

Stealth Marketing Fluids

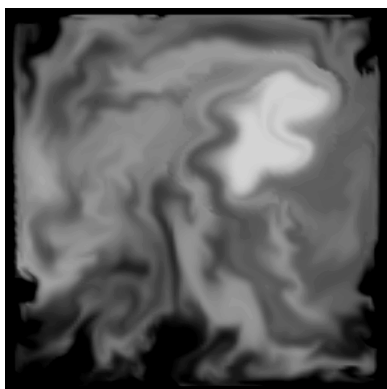
Masato Ishimuroya
freelance

ABSTRACT

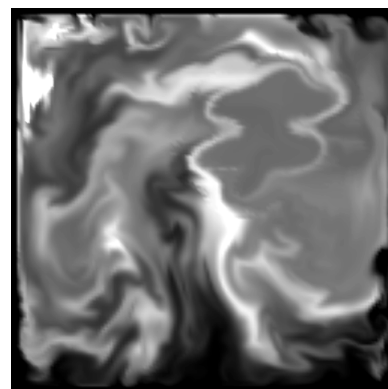
We propose a novel method to express simple objects *casually* in fluid simulations.



(a)



(b)



(c)

Figure 1: Results of proposed method.

(a)Target object to promote. (b)(c)Result frames used proposed method.

Introduction

Computer graphics (CG) plays an important role in video works and games. In recent years, due to the development of algorithms and hardwares, fluid simulation has become approachable and been extensively used in various situations such as movies and video works. We assume that it is possible to publicize overlooked products by inserting hidden messages in these scenes. An example of such products is beloved Kimwipes®. It has great functionality, high availability and lovely case box and name. Also, a sport using Kimwipes was proposed recently. However, in spite of these advantages, not many people know about these wipers. If we can show hidden messages

of Kimwipes in fluid simulation scenes, the product publicity is expected to increase efficiently. The main challenge is that the messages should be inserted *by stealth* - not only the messages shouldn't stand out but also must be independent of the particular scene. In this paper, we propose a method to express objects casually (or by stealth) in fluid scenes.

Related Works

Fluid Simulation

Stam (1999) and Fedkiw et al (2001) introduced efficient and stable methods for performing fluid simulations to the computer graphics community. The book by Bridson (2015) gives an overview of solvers. However, the interaction between users and solvers is provided only via external force term in simulation, thus control of fluid behaviour had been a hard problem initially (Imagine the situation where you have to put suspended dust into a dustbox but only allowed to fan with your hands).

Fluid Control

Key-frame approach (Treuille, 2003) and Target-driven approach (Fattal, 2004) efficiently calculate force field for creating smoke that closely resembles the target shape. Sato et al (2015) uses stream functions for deforming smoke flows so that fluid is kept incompressible. However, these methods solely aim to approximate the smoke shape to the target one, and have different meanings from our method, which implies objects in fluid scenes. In this context, fluid control can be regarded as a special effect for direct advertising. And these methods have drawbacks that solvers may need to recalculate forces from current density of smoke every time step and the results are inapplicable in other scenes.

Texture Synthesis

A texture-synthesis approach converts target shapes to velocity field and embed it in given scenes. It has been extensively researched by others. To name a few, Ma et al (2009) proposed deformation of fluid using motion-field texture, and up-resolution of Wavelet Turbulence (Kim et al, 2008) uses noise texture. The goal of these method is to calculate velocity fields from object's shape and change fluid behaviour by synthesizing them to precomputed scenes.

The concept of our method is similar to the texture-synthesis approach. However, since there is no need to calculate complex formula or interpret detailed user input, our operations are much simpler and easier to implement. Although this paper focuses on 2D simulations in this paper, our method can be applied to 3D simulation in a straightforward manner.

Our Method

Navier-Stokes Equation

We use incompressible Navier-Stokes equations in order to simulate fluid velocity:

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} = \frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{v} + \vec{f} \quad (1)$$

$$\nabla \cdot \vec{v} = 0, \quad (2)$$

where $\vec{v}$ and p are the fluid velocity and the pressure at the position respectively, t is time and ρ and ν are constants. Equation (1) states that the velocity should conserve momentum, and Equation (2) (called divergence-free) implies incompressibility, which is an important feature of fluids.

