International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.9 (2015), pp.373-382
http://dx.doi.org/10.14257/ijmue.2015.10.9.38

An Optical Flow using SVD for Video based Fluid Animation

HaengA An and Jinho Park

Global School of Media, Soonsil University, Korea, Global School of Media, Soonsil University, Korea
gid0421@gmail.com, c2alpha@ssu.ac.kr

Abstract

This paper proposes a novel optical flow method using SVD(Singular Value Decomposition) for extracting flow velocity field included in an input video. SVD calculates principal components of the image sequence, and the intermediate frame between sequential images is approximated. We generate optical flow for fluid movements using a groundbreaking optical flow estimation method which is the combination of Lucas/Kanade and Horn/Schunck to achieve the best of the local and global differential methods. While fluid simulation plays central role of generating fluid animations, our video based flow field generation scheme shows the potential for novel paradigm on physics based animation.

Keywords: *Fluid Animation, Optical Flow, Image Processing, Advection, Flow Visualization*

1. Introduction

This paper proposes fluid animation method which analyzes flow field from fluid video. The input video is single viewpoint video without illumination change, supposes there is no camera motion and considers complex turbulence as also input range. Existing fluid animation production mostly relies on fluid simulation, on the other hand, this paper propose effective process which calculates 2d velocity field of smoke motion in video without fluid simulation. Regarding particle, streakline and styled objects as rendering primitives, our method generates fluid animation which follows smoke motion of video.

Although flow field has made remarkable process in graphics field over the past decade, it also has limits of the way which rely on simulation. A method of making fluid animation is much more solid and can produce videos like actual video by output in fluid mechanics pass down the graphics field. So this is applied in various contents industry such as movie, advertisement, game and other industries which include fluid.

There are various methods to solve the Navier-Stokes equation which is fluid equation to compatible with fluid, applied field. Methodology based on grid lead beginning of the fluid animation method and methodology based on particle is come to the fore in order to describe detail part. These days methodology based on hybrid which takes advantages from both sides starts to developed. However, although diverse attempt is suggested to shorten calculation time due to the improvement of algorithms and improve the quality, it requires enormous calculation time for the same level of actual image and also entailing boring parameter tuning to get desired result is pointed out limits of the method.

There are many heated studies that mimic objects from real one in various areas but these tries are not brisk in fluid area. It is hard to get the same quality of actual image only using handworks or theoretical calculation. Character animation field has studied capturing the actor’s motion in the past twenty years ago and now the method which combines with physically simulating skeletons is ongoing. Face animation field also conducted aimed to peoples face movement such as emotion capture using in movie called avatar. Actual image capture study in fluid field is at an early stage because of the characteristic that the movement of fluid is complex, hard to find characteristic point and cannot sustain status. Laser sweeping captures the movement of gas through the distorted of noise image or captures the surface of fluid combine with fluid simulation method. These studies consider capturing three-dimensional flow field based on vision technology as target. There is a study which modeling surface of water from single video using Shape from shading but few studies exist make fluid animation using single smoke animation.

To sum up our study, we enhance the efficiency of optical flow making middle frames like taken by super high speed camera because fluid does not have feature.

2. Related Works

For recent ten years, fluid animation in computer graphics has achieved impressive improvements to satisfy the needs of visual effects. Stam [1] builds the foundation of numerically stable fluid simulation methods by introducing the fractional step method to computer graphics. On top of Stam's method, Foster *et al*. [2] adopts level set method for tracking liquid boundary. These methods analyze fluid movement in grid based Eulerian view rather than classical mechanics view which focuses on the movement of individual objects. Grid based fluid simulation methods require large simulation resolution to capture sufficient details and large simulation resolution causes heavy computation load proportional to the domain resolution. Smoothed Particle Hydrodynamics(SPH) based methods such as [3] catches popularity in computer graphics field since the methods consider fluid as individual particle and are easy to describe the details of fluid such as water drop and vapor. The incompressible constraint for fluid mass is coupled to common SPH to overcome drawback of SPH that hard to reconstruct smooth fluid surface only by particles [4-5]. Recently, Raveendran *et al*. [6] develops a hybrid approach to constitute the best of grid and particle based fluid simulation methods.

