The Photodynamic Tool: Generation of Animation from a Single Texture Image

Hidetomo Sakaino
NTT Communications Corporation
h.sakaino@ntt.com

ABSTRACT

This paper proposes a new animation editing method for creating animations from a single texture image. A standard warping method can not handle a physical motion effect because of a manual control. The Navier-Stokes equation based method is known to be able to animate a fluid-like animation. However, the difficulties of defining initial- and boundary-conditions with many control parameters remain unsolved for a stable and desired animation. And it offers a heavy computation. In this paper, an advection equation that is a physical-based and compact equation is used. A unique interface is also introduced with a gallery of many velocity vectors: convergence, rotation and vibration. Our method is versatile because that the combination of such velocity patterns can animate a quasi-linear motion over time by solving a set of the linear equations. We have carried out several experiments using some real images. The results of animations show that the PDT is effective, efficient, and easy to use.

1. INTRODUCTION

Recently, a wide variety of an image/animation editting method has been developed [1,2,4,5,6] for both nonprofessionals and animators. For example, changing a scene of a very old photo can help remind of past memories. For *e*-learning or business presentations, use of an animation is way better than a static image.. The conventional tools based on warping method [1], such as After Effects^{TM} or Elastic Reality^{TM} can somehow meet such requirements, however, these are limited to create physical phenomena of a rising smoke.

Current state-of-art computer graphics methods use the Navier Stokes (NS) equation [2,5] when animating fluid like images. Because many parameters need to be set up, the graphical user interface (GUI) has to be well designed [2]. A more visual guidance for selecting the proper icon will be required. Beside the NS based method, the previous methods require the user to control the changes of a motion effect every frame independently by hand. A nonprofessional user will require a plenty of time to decide a complicated motion or will fail to create a natural looking motion.

In this paper, we introduce a new animation generation method from a single image based on the unique interface that provides basic velocity vector patterns We call this interface the photodynamic tool (PDT). Experimental results showed that the PDT is effective and easy to use. In addition, because of its compactness, the PDT can be used as a plug-in for PowerPoint^{TM} and After Effects^{TM} presentations.

2. ANIMATION GENERATION METHOD

2.1 Warping method [1]

A standard warping tool has a simple interface, however, a problem arises when it is applied to an image with a complicated texture such as a scene containing smoke, rippling water, or an oilish painting. Point based correspondence between two consecutive frames limits the ability to generate an image with a fluid-like texture. Because such a texture image exhibits irregular changing patterns of divergence, developing, and decaying. Point correlation of the motions between two frames has rather a one-to-many or a many-to-one relationship.

Defining interactively on segmented lines or curves on an image can not meet with textural changes when a user wants to move part of the image.

After Effects^{TM} tool can give more complicated deformations in an image by selecting from icons. Deformation patterns are not sufficient and velocity vectors should be explicitly shown for a visual guidance. The time layer is helpful to decide motion effects over time. However, if a user wants to create thousands of frames over time, one has to interactively handle all icons by hand. A smooth but unnatural looking animation will be produced This is because a physical model is insufficient for bridging a smooth relationship between two consecutive frames.

2.2 NS equation based method [2,5]

For a fluid-like animation, the NS equation based modeling method [2,5] can be considered. The property of heat conduction phenomena exhibiting a spiral motion can be successfully animated. An interactive menu is presented; however, many parameters such as " temperature" to design a motion will need to be defined.

0-7803-9332-5/05/$20.00 ©2005 IEEE

It is unclear how to know physical interrelations of these parameters. Furthermore, computation load is relatively high. Due to the use of the heat property, it is also hard to generate linear motion. To obtain a desired smoke motion, a key framing method [6] using two frames has been proposed. This approach is out of the scope of this paper.

2.3 Patch based method [3]

Recently, a facial animation has become an active area of research. Simple deformation is easy to create by applying a warping method. For more convincing facial animation, a combination of the parts of a face (eyes, mouth, etc.) extracted from a database can be used [3]. The drawback of this method is that many facial expressions for each person are required. So this method can be used only if the faces of the users are pre-registered.