Overview

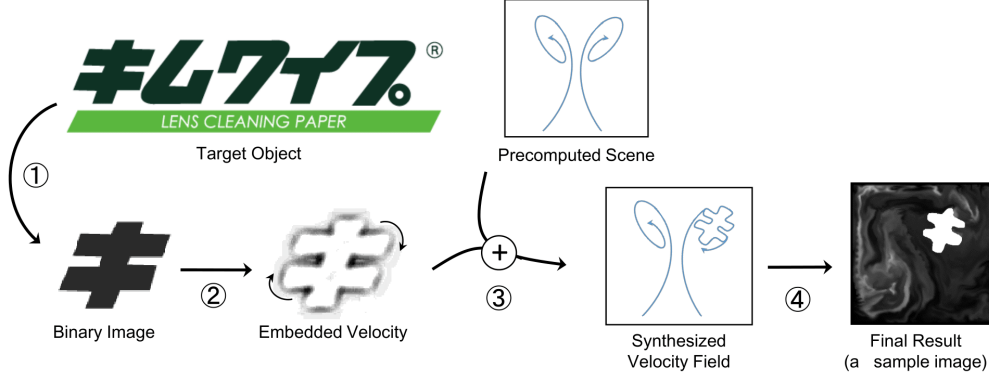


Figure 2: Overview.

Our algorithm consists of the following 4 steps (See Figure 2):

1. Prepare a 2D image which expresses an object and a fluid scene.
2. Generate velocity field from the prepared image.
3. Synthesize the velocity field and the precomputed scene.
4. Advect smoke or particles using the synthesized velocity field and display the result.

The precomputed velocity field referred in Step 1 is calculated using Equation (1) and Equation (2). We recommend using a binary image with gray scale for a 2D image. In following sections, we describe how to execute Step 2 and Step 3 in detail.

Generating Velocity Field

In this section, we propose a method to make velocity field from a single binary image. We denote the scalar function determined by pixel values in the image $\phi(\mathbf{x})$ (Figure 3(a)). Our goal in this step is to obtain velocity field $\mathbf{u}(\mathbf{x})$, where $\mathbf{x}$ is a position in the image. As shown in Bridson et al (2007), it is known that on arbitrary scalar (or 3D vector) field, the result of taking curl operator on it satisfies the divergence-free condition at Equation (2) ($\nabla \cdot (\nabla \times \psi) = 0$). Regarding $\phi(\mathbf{x})$ as scalar potential field, the vector field taken curl

$$\nabla \times \phi = \left(\frac{\partial \phi}{\partial y}, -\frac{\partial \phi}{\partial x} \right) \quad (3)$$

satisfies incompressibility. But there is a subtle problem in differentiation. Since ϕ is a binary function, ϕ is constant (0 or 1) in almost every region except for sharp edge of object, which indicates $\frac{\partial \phi}{\partial y} = \frac{\partial \phi}{\partial x} = 0$ there. Thus, before differentiation, as with the method of Fattal et al

(2003) we blur ϕ by convolution using a Gaussian kernel and obtain $\tilde{\phi}$. As a whole, the embedded velocity field (Figure 3(c)) is generated by

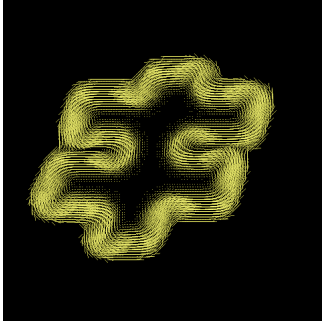
$$\mathbf{u} = \nabla \times \tilde{\phi} = \left(\frac{\partial \tilde{\phi}}{\partial y}, -\frac{\partial \tilde{\phi}}{\partial x} \right). \quad (4)$$



(a) Binary scalar field ϕ .



(b) Blurred (and tone inversed) field $\tilde{\phi}$.



(d) Advection Particles. *[video; see the HTML edition]*

(c) $\mathbf{u} = \nabla \times \tilde{\phi}$.

Figure 3: Results of velocity field generation.