Fluid mechanics and image processing have developed many fluid movement capture methods prior to computer graphics. In experimental fluid mechanics, Particle image velocimetry (PIV) [7] obtains time varying velocity fields by measuring the displacements of numerous fine particles that accurately follow the motion of the fluid. With the almost same motivation to our paper, Dosh and Bors [8] couples physical models and image estimation techniques for modelling the movement of fluids. The estimated optical flow is corrected by a designed diffusion step through solving the Navier-Stokes equation. However, the weighted blending of input optical flow and simulated velocity field is hard to verify its physical accuracy. The blending may degenerate temporal coherency of output image sequences. Atmospheric images used in the paper can yield reasonable velocity field since satellite camera is very far from ground. However, fluid in daily life such as cigarette smoke is not adequate for an example of the paper where 2D fluid simulation is applied to projected real 3D fluid. Several researches in computer graphics have been interested in capturing fluid motion to generate fluid animation. Hawkins *et al*. [9] propose a capturing system for time-varying volumetric smoke. A laser sheet is swept repeatedly through the volume, and the scattered light is imaged using a high-speed camera. Li *et al*. [10] propose a video-based approach for producing water surface models by combining shape from shading (SFS) and shallow water model. Atcheson *et al*. [11] a Schlieren tomography system for capturing full 3D, non-stationary gas flows on a dense volumetric grid. Wang *et al*. [12] present an image-based reconstruction framework to model real water scenes captured by stereoscopic video. By combining an image based reconstruction with physically-based simulation, they fill in missing regions, remove outliers, and refine the geometric shape. A developed model of [13] is capable for simulating the event based runoff in the watersheds. [14] Propose the residence time distribution (RTD) characteristics, flow pattern, and dispersion coefficient of rough rice in a plug flow fluid bed dryer under various experimental conditions were investigated.

Extending the [15] paper, we approach the fluid animation on making new frames and analyzing the velocity field.

3. Our Method

3.1. Intermediate Frames Generation

We utilize PCA to synthesize intermediate frames between sequential input fluid image sequences. While PCA has been used in various computer graphics problems, image based rendering with PCA is close to our problem. TensorTextures [16] synthesizes a novel image with arbitrary illumination (camera) representation from image data with specific illumination(camera) directions. We follow the TensorTextures idea to synthesize intermediate image frame from input fluid images. Given image sequences I_i, $1 \leq i \leq T$ with N by M resolution, image data D is defined as $D \in R^{NM \times T}$, where T is the number of frames applied to PCA, NM is the number of pixels of each example image in Figure 1. Each column vector of N by M corresponds to the flattened vector with intensity value of each image.

We calculate principal components of D with Singular Value Decomposition (SVD). Column vectors of U_I is eigenvectors of DD^T with respect to image pixel data. Σ is diagonal matrix with eigenvalues of DD^T. Column vector of U_T is eigenvectors of D^TD. While U_I consists of eigenvector of DD^T, the eigenvector computation requires tremendous memory because both row and column numbers of DD^T equal to N*M and usually causes memory overflow for high resolution images. Instead, we utilize U_T for PCA process since the size of D^TD is $T \times T$ and $\ll N * M$. We use all eigenvectors since our image reconstruction requires lossless compression. Projected data P onto the space with eigenvectors as axes is defined as:

$$P = U_T^T \; * \; D^T \tag{1}$$

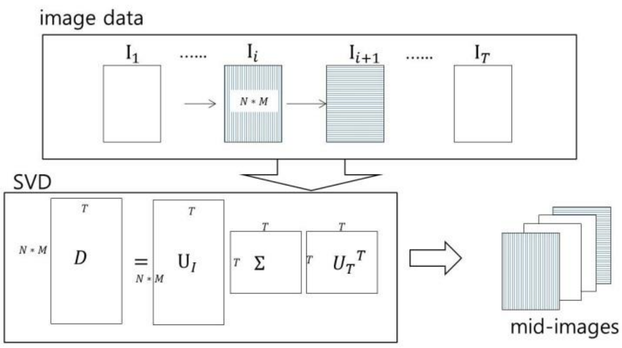

Figure 1. Systemic Diagram of Intermediate Frames Generation

For a representation vector v_t for specific time t in input image sequence, reconstructed $v_t^T * P$ equals to original image for time t. We assume that n intermediate images from i-th and $i + 1$-th input images. When u_i and u_{i+1} are i-th and $i + 1$-th row vector of U_T^T, respectively, k -th representation vector u_k i

$$u_k = \left(1 - \frac{k}{n-1}\right) u_i + \frac{k}{n-1} u_{i+1}, \;\; 0 \leq k \leq n-1 \tag{2}$$

Then, we obtain new k-th intermediate frame between i-th and i+1-th images by $u_k^T P^T$ in Figure 2.

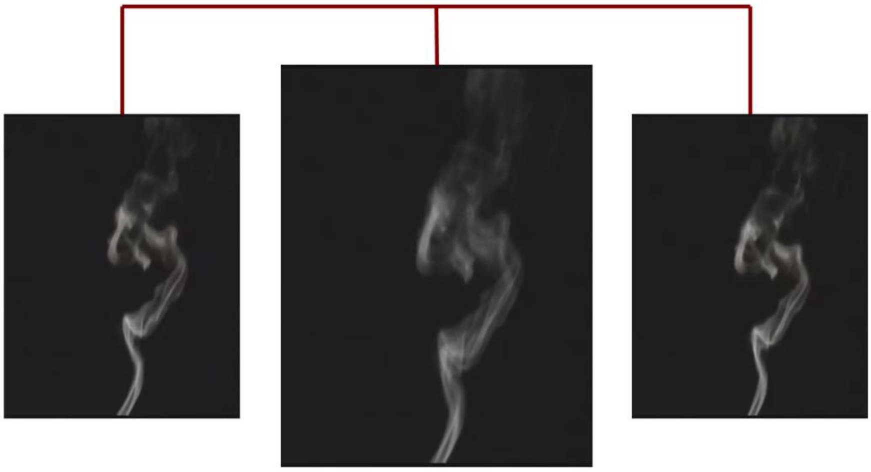

Figure 2. Generation of Intermediate Frame between Two Images

3.2. Velocity Field Generation

We generate optical flow for fluid movements using a groundbreaking optical flow estimation method [17] which is the combination of Lucas/Kanade (LK) [18] and Horn/Schunck (HS) [19] to achieve the best of the local and global differential methods. Lucas Optical flow methods have a common assumption: Grey values of image objects in subsequent frames do not change over time. LK takes less time to calculate the optical flow because of the small area window. However, LK has the disadvantage that cannot calculate the large movement because of using this small area window. On the other hand, HS calculates every pixel data to make optical flow in all frames, so it takes long calculation time. Considering this ambivalence, we use a combination of LK and HS to get the optical flow by calculating the selectively pixels depending on the features in Figure 3. with $I(x+u, y+v, t+1) = I(x, y, t)$ and displacement field (u, v), we obtain $\nabla I^T \cdot (u, v) = -I_t$. However, it is hard to apply the optical flow equation to fluid scene. Contrary to rigid body suitable for general optical flow method, features of fluid are not determined in straightforward way since smoke has not explicit boundary and liquid surface is not topologically invariant since liquid can be merged and collapsed. Therefore, without any special manipulations, fluid motion estimation is hard to achieve good results by current optical flow method. The hybrid method [14] minimizes the following energy function to obtain motion vector u:

$$E(u)= \int_{\Omega} (K_\rho * ((I_x u + I_y v + I_t)^2) + \alpha \parallel \nabla \mathbf{u} \parallel^2) dxdy \tag{3}$$

Where $u = (u, v)$ and K_p is Gaussian distribution with standard deviation ρ and smoothing weight α serves as regulation parameter. The first and second terms in Eq. 3 represents LK and HS formulations, respectively. We apply Eq. 3 to achieve fluid velocity fields with intermediate images from sequent input images. The calculated velocity fields advect our rendering primitives such as particle and streaklines. Streakline is defined as the locus of points of all the fluid particles that have passed continuously through a particular spatial point in the past. Dye steadily injected into the fluid at a fixed point extends along a streakline [20].