2.4 Proposed method: PDT

The proposed method focuses the following four parameters: 1) ease of use, 2) efficiency, 3) controllability, and 4) non-linearity. The corresponding solutions we apply are 1) visualization of many velocity vector patterns in a display to help the user imagine how a motion will appear, 2) use of a linear equation as a main equation to reduce the computing load, 3) direct use of velocity to handle each pixel in an image, and 4) allowing the mixing of many complicated velocities to generate quasi-linearity. The following section describes the proposed method in details.

3. PDT SYSTEM

3.1 Projection method

To create physically realistic animation, the projection rule between two consecutive frames is very important. As shown in Fig.1, the PDT uses a new method that allows it to convert neighboring pixels from frame “n” into one pixel at frame “n+1”. In general, this is a representation for physical changes which locally interact but are globally influenced by time and space iteration. A warping method is basically a point-to-point transformation. This kind of the projection limits the ability to exhibit complicated physical phenomena.

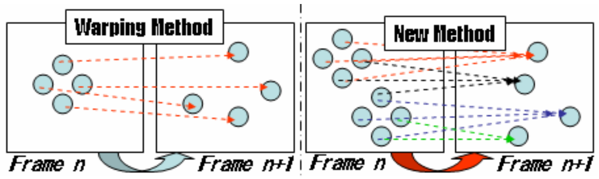


Fig. 1 Difference between two projection methods

3.2 Learning from physical motions

The velocity of a pixel plays an important role in animation. Fig.2 illustrates the relationship between pixel movement and velocity vectors, the main design concept for the PDT. Consider a stick consisting of four pixels in a frame moving from left to right in a display. If all four pixels have the same velocity vector, no deformation of the stick will appear. Should every single pixel has a different velocity vector, the stick image will become increasingly distorted over time.

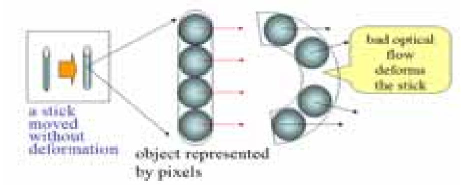

a stick moved without deformation
object represented by pixels
bad optical flow deforms the stick

Fig.2 Relationship between a pixel and a velocity vector.

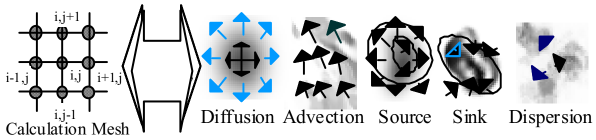


Fig.3 A calculation mesh and some examples of physical phenomena with velocity changes.

Table.1 Basic velocity field patterns

Basic Pattern	Function	Basic Pattern	Function
copy	copy	Vib.	vibration
line	line	smooth	9-points averaging
curve	wave	NS-eq	Fluid flow
power	variation	div.	spreading
vortex	rotation	Conv.	shrink
incline	stepwise	delete	correction

Table 2 Velocity profile for physical phenomena.

Physical Phenomena	Characteristic	Velocity Profile	Variation	Example
bending	pivot	inclination	angle	branch
rotation	angular velocity	vortex	speed	typhoon/hurricane
Div./Conv.	rotA=0, rotA≠0	spread	magnitude	cloud, firework
vibration	sinusoidal	repeat	periodicity	sea,river
translation	shift	line	speed	vehicle

3.3 Design of the velocity vector field

In an animation, there are a wide range of velocity vector patterns. In order to model such patterns, we have applied the fluid dynamics theory [7] and simplified some basic fluid flow patterns into several velocity vector patterns. Fig.3 shows some examples of physical phenomena. Based on these, basic velocity patterns and some options as commands are incorporated into the PDT.

One of the most important commands is a smoothing command which is a kind of low pass filtering applied to a velocity field. This effect is needed to propagate locally given velocity patterns to a broader region when iteratively applied. If the mixture of twin vortexes and lines are given, not only a smoothing but also a quasi-linearity among them can be accomplished. This simulates the function of the NS equation. This property can approximate a wide range of physical movements. But for stronger nonlinearity, if necessary, the NS eq. effect can be applied.

