Bulletin of the JSME Vol.8, No.2, 2014
Journal of Advanced Mechanical Design, Systems, and Manufacturing

Velocity calculation of 2D geometric objects by use of surface interpolation in 3D

Yojiro MANDACHI*, Shin USUKI** and Kenjiro T. MIURA*
*Graduate School of Engineering, Shizuoka University
3.5.1 Johoku, Kitaku, Hamamatsu, Shizuoka, Japan
E-mail: tmkmiur@ipc.shizuoka.ac.jp
**Research Institute of Electronics, Shizuoka University
3.5.1 Johoku, Kitaku, Hamamatsu, Shizuoka, Japan

Received 23 October 2013

Abstract
In some engineering applications, it is desirable to calculate the local velocity of geometric objects or boundaries of physical phenomena to understand the deformation speed of the objects or the mechanisms of the physical phenomena. Furthermore, knowledge of the velocity distribution will help designers create better mechanical systems. To determine the velocity of the boundary of a 2D object or 2D phenomenon, we first construct an implicit surface taking the time component into consideration by the radial basis function, then we calculate the local velocities of the boundary using the generated implicit surface. We apply our method to calculate flame velocity in a combustion chamber.

Key words : Velocity calculation, 2D object, Implicit surface, Radial basis function

1. Introduction

In recent years, the development of image acquisition techniques has enabled researchers to obtain images such as flames in combustion chambers, material transported in human cells, etc. It is frequently necessary to calculate the velocity of the 2D objects in these images. One often use the optical flow method to estimate the motion of the object in a sequence of images. But it is difficult to match points between images, especially flame images.

Our approach to this problem is to use 3D implicit surface taking the time component into consideration. This was presented previously as a method of morphing (Turk and O'Brien, 1999). However, it has been described that the velocity of a time-dependent implicit surface cannot be defined unambiguously (Stam and Schmidt, 2011). In this paper, we propose a method to calculate the velocity of an implicit surface by splitting it into global translational components, global rotational components and local deformational components.

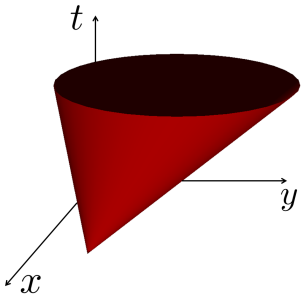

Fig. 1 3D surface

Paper No.13-00263
[DOI: 10.1299/jamdsm.2014jamdsm0012]

2. Implicit Interpolation by Radial Basis Function

We represent a time-variable 2D shape as a 3D surface. For example, Fig. 1 depicts a translating and expanding circle. This surface is generated by interpolating contour points in the consecutive images of the circles. We use the radial basis function for the interpolation as reported previously (Carr, 2001).

2.1. Radial Basis Function

The radial basis function is the monotone function of the distance from the origin. Among several types of function, we use the triharmonic spline suitable to interpolate a function of three variables, defined by

$$\phi(r) = r^3 \tag{1}$$

2.2. Interpolation of Points

Given a set of points $\{\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_k\} \subset \mathbb{R}^3$ and a set of function values $\{h_1, h_2, ..., h_k\} \subset \mathbb{R}$, we find a function $f : \mathbb{R}^3 \to \mathbb{R}$ *such that*

$$f(\boldsymbol{c}_i) = h_i \ \ (i = 1, 2, ..., k) \tag{2}$$

For Example, when one assign points on the surface with zero values, $f = 0$ is the expected surface.

Although there are an infinite number of functions that satisfy Eq. (2), typically we consider the function that minimizes the energy E defined by

$$E = \int_{\mathbb{R}^3} \left\{ f_{xxx}^2(\boldsymbol{x}) + f_{xxy}^2(\boldsymbol{x}) + f_{xxz}^2(\boldsymbol{x}) + \ldots + f_{zzz}^2(\boldsymbol{x}) \right\} d\boldsymbol{x} \tag{3}$$

where, for example, f_{xxx} represents the third-order partial derivative of f with respect to x.