Streaklines can be expressed as,

$$\begin{cases} \dfrac{d\vec{x}_p}{dt} = \vec{u}_p(\vec{x}_p, t) \\ \vec{x}_p(t = T_p) = \vec{x}_{p0} \end{cases}$$

Where, $\vec{u}_p$ is the velocity of a particle P at location $\vec{x}_p$ and time t. The parameter T_p, parametrizes the streakline $\vec{x}_p(t, T_p)$ and $0 \leq T_p \leq t_0$, where t_0 is a time of interest.

Our Lagrangian rendering primitives move according to the velocity field through simple explicit advection: $p_{t+1} = p_t + u\Delta t$. For volume rendering, one could apply implicit advection of density fields using semi-Lagrangian [1] or back-and-forth [21] advection schemes. The semi-Lagrangian method is a numerical advection method and is widely used in numerical weather prediction models for the integration of the equations governing atmospheric motion. A Lagrangian description of a system focuses on following individual air parcels along their trajectories as opposed to the Eulerian description, which considers the range of change of system variables fixed at a particular point in space. A semi-Lagrangian scheme uses Eulerian framework but the discrete equations come from the Lagrangian perspective [22]. Recently, Back and Forth Error Compensation and Correction (BFECC) were developed for interface computation by using the level set method. It is known that it reduces numerical dissipation and diffusion in various advection steps in fluid simulation such as velocity, smoke density. BFECC can be implemented easily on top of the semi-Lagrangian integration of advection equations and moreover provides second order accuracy both in space and time.

As a result, new velocity field is generated between frame-i and intermediate frame. Comparing two images, transforming the vector field can be seen in the changing parts in Figure 4.