Divergence and convergence commands are useful when animating spreading/dispersing smoke and zooming in/out scene. The divergence effect shows a similar phenomenon of diffusion. A vibration command is used after any velocity patterns are input and then their magnitude and orientation will be changed over time. Line and curve commands can move an object either linearly or nonlinearly. Power command changes the local region of a velocity magnitude when multiple vectors are input. Incline command is useful for moving an articulated object such as a bending finger. Almost all velocity vector patterns can change their orientation and magnitude by handling a slider. From these patterns, applicable objects can be assumed as follows (Table 2); fine branch blowing in the wind, smoke from a locomotive, and a spiral typhoon.

3.4 Editing system overview

As shown in Fig.4(a), in order to create an animation from one image, a user using a mouse drops the image onto a velocity edit palette viewer of the PDT. From a gallery of basic velocity vector patterns (b), the user chooses the patterns as displayed in Table 1. In (a), a divergence flow pattern is overdrawn on a locomotive. Most patterns can control both an orientation and magnitude which examples are shown in (c). When the editing is completed, the user only pushes an edit complete button (a). A system will start to solve an advection equation upto the desired number of frames.

3.5 Numerical scheme

A discretized formula of an advection equation [7] is shown in equation (1). The discretization is made on both space and time. In eq.(1), the **x**, I, and **u** are, respectively, integer pair (i, j) which are a pixel address, an intensity, and a velocity. In this paper, we apply the finite difference method for the first derivatives. Superscript *n* shows a discrete time step and *n+1* means the next frame. The forward time difference is applied to the left hand of above equation. The first deverivate of the space needs image intensities of neighboring pixels. A third order function of an *Advect()* [8] equals $\mathbf{u}_{i,j}^{n}\cdot\nabla I_{i,j}^{n}$ of eq.(1). The most excellent performance of eq. (1) is that each pixel of each velocity vector can be influenced many-to-many and many-to-one as discussed in Fig.1. So, each pixel intensity can shift in the image step by step. This leads to a local change of texture in an image. Global changes can be made due to the integration of the local changes based on eq. (1).

In the initial step *n=0*, one image **I** and velocity vectors **u** are given by the user. An image sequence is synthesized by time integration. Thus, pixel-wise and smooth changes of multiple images will be generated. A computational load is very low because of the linearity.

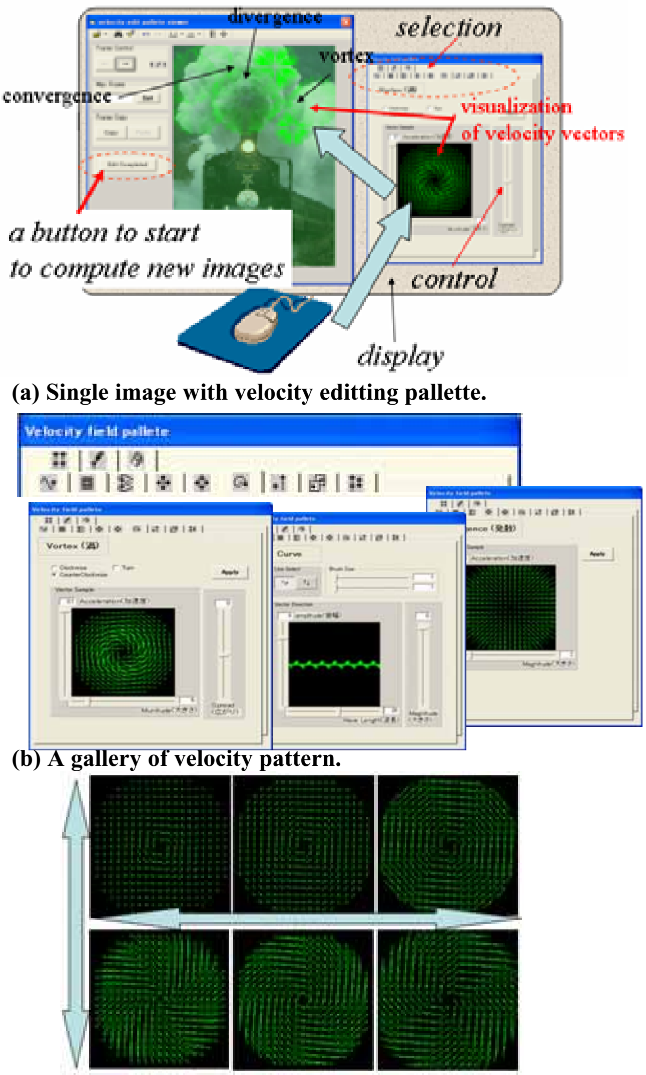
$$I_{i,j}^{n+1}=I_{i,j}^{n}-\mathbf{u}_{i,j}^{n}\cdot\nabla I_{i,j}^{n}\ . \qquad (1)$$