We can solve this problem using weighted sums of the radial basis function as

$$f(\boldsymbol{x}) = \sum_{j=1}^{k} d_j \phi(|\boldsymbol{x} - \boldsymbol{c}_j|) + P(\boldsymbol{x}) \tag{4}$$

where $\boldsymbol{x}$ is an arbitrary point in a 3D space $\{x, y, z\}$, d_j is a weight and $P(\boldsymbol{x})$ is a quadratic polynomial defined by

$$P(\boldsymbol{x}) = p_0 + p_1 x + p_2 y + \ldots + p_9 zx \tag{5}$$

Then, Eq. (2) can be rewritten as

$$\sum_{j=1}^{k} d_j \phi(|\boldsymbol{c}_i - \boldsymbol{c}_j|) + P(\boldsymbol{c}_i) = h_i \ \ (i = 1, 2, ..., k) \tag{6}$$

Furthermore, to minimize Eq. (3), Eq. (7) must be satisfied.

$$\sum_{i=1}^{k} d_i = \sum_{i=1}^{k} d_i x_i = \sum_{i=1}^{k} d_i y_i = \ldots = \sum_{i=1}^{k} d_i z_i x_i = 0 \tag{7}$$

Coefficients d_j and p_j, which satisfy Eq. (6) and Eq. (7), are calculated by solving the following system of equations.

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \ldots & \phi_{1k} & 1 & c_1^x & c_1^y & \ldots & c_1^z c_1^x \\ \phi_{21} & \phi_{22} & \ldots & \phi_{2k} & 1 & c_2^x & c_2^y & \ldots & c_2^z c_2^x \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{k1} & \phi_{k2} & \ldots & \phi_{kk} & 1 & c_k^x & c_k^y & \ldots & c_k^z c_k^x \\ 1 & 1 & \ldots & 1 & 0 & 0 & 0 & \ldots & 0 \\ c_1^x & c_2^x & \ldots & c_k^x & 0 & 0 & 0 & \ldots & 0 \\ c_1^y & c_2^y & \ldots & c_k^y & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1^z c_1^x & c_2^z c_2^x & \ldots & c_k^z c_k^x & 0 & 0 & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \\ p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_9 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{8}$$

where $\phi_{ij} = \phi(|\boldsymbol{c}_i - \boldsymbol{c}_j|)$ and $\boldsymbol{c}_j = (c_j^x, c_j^y, c_j^z)$

[DOI: 10.1299/jamdsm.2014jamdsm0012]

3. Extraction of Points from Consecutive Images

In this section, we describe a method to extract 3D points from consecutive images.

One can assign a corresponding time t to each image. A point extracted in the image corresponding to a certain time $t = T$ has a 2D coordinate (X, Y) as shown in Fig. 2 left. Adding a time dimension to this coordinate, the point can be treated as 3D coordinates (X, Y, T) as shown in Fig. 2 right.

Fig. 2 Extraction of points

However, with extraction of only contour points, the right-hand values of Eq. (8) are all zero, and therefore we only derive a trivial solution. Then, we provide points translated along a normal direction outside (positive value) and inside (negative value) as shown in Fig. 3. The normal vector is approximately calculated using neighboring contour points, and the amount of offsets is the distance corresponding to 1 pixel in the image in this paper.

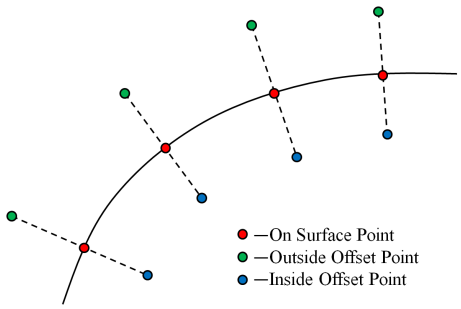

Fig. 3 Offset points

4. Method of Velocity Calculation

It is assumed that the velocity of the 2D object in the consecutive images has three components. the first is the global translational component, the second is the global rotational component, and the third is the local deformational component. We assume that the first corresponds to the translation of the center of gravity of the object, the second corresponds to the rotation of the principal axis of inertia of the object, and the third is oriented in the normal direction of the object's contour. In this section, we describe a method to calculate the velocity.

4.1. Translational Component

The location of the center of gravity of the object can be calculated in each image. Then, we interpolate the center of gravity coordinate $\boldsymbol{G}(g_x, g_y)$, and generate functions $g_x(t)$ and $g_y(t)$ respectively. The translational component of the velocity $\boldsymbol{V}_t$ is derived as a time derivative of the functions, i.e.,

$$\boldsymbol{V}_t = \left(\frac{dg_x(t)}{dt}, \frac{dg_y(t)}{dt}\right) \tag{9}$$

We use the spline curve for the interpolation. The spline function is a set of piecewise polynomial functions. Using this, we interpolate points as shown in Fig. 4.