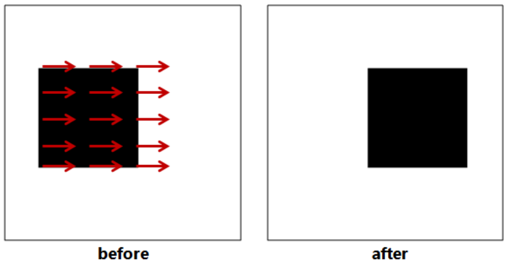

Figure 3. Combination of Lucas/Kanade and Horn/Schunck

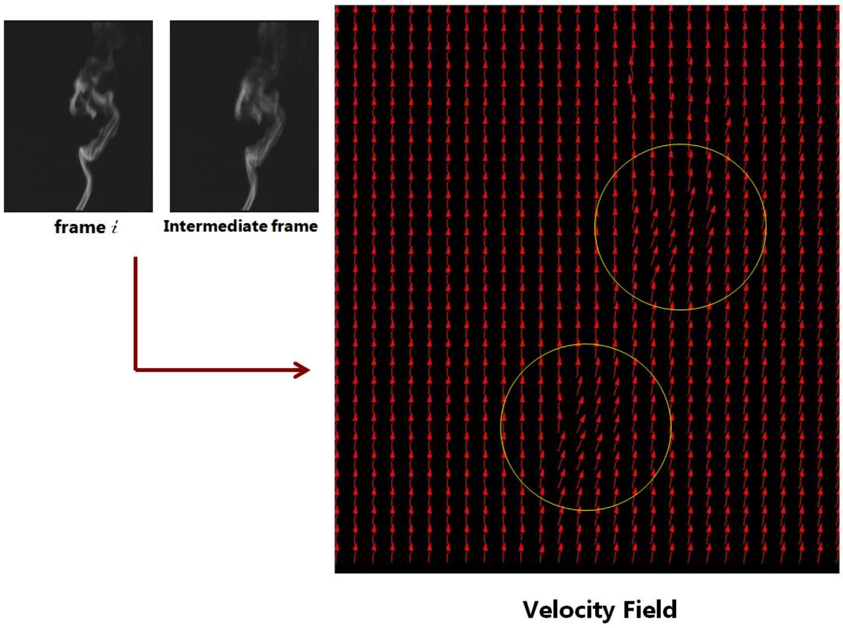

Figure 4. New Velocity Field between Frame-i and Intermediate Frame

4. Summary of Optical Flow

This paper utilizes optical flow methods such as Lucas-Kanade and Horn-Schunck methods. While the optical flow methods are common in computer vision or image processing, we introduce the concepts of the methods for novices by adopting the fundamental knowledge from [23].

Assumptions behind the Lucas-Kanade and Horn-Schunck Optical Flow Estimation methods

Both the Horn Schunck and Lucas-Kanade method assumes that brightness constancy does not change over time. Also, the Horn-Schunck method assumes that the flow field is globally smooth (or that neighboring velocities are nearly identical). The Lucas-Kanade method assumes that the velocity is locally constant, and neighboring points belong to the same patch that has similar motion.

Differences between Lucas-Kanade and Horn-Schunck methods

The Lucas-Kanade method is a sparse/local method, while the Horn-Schunck method is a dense/global one. The Lucas-Kanade would obtain less noise compared to the dense method as Horn-Schunck. The Horn-Schunck is a dense one, which means it will have denser flow fields. The pixel displacement of the Lucas-Kanade is constant within a small vicinity, whilst that of Horn-Schunck is smooth over the time domain.

The threshold to constrain Lucas-Kanade Optical Flow estimation

This is done to ensure there is a numerical stability. We obtain a useful measure of reliability of motion for each computation. Fields that have a fairly undefined structure are excluded from potential error because of this (flow while there seems to be no flow).

5. Singular Value Decomposition

The paper utilizes PCA and SVD without detailed mathematical explanation. We adopt fundamental descriptions on SVD from [24] for novice readers. The singular value decomposition of MxN matrix A is its representation as $A = U\ W\ V^T$, where U is an orthogonal MxM matrix, V - orthogonal NxN matrix. The diagonal elements of matrix W are non-negative numbers in descending order, all off-diagonal elements are zeros. The matrix W consists mainly of zeros, so we only need the first min (M,N) columns (three, in the example above) of matrix $U$to obtain matrix A. Similarly, only the first min (M,N) rows of matrix V^T affect the product. These columns and rows are called left and right singular vectors. The singular value decomposition has many useful properties. For example, it can be used to: solve underdetermined and overdetermined systems of linear equations, matrix inversion and pseudoinversion, matrix condition number calculation, vector system orthogonalization and orthogonal complement calculation.

The Jacobi algorithm was one of the first to perform the singular value decomposition. It reduces a rectangular matrix to a diagonal matrix by using a sequence of elementary rotations. The method can find all the singular values with high precision, including very small ones. However its performance was rather low, thus the iterative QR algorithms became more popular.

The basis of the most popular modern singular value decomposition algorithms lies in the matrix reduction to a bidiagonal form by orthogonal transformation (this problem is sufficiently simple and requires a finite number of operations to solve it) and its diagonalization by using an iterative QR algorithm. Usually, algorithms differ only by their QR algorithm implementation. At first, the algorithm family based on the Golub-Kahan-Reinsch algorithm was the most wide spread. It is this very same method that was implemented in LINPACK/EISPACK. The advantages of the method are its simplicity and compactness (300 lines of code). Unfortunately, it has no other advantages. Although this algorithm is capable of solving the problems, its convergence and precision are somewhat poor.