Synthesize A Scene

We denote the embedded velocity field $\mathbf{u}(\mathbf{x})$ and precomputed velocity field in a fluid scene $\mathbf{U}(\mathbf{X})$. Synthesis requires a transformation of $\mathbf{u}$ from local coordinate system $\{\mathbf{x}\}$ to global one $\{\mathbf{X}\}$. According to (Treuille, 2006), with rotation matrix Φ and parallel moving vector t , resampling operator $[R_{\Phi,t}] : \mathbf{x} \mapsto \mathbf{X}$ is given by

$$[R_{\Phi,t}\mathbf{u}](\mathbf{x}) = \Phi\mathbf{u}(\Phi^{-1}\mathbf{x} - t) \quad (5)$$

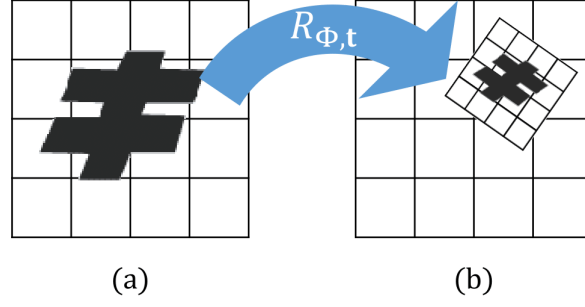


Figure 4. (a)Local coordinate velocity field $\mathbf{u}(\mathbf{x})$. (b)Velocity field $\mathbf{u}(\mathbf{X})$ transformed to global coordinate.

However, since ϕ is originally an image, with image processing software user can easily transform coordinate system beforehand in 3 steps: ①Prepare a blank image $B(\mathbf{X})$ with the same size as the synthesized velocity field. ②Use an image processing software to move ϕ to the desired position in $B(\mathbf{X})$. ③Define it as $\phi(\mathbf{X})$ anew and make $\mathbf{u}(\mathbf{X})$ as mentioned above. This procedure is easy since it need not transform matrices and vectors. Using transformed coordinates $\mathbf{u}(\mathbf{X})$, the synthesis velocity field are given by

$$\mathbf{U}'(\mathbf{X}) = \text{Syn}(\mathbf{U}, \mathbf{u}(\mathbf{X}), \mathbf{X}) = \text{Syn}(\mathbf{U}, R_{\phi,t}\mathbf{u}, \mathbf{X}). \quad (6)$$

Synthesis function Syn is determined by either of following two methods. The first is simply linear combination,

$$\text{Syn}(\mathbf{U}, \mathbf{u}, \mathbf{X}) = \mathbf{U}(\mathbf{X}) + \alpha \mathbf{u}(\mathbf{X}), \quad (7)$$

where α is a constant which is independent of positions. Since $\mathbf{U}, \mathbf{u}$ are both divergence-free, the result of summation in Equation (7) also satisfies divergence-free constraint.

The latter is defined by

$$\text{Syn}(\mathbf{U}, \mathbf{u}, \mathbf{X}) = \begin{cases} \alpha \mathbf{u}(\mathbf{X}) & \text{if } |\alpha \mathbf{u}(\mathbf{X})| \geq |\mathbf{u}(\mathbf{X})| \\ \mathbf{U}(\mathbf{X}) & \text{otherwise.} \end{cases} \quad (8)$$

Though the result using Equation (8) satisfies incompressibility locally, this does not apply to the entire scene. However, embeded velocity $\mathbf{u}(\mathbf{X})$ is not affected by precomputed velocity $\mathbf{U}(\mathbf{X})$.

After synthesis, by advecting smoke density or particles along the velocity field we can visualize the fluid flow.

Results and Discussion

Generate Velocity

Figure 3(c) and 3(d) show the result of the generated velocity field and advected particles along the field respectively. These imply that incompressibility is conserved due to the density of neighbouring particles being persistent. Moreover, the object can be more influential by lengthening the radius of Gaussian-filter.

Synthesize A Scene

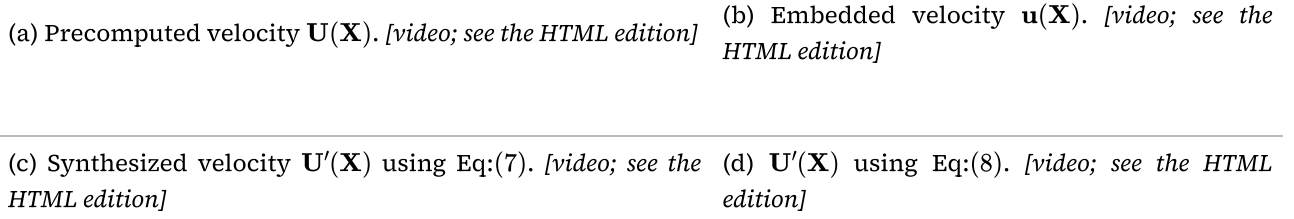


Figure 5: Results of Synthesizing velocity field.