[DOI: 10.1299/jamdsm.2014jamdsm0012]

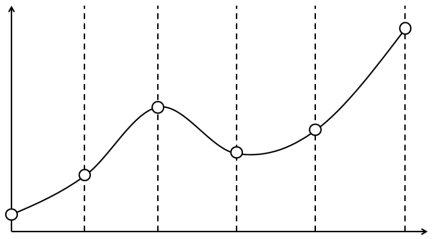

Fig. 4 Spline function

The polynomial is often third-order, and its first and second derivatives are continuous at the boundary between segments. Thus, the interpolation function $x(t)$ and its derivative are defined as

$$x(t) = at^3 + bt^2 + ct + d \tag{10}$$

$$\frac{dx(t)}{dt} = 3at^2 + 2bt + c \tag{11}$$

4.2. Rotational Component

The angle of the principal axis of inertia of the object can be also calculated in each image. As is the case in the translational component, we interpolate the angle θ, and generate a spline function $\theta(t)$. As shown in Fig. 5 ,the rotational component of the velocity $\boldsymbol{V}_r$ on a point $\boldsymbol{P}(p_x, p_y)$ is derived as

$$\boldsymbol{V}_r = ((g_y - p_y)\frac{d\theta(t)}{dt}, (p_x - g_x)\frac{d\theta(t)}{dt}) \tag{12}$$

The principal axis might be unstably calculated for nearly circular shapes.

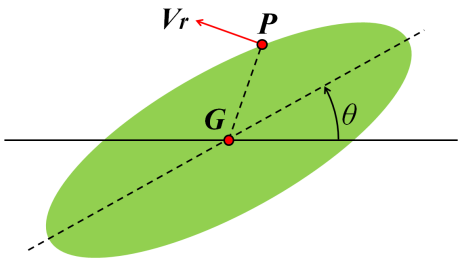

Fig. 5 Rotational Component

4.3. Deformational Component

The translational component and the rotational component described in Subsections 4.1 and 4.2 are rigid motional components. In order to take into account only the non-rigid deformation, we subtract these rigid motions from the original points' coordinates. We generate an implicit surface $f(x, y, t) = 0$ that interpolates these points, and the deformational component of the velocity $\boldsymbol{V}_d$ is

$$\boldsymbol{V}_d = \left(\frac{dx}{dt}, \frac{dy}{dt}\right) \tag{13}$$

Hereinafter, we derive the velocity $\boldsymbol{V}_d$.

Total differential of f with respect to time t is

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} + \frac{\partial f}{\partial t} \tag{14}$$

On the surface, this value is equal to zero, i.e.,

$$\frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} + \frac{\partial f}{\partial t} = 0 \tag{15}$$

[DOI: 10.1299/jamdsm.2014jamdsm0012]

And, we define the deformational component as a normal component of the velocity of the surface, i.e.,

$$\boldsymbol{V}_d = c\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) \tag{16}$$

From Eq. (15) and Eq. (16), we can derive the deformational component as

$$\boldsymbol{V}_d = \left(\frac{-\frac{\partial f}{\partial t}\frac{\partial f}{\partial x}}{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}, \frac{-\frac{\partial f}{\partial t}\frac{\partial f}{\partial y}}{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}\right) \tag{17}$$

4.4. Results for Sample Data

As examples, we show the results of velocities of some sample objects.

First, we calculate the velocities of a translating and expanding circle. Fig. 6 top row shows the results obtained using our method, while the bottom row shows the results where only the normal component of the velocity is taken into consideration. In the figure, the gray region, the black lines, the red lines, the green lines and the blue lines represents the object ,the total velocities, the normal velocities, the translational velocities and the rotational velocities respectively. Our method can be used to obtain a natural result with the provided motion.