There are two singular value decomposition algorithms in the LAPACK library which eliminate the defects of the LINPACK algorithm. The first algorithm (xBDSQR and xGESVD subroutines), which was a prototype for an algorithm described here, has better precision and convergence than its LINPACK analog, so it replaces its predecessor.

It should be noted that the new algorithm finds small singular values of a bidiagonal matrix with better precision. The LINPACK variant of the QR iteration finds all the singular values with the same absolute error. The biggest singular value is obtained with an accuracy close to machine precision. It is sufficient for problems where the absolute error of singular values is critical: when solving systems of linear equations, inverting matrices, *etc* But sometimes smaller singular values are important, whose relative error appears to be too large. For instance, if the first singular value equals 1 and was found with absolute error 10^{-15} (15 significant digits), the singular value which is equal to 10^{-10} found with the same absolute error will have only 5 significant digits. The new algorithms find all the values with the same relative error, so that both singular values will have 15 significant digits.

Unfortunately, the errors of reducing a general matrix to bidiagonal form bring this advantage to nothing - they corrupt small singular values which can be found precisely by using the new algorithm. Therefore, if your matrix isn't bidiagonal, you will get the only advantage of a new algorithm - better convergence. However, some matrix types could be reduced to bidiagonal form without changing small singular values. Such algorithms are not presented on the site at the moment. If small singular values precision is critical, it is worth finding theoretical material and implementing it by yourself. There are few algorithms designed in this field (hopefully this will change for the better).

As stated above, there are two singular value decomposition algorithms in the LAPACK library. The second algorithm (which is the "divide-and-conquer" algorithm) divides a task of big bidiagonal matrix SVD decomposition into some smaller tasks which are solved by using the QR algorithm. This algorithm shows better performance than the QR algorithm when working with big matrices. For instance, the square matrix singular value decomposition by "divide-and-conquer" when N=100 is 2-4 times faster than by a simple QR algorithm (including the time required to reduce the matrix to bidiagonal form), and is 6-7 times faster when N=1000. However, this algorithm is technically too complex, so it is unlikely to be included in the ALGLIB library in the near future. If the performance of big matrices singular value decomposition is critical to your tasks, please refer to the LAPACK library.

6. Conclusion

The paper presents a fluid animation system which achieves smoke flow from input video without physics based fluid simulation. The intermediate image generation enhances fluid motion estimation over common optical flow methods. By experimental results, we demonstrate that our system effectively generates fluid animation following fluid motion in input video.

As future work, we will tackle 3D fluid animation using videos from multiple cameras. Corresponding point finding problem would be technical difficulty since it is difficult to define feature vector in fluid. 3D fluid reconstruction research will be also helpful to stereoscopic field. We assumed that the camera position is fixed. Fluid capture for moving camera is also our future research topic. We believe fluid animation field will inherit capture and reuse paradigm from character and facial animation.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2012R1A1A2006094).