Figure 5(a) shows a precomputed fluid scene $\mathbf{U}(\mathbf{X})$, and Figure 5(b) shows an embedded velocity field $\mathbf{u}(\mathbf{X})$. We used mantaflow (Thuerey & Pfaff, 2016) in precomputing as the solver, and Semi-Lagrange scheme as advection of velocity and smoke density. Figure 5(c) and 5(d) are respectively the results of synthesizing target velocity field and precomputed one by Eq:(7) and (8). It can be seen that the target objects are embedded so that they doesn't change the global shape of smoke. Figure 1(b) and 1(c) are the snapshots of the movie Figure 5(c) and 5(d). We can see the target objects clearly in a single frame, but when watching movie, the target doesn't stand out too much and our purpose is achieved.

Limitation

Synthesising velocity field can be useless when the direction and the magnitude of the embedded velocity is not carefully calculated. Figure 6(a) shows the synthesis of $\mathbf{U}$ and the velocity field with the opposite direction of Figure 5(b). It may be necessary to check the synthesised result and adjust the parameters accordingly. As mentioned above, synthesised velocity field from Eq:(8) can adversely affect the result since it does not satisfy incompressibility. Figure 6(b) shows the advected particle with the velocity field of Figure 5(d). One can observe that the density of particle is different from other areas.



Figure 6: Results of Synthesize of velocity field.

Conclusion

In this paper we proposed a new method on creating incompressible velocity field from a grey scale image and synthesising it with a fluid scene. Applying our method to 3D simulation, subjective evaluation using a questionnaire and adjusting a noticeable degree of the object without depending on the scene are left for future works.

Acknowledgement

The author would like to thank Seiichi Uchida for the English proofreading of this paper.

Reference

1. Bridson, R. (2015). *Fluid Simulation for Computer Graphics Second Edition*. AK Peters/CRC Press.
2. Bridson, R., Houriham, J., & Nordenstam, M. (2007). Curl-noise for procedural fluid flow. In *ACM SIGGRAPH 2007 Papers*.
3. Fattal, R., & Lischinski, D. (2004). Target-driven smoke animation. *ACM Transaction on Graphics*, 23(3), 441–448.
4. Fedkiw, R., Stam, J., & Jensen, H. W. (2001). Visual simulation of smoke. In *Proceedings of the 28th annual conference on computer graphics and interactive techniques*, 15–22.
5. Kim, T., Thürey, N., James, D., & Gross, M. (2008). Wavelet turbulence for fluid simulation. In *ACM SIGGRAPH 2008 Papers*.
6. Ma, C., Wei, L.-Y., Guo, B., & Zhou, K. (2009). Motion field texture synthesis. In *ACM SIGGRAPH Asia 2009 Papers*.
7. Sato, S., Dobashi, Y., Yue, Y., Iwasaki, K., & Nishita, T. (2015). Incompressibility-preserving deformation for fluid flows using vector potentials. *The Visual Computer*, 31(6), 959–965.
8. Stam, J. (1999). Stable fluids. In *Proceedings of the 26th annual conference on computer graphics and interactive techniques*, 121–128.
9. Thuerey, N., & Pfaff, T. (2016). MantaFlow. <http://mantaflow.com>.
10. Treuille, A., Lewis, A., & Popovic, Z. (2006). Model reduction for real-time fluids. In *ACM SIGGRAPH 2006 Papers*, 826–834.
11. Treuille, A., McNamara, A., Popovic, Z., & Stam, J. (2003). Keyframe control of smoke simulations. *ACM Transaction on Graphics*, 22(3), 716–723.

Ishimuroya, M. (2017). Stealth Marketing Fluids. *Scientific Sports*, 2(1), 1–8.

© 2017 Masato Ishimuroya. Licensed CC BY 4.0. Published by the International Kimwipe Table Tennis Association. <https://journal.iktta.org/articles/2/1/2-1-1-abstract.html>