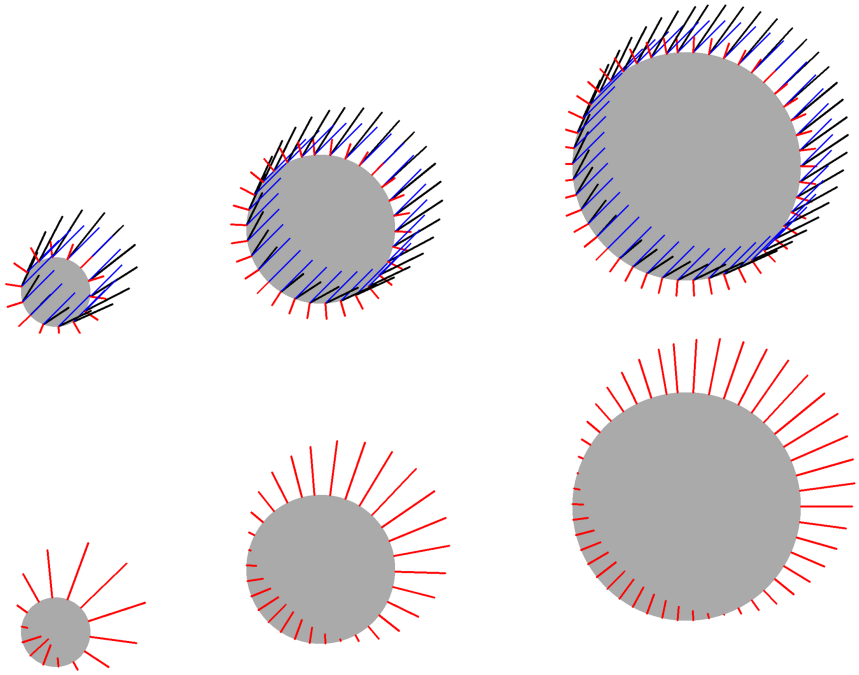

Fig. 6 The sequence of a translating and expanding circle and calculated velocities. Top:our method's velocities.Bottom:simple normal velocities.

Next, we calculate the velocities of a rotating ellipse as shown in Fig. 7. Our method also can obtain a natural result with rotational motion.

[DOI: 10.1299/jamdsm.2014jamdsm0012]

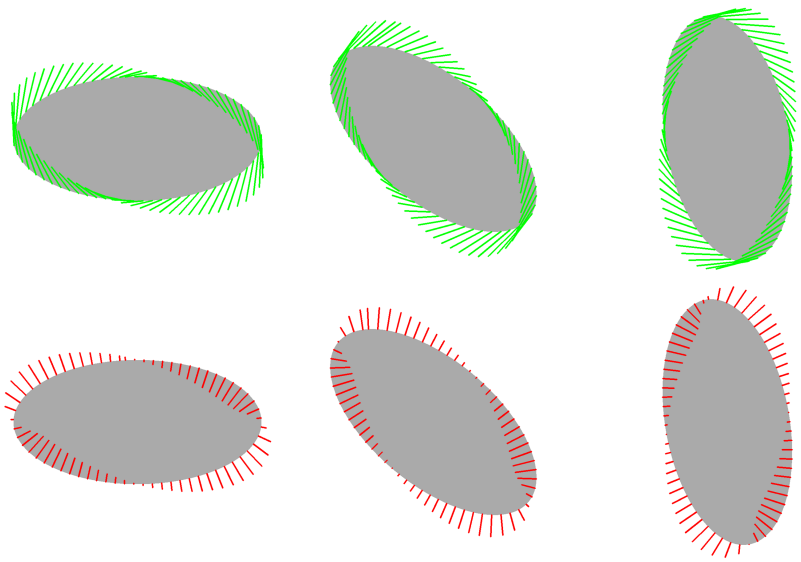

Fig. 7 The sequence of a rotating ellipse and calculated velocities. Top:our method's velocities.Bottom:simple normal velocities.

5. Results of Flame Velocity Calculation

We applied our method to calculate the flame velocity in a combustion chamber. In this section, we describe the experimental method and its results.

We used the experimental instrument described previously (Yumoto, 2011), as shown in Fig. 8. The combustion chamber is observed through an air-cooled borescope (SMETec). It has a 70-degree view angle and 70-degree view direction. Images are captured by a monochrome camera with C-MOS image sensor (Photron: FASTCAM-MAX). An image intensifier (Hamamatsu Photonics: C9548-03) is used to intensify weak emission of the initial flame.