References

[1] J. Stam, Editor. “Stable fluids. Proceedings of the 26th annual conference on Computer graphics and interactive techniques”, ACM Press/Addison-Wesley Publishing Co., **(1999)**, pp.121-128.
[2] N. Foster and R. Fedkiw, Editors. “Practical animation of liquids”, Proceedings of the 28^{th} annual conference on Computer graphics and interactive techniques, **(2001)**, pp. 23–30.
[3] M. Mu ̈ller, D. Charypar and M. Gross, Editors. “Particle-based fluid simulation for interactive applications”, Proceedings of the 2003 ACM SIGGRAPH/ Eurographics symposium on Computer animation, Eurographics Association, **(2003)**, pp. 154-159.
[4] B. Solenthaler and R. Pajarola, “Predictive-corrective incompressible SPH”, ACM transactions on graphics (TOG), vol. 28, **(2009)**, pp.1-6.
[5] M. Macklin and M. Mu ̈ller, “Position based fluids. ACM Transactions on Graphics”, (TOG), vol. 32, no. 4, **(2013)**, pp. 04.
[6] K. Raveendran, C. Wojtan, and G. Turk, Editors. “Hybrid smoothed particle hydrodynamics”, Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation, **(2011)**, pp. 33-42.
[7] R. J. Adrian and J. Westerweel, “Particle image velocimetry”, Number 30. Cambridge University Press, **(2011)**.
[8] A. Doshi and A. G. Bors, Editors. “Robust processing of optical flow of fluids”, Image Processing. IEEE Transactions on, vol. 19, no. 9, **(2010)**, pp. 2332–2344.
[9] T. Hawkins, P. Einarsson and P. Devebec, “Acquisition of time-varying participating media”, ACM Transactions on Graphics (TOG), vol. 24, **(2005)**, pp. 812-815.
[10] C. Li, D. Pickup, T. Saunders, D. Cosker, D. Marshall, P. Hall and P. Willis, “Water surface modeling from a single viewpoint video”, Visualization and Computer Graphics, IEEE Transactions on, vol. 19, no. 7, pp. 1242-1251.
[11] B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor and H. P. Seidel, “Time resolved 3d capture of non-stationary gas flows”, In ACM Transactions on Graphics (TOG), , vol. 27, **(2008)**, pp.1-9.
[12] H. Wang, M. Liao, Q. Zhang, R. Yang and G. Turk, “Physically guided liquid surface modeling from videos”, ACM Transactions on Graphics (TOG), vol. 28, **(2009)**, pp.1-11.
[13] T. Reshma, P. S. Kumar, Dr. M. J. R. Kanth Babu and K. S. Kumar, “Simulation of runoff in watersheds using SCS-CN and Muskingum-cunge methods using remote sensing and geographical information systems”, IJAST, vol. 25, December **(2010)**, pp.31-42.

[14] M. Khanali, S. Rafiee, A. Jafari and A. Banisharif, "Study of Residence Time Distribution of Rough Rice in a Plug Flow Fluid Bed Dryer", IJAST, vol. 48, November **(2012)**, pp.103-114.
[15] H. An and J. Park "A Lagrangian Approach on Video based Fluid Animation", The First Workshop on Art, Culture, Game, Graphics, Broadcasting and Digital Contents 2015, The Convergent Research Society Among Humanities, Sociology, Science and Technology, **(2015)**.
[16] A. M. Vasilescu and D. Terzopoulos "Tensor textures: multilinear image-based rendering", In ACM Transactions on Graphics (TOG), vol. 23, pp. 336-342.
[17] A. Bruhn, J. Weickert and C. Schnorr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods", International Journal of Computer Vision, vol. 61, no. 3, pp. 211-231.
[18] B. Lucas and T. Kanade "An iterative image registration technique with an application to stereo vision", In IJCAI, vol. 81, **(1981)**, pp. 674-679.
[19] B. Horn and B. G. Schunck, "Determining optical flow", Technical Symposium East, **(1981)**, pp. 319-331.
[20] http://en.wikipedia.org/wiki/Streamlines,_streaklines,_and_pathlines
[21] B. Kim, Y. Liu, I. Llams and J. Rossignac, Editors. "Flowfixer: Using BFECC for fluid simulation", Proceedings of the First Eurographics conference on Natural Phenomena, **(2005)**, pp. 51-56.
[22] http://en.wikipedia.org/wiki/Semi-Lagrangian_scheme
[23] http://www.timzaman.com/2011/04/2d-optical-flow/
[24] http://www.alglib.net/matrixops/general/svd.php

Authors

HaengA An, is an undergraduate student of Global School of Media, Soongsil University. She is currently working as an undergraduate research assistant of Computational Design Lab at Soongsil University.

Jinho Park, received his B.S. and M.S. degrees in Applied Mathematics in 1999 and 2001, respectively, and PhD in Computer Science in 2007 from Korea Advanced Institute of Science and Technology. He is an assistant professor in the Global School of Media, Soongsil University, South Korea. His research interests include fluid animation and scientific visualization.