where $\Delta x=\Delta y=1.0, \Delta t=0.25,$

$$Advect(I,u,v,i,j,n)=$$

$$u_{i,j}^{n}\frac{-I_{i+2,j}^{n}+4I_{i+1,j}^{n}-4I_{i-1,j}^{n}+I_{i-2,j}^{n}}{4\Delta x}+\alpha|u_{i,j}^{n}|\frac{I_{i+2,j}^{n}-4I_{i+1,j}^{n}+6I_{i,j}^{n}-4I_{i-1,j}^{n}+I_{i-2,j}^{n}}{12\Delta x}\ .$$

$$+v_{i,j}^{n}\frac{-I_{i,j+2}^{n}+4I_{i,j+1}^{n}-4I_{i,j-1}^{n}+I_{i,j-2}^{n}}{4\Delta y}+\alpha|v_{i,j}^{n}|\frac{I_{i,j+2}^{n}-4I_{i,j+1}^{n}+6I_{i,j}^{n}-4I_{i,j-1}^{n}+I_{i,j-2}^{n}}{12\Delta y}$$



(a) Single image with velocity editting pallette.

(b) A gallery of velocity pattern.

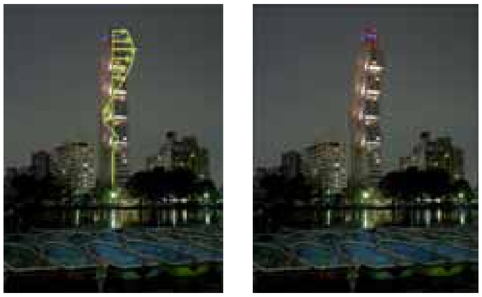
(c) Changeable orientations and density of vortex pattern.

Fig.4 Examples of palettes of basic velocity patterns.

4. EXPERIMENTS

4.1 Building

This experiement shows a scenario where a presenter wants to vibrate a tall building to explain what will happen when an earthquake occurs. To create such an animation, a sine wave is used and its magnitude is varied from zero at the bottom of the building to the largest at the top of it. Finally, a vibration command is applied into the image in Fig.5(a). An animated image (b) shows the smooth and periodical change (5 cycle per frame) caused by the vibration.

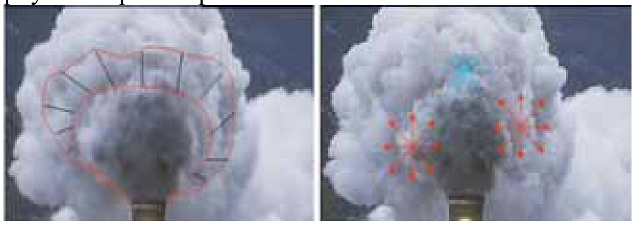


(a) velocity vectors. (b) Vibrating building.

Fig.5 Periodical changes of a building.