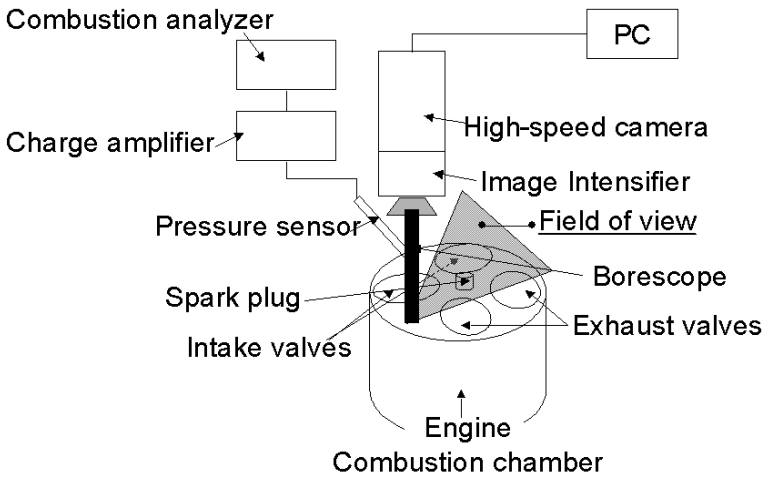

Fig. 8 Experimental instrument

Using this instrument, we acquired 25 consecutive images of a propagating flame. Fig. 9 depicts images in 7th, 14th, and 21st frames. The image resolution is 256×256 pixels and the frame rate is 8000 fps.

We smoothed the images by threshold processing and Gaussian filter, and extracted the edges with a Laplacian filter. After binarization, the images were processed as shown in Fig. 10.

We extracted contour points in these images, and generated an implicit surface that interpolates them. Fig. 11 depicts the surface shown from several view-points. The number of control points for RBF interpolation was 1816 (908 contour points, 454 outside offset points, and 454 inside offset points). The object's shape at an arbitrary time can be determined

[DOI: 10.1299/jamdsm.2014jamdsm0012]

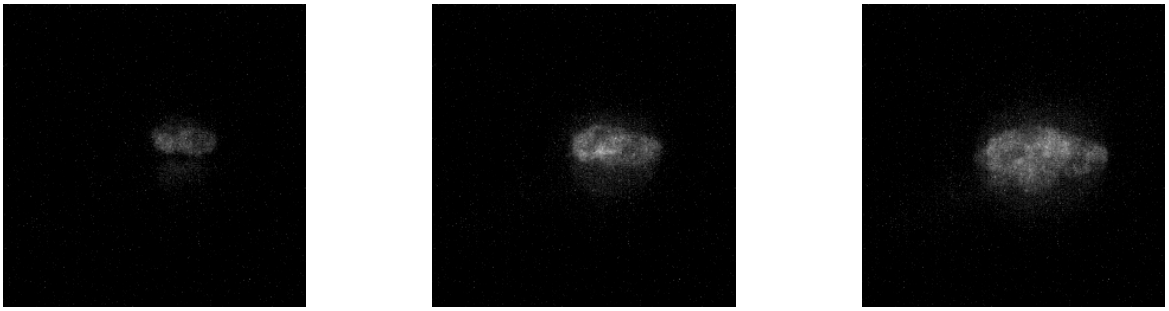

Fig. 9 Flame images

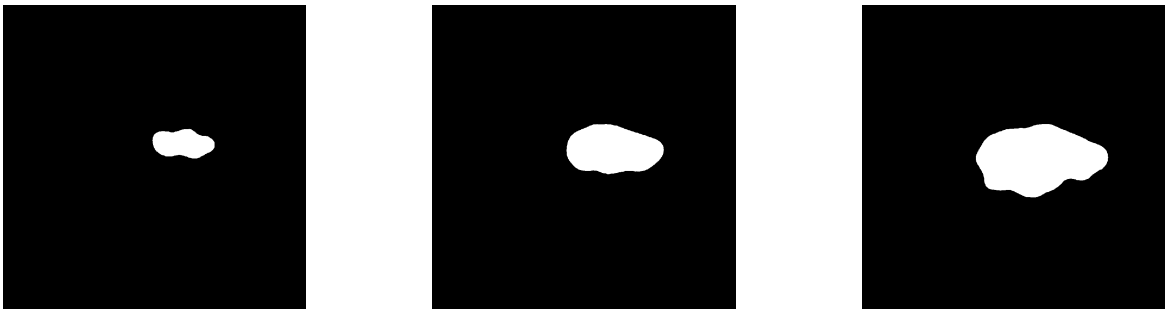

Fig. 10 Flame images after processing

by slicing this surface perpendicularly to the t-axis.

