
(h) Image 651



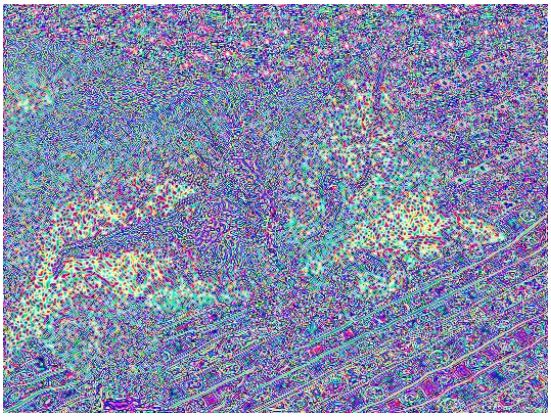
(i) Image 201



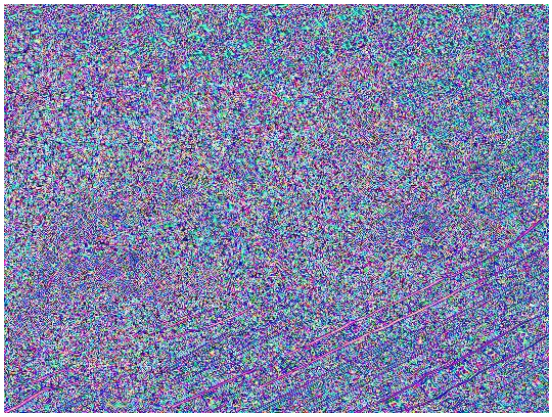
(j) Image 601



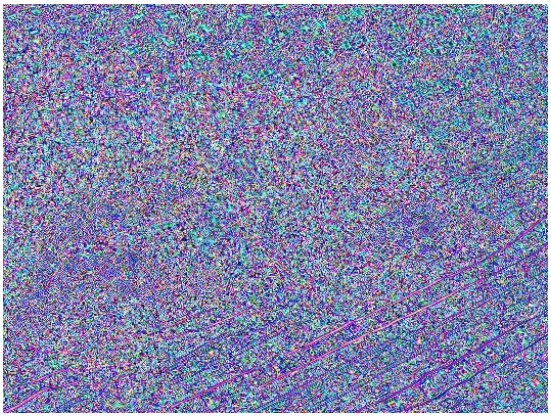
(k) Image 251



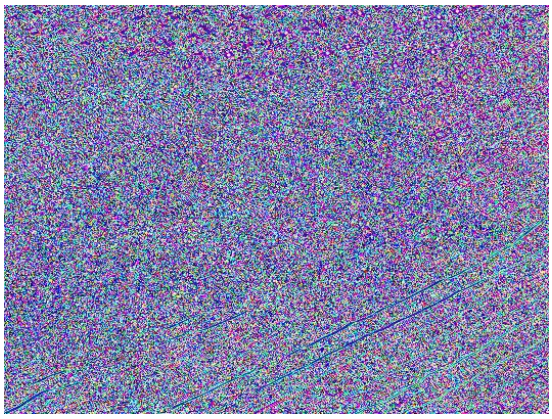
(l) Image 551



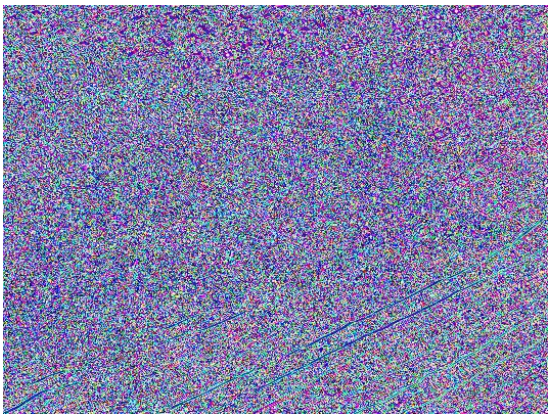
(m) Image 301



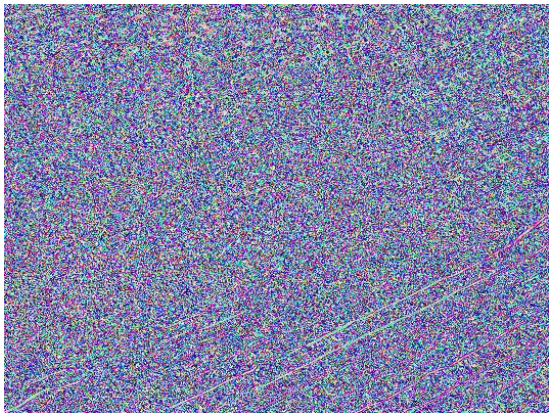
(n) Image 501



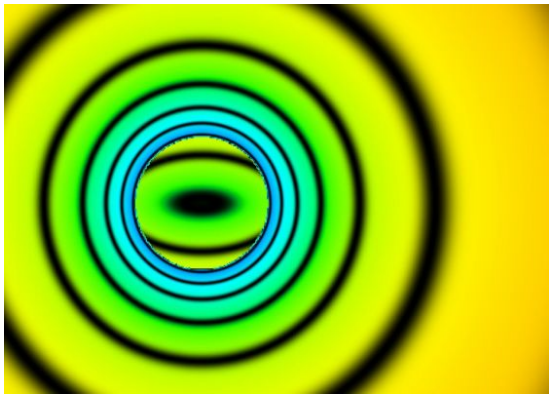
(o) Image 351



(p) Image 451

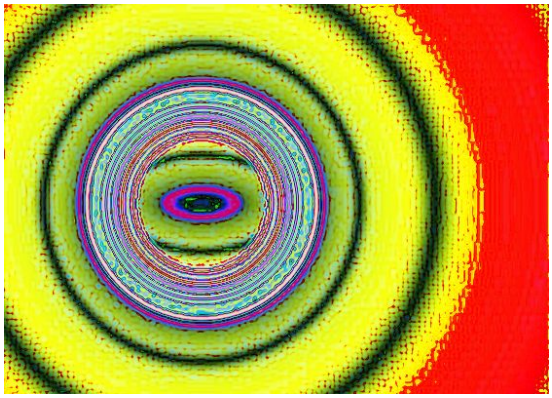


(q) Image 401



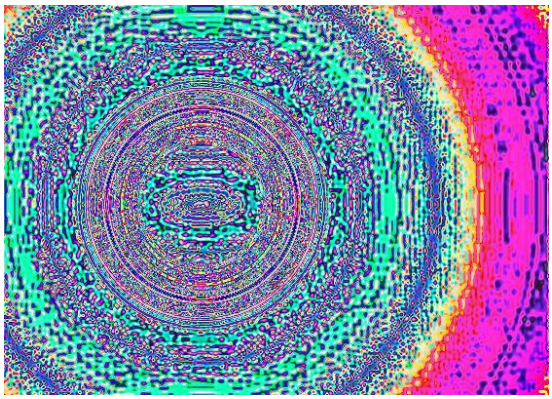
(r) Picture 1

(s) Picture 2

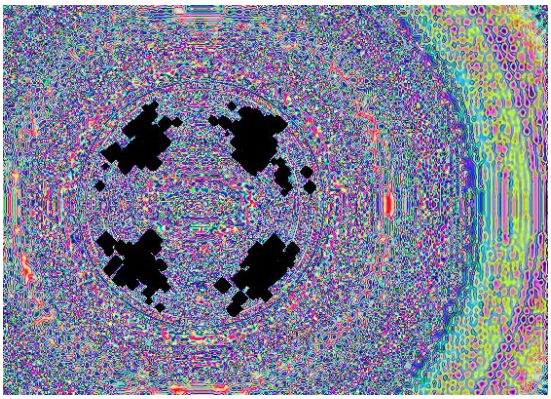


(t) Picture 3

(u) Picture 4



(v) Picture 5



(w) Picture 6

5 Outlook

Applications of the present approach are conceivable in several directions: One may see value in the graphical embellishment of images which the method provides in a very flexible way. One may even enjoy the cryptographic application to evolve completely unrecognizable images into interpretable ones if the parameter h_1, h_2, h_3, q, dt of the evolution method are known (as a key), such as recovering image 801 from image 401. On the other hand, using images as initial values for a large system of evolution equations as (11) allows us to test the robustness of integration schemes very profoundly. The intentional design of initial data will hardly generate such diversified situations as they are provided effortlessly by some images of non-trivial structure. There are many topics on which observations are easily made and which are not studied here: How does the energy expectation value change if we change the