4.2 Smoke

A smoking locomotive image is used. A standard warping method (WM) with hand drawn control curves and the PDT with divergence and convergence vector patterns are compared in Fig.6. Note that a few of curves and arrows in Figs.6 (a) and (b) are illustrated by author for presentation, not by a tool. The WM (a) generates spatially a smoother and blurred texture than the original image. In addition to this issue, the WM has another inevitable issue when handling a number of frames. It is needed to handle all frames of control curves by hand. Some regions moves and the other ones remain still. The animation looks unnatural. This is due to a manual control of local motions per frame. Thus the WM is limited to animate a smoke pattern. On the other hand, the PDT (b) generates more physical change, finer texture, and higher contrast than the result of the WM. This is due to using a physical equation per frame.

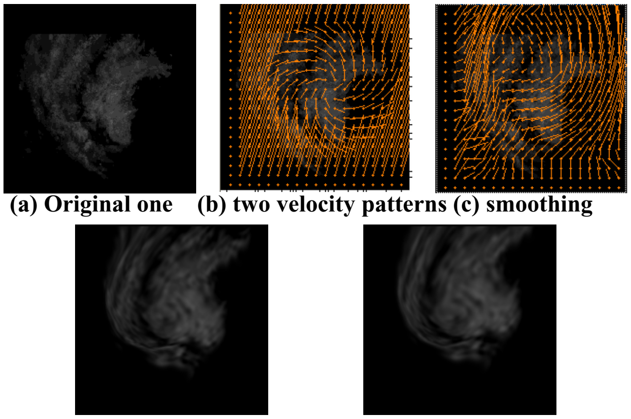


(a) Warping method (b) PDT

Fig.6 Example of a rising smoke animation

4.3 Typhoon

Predicting the course and extent of typhoons is an important topic in meteorology. Fig.7(a) shows one from a typhoon image sequence observed by a radar in Japan over an area of 500x500 km^2. The greyscale corresponds to the amount of a precipitation. To show what a quasi-linear effect of the PDT has, we compare with the NS and without the NS applied to velocity vectors. It is assumed that a presenter aims to move the typhoon upwards and right. To do this, in the first step, inclined lines and one vortex are selected from a menu (b). In the second step, a moving averaging filter is iteratively applied to (b). A quasi-linear velocity pattern (c) is generated, smoothing between lines and the vortex. Finally, a spirally moved typhoon is obtained via eq.(1) (d). The NS eq. is also applied to (b) and eq.(1) is used. (e) is obtained using the NS. Both (d) and (e) look similar. In conclusion, a computation cost has been greatly reduced under a quasi-linear effect.



(a) Original one (b) two velocity patterns (c) smoothing

(d) motion without NS eq. (e) motion with NS eq.

Fig.7 Spiral motion of Typhoon.

5. CONCLUSION AND FUTURE WORK

Animations with plausible motions have been efficiently generated by using the PhotoDynamic Tool (PDT) on a note computer. The PDT is limited to apply to an articulated motion, a scene with occlusion, shadow, and reflection. To enable more realistic animations, we will extend the PDT to include a 3D equation and an image-based rendering application [4] will be performed.

REFERENCES

[1] A warping tool: http://www.debugmode.com/winmorph/.

[2] P. Witting, “Computational fluid dynamics in a traditional animation environment”, Proc. of SIGGRAPH, pp. 129-136, 1999.

[3] E. Cosatto and H. P. Graf, “Sample-based synthesis of photo-realistic talking heads”, Computer Animation, pp. 103-110, 1998.

[4] P. Debevec, C. Taylor, and J. Malik, “Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach”, Proc. of SIGGRAPH, pp. 11-20, 1996.

[5] J. Stam, “Stable fluids”, Proc. of SIGGRAPH, pp. 121-128, 1999.

[6] A. Treuille, A. McNamara, Z. Popovic, and J. Stam, “Keyframe control of smoke simulations”, Proc. of SIGGRAPH, pp. 716-723, 2003.

[7] M. B. Abbott, “Computational fluid dynamics: an introduction for engineers”, Wiley, New York, 1989.

[8] H. Sakaino, “A Unified Prediction Method for Heterogeneous Weather Radar Patterns”, IEEE Workshop on Applications of Computer Vision, pp. 296-303, 2002.