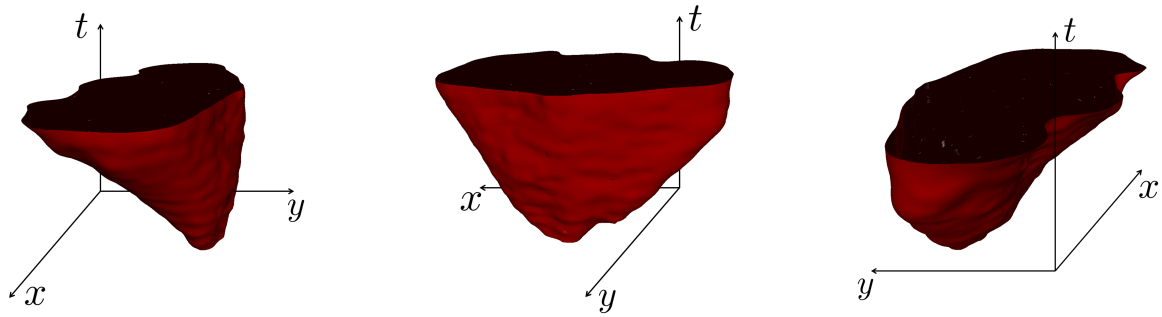

Fig. 11 Generated implicit surface

Using this surface, we calculated the velocities of the flame as shown in Fig. 12. From this result, the flame translates globally and expands locally. For the flame case, the rotational component is thought to be smaller than other components. Thus we conclude that this result is valid. The computational time is shown in Table 1. We speeded up solving Eq. 8 by parallel processing on the GPU.

[DOI: 10.1299/jamdsm.2014jamdsm0012]

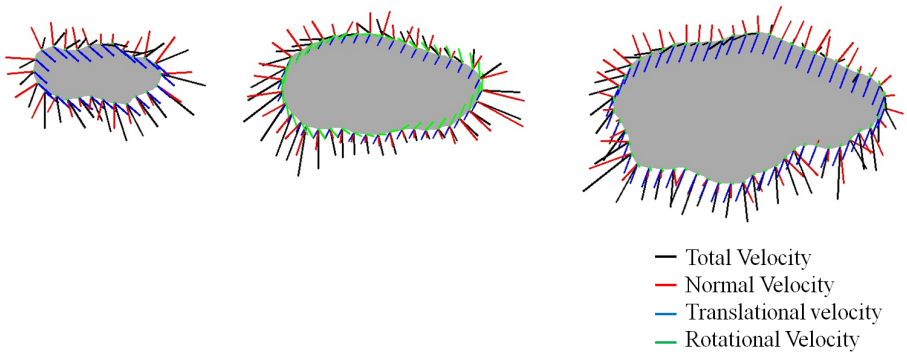

Fig. 12 Velocities of a flame

Table 1 Computational time. Computations were performed on a 3.4 GHz Core i7-2600 with 8 MB RAM and GeForce GTX 560Ti.

	Computational time [s]
Generating an implicit surface	0.31
Calculating velocities per frame	0.25

6. Conclusion

In this paper, we have presented a method to calculate a 2D object in consecutive images. We performed calculations by splitting into the global rigid motional component and the local deformational component. The results of sample simulations confirmed that our method can be applied to translational motion, rotational motion and scaling deformation of an object. Using our method, we could calculate valid velocities of a flame in consecutive images.

In future work, we plan to apply our method to deformation other than scaling, taking into account the tangential component of the deformational velocity. Furthermore, we will extend our method for 3D objects.

References

Carr, J. et al., Reconstruction and representation of 3d objects with radial basis functions , Computer Graphics, Vol. 35, No. 3(2001), pp. 67-76.

Stam, J. and Schmidt, R., On the velocity of an implicit surface , ACM Transactions on Graphics, Vol. 30, No. 3(2011), pp. 21:1-21:7.

Turk, G. and O'Brien, J., Shape transformation using variational implicit functions , Computer Graphics, Vol. 33, No.3(1999), pp. 335-342.

Yumoto, M. et al., Influence of injection and flame propagation on combustion in motorcycle engine - Investigation by visualization technique - , SAE2011-32-0566 / JSAE20119566(2011).

[DOI: 10.1299/jamdsm.2014jamdsm0012]