pixel count of the image by established resampling methods? How the image content determines the stable time step? How information truncation (such as storing an evolved state as an image on disk, and reconstructing the state from the stored image) affects our ability to evolve the truncated image back to the original?

Acknowledgment

Electronic imaging is a fascinating field and a rich source of inspiration for those who try to understand discretization and algorithms in their relation to systems and processes of the real world. I'm grateful to all those managers of Kodak AG and of Eastman Kodak Company who once involved me in electronic imaging projects and thus provided me with this source of inspiration. These were particularly Wolfgang Ort, Klaus-Dieter Bier, John A. Weldy, Terence S. Lund, and William Hutchinsion. Further, I am grateful to my former research colleagues from whom I learned a lot: Rainer Brändle, Martin Landis, Dieter Horlacher, Thomas Dera, Frank Joos, Javier Lipiz, Cynthia S. Bell, and Kenneth A. Parulski.

References

[1] G. Wyszecki, W.S. Stiles: Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd edition, Wiley 1982

[2] Ulrich Mutze: Separated quantum dynamics
www.ma.utexas.edu/mp_arc/c/08/08-69.pdf (2008)
www.ulrichmutze.de/articles/08-69[1].pdf

[3] Ulrich Mutze: An asynchronous leap-frog method
www.ma.utexas.edu/mp_arc/c/08/08-197.pdf (2008)
www.ulrichmutze.de/articles/leapfrog3.pdf

[4] www.ulrichmutze.de/art.html

[5] www.ulrichmutze.de/softwaredescriptions/cpmlisting.pdf

[6] www.ulrichmutze.de/sourcecode/vqm2.zip

Last modification: 2008-10-24