A Novel Algorithm for Fluid Simulation Using Incompressible Turbulence Function

Anh Q. Nguyen and Bac H. Le
University of Science, Ho Chi Minh City, Vietnam
Email: {nqanh, lhbac}@fit.hcmus.edu.vn

***Abstract*— In this paper, we present a novel algorithm for simulating fluids at high resolution quickly. Instead of solving the Navier-Stokes equations over a highly refined mesh, we only require a low grid resolution to resolve the underlying base flow. We then use a novel incompressible turbulence function and compute transport of turbulent energy using a complete model, which generate accurate production terms that allows us to capture turbulence effects. We will show how our technique complements previous work and demonstrate that it can efficiently generate detailed simulations with low computation cost and also suitable for parallel architectures.**

***Index Terms*—Physically Base Animation, Fluid Simulation, Incompressible Flow, Turbulence.**

I. Introduction

In physic, fluids fall into two categories incompressible and compressible flow. Incompressible flow is a liquid, such as water. Compressible flow corresponds to gas such as air or steam. Compressible flow is called compressible because we can easily change the volume of this fluid. All fluids, even water can change their volume. However, we simply ignore compressibility in fluids like water because it is very difficult and require very special condition to compress them. So we refer them simply as incompressible.

The phenomena of fluids such as smoke and water are fascinating to watch, but the physical simulation of fluids is one of the most challenging problems because of their chaotic, turbulent nature. To simulate fluids, there are two common techniques are grid based and particle based simulations. Grid based simulations are typically highly accurate, although relatively slow compared to particle based solutions. Particle based simulations are usually faster, but they usually do not look as good as grid based simulations. In this work we will focus on fluid simulations with grid based techniques due to their widespread use.

Although many methods have used grid based techniques to produce visually compelling results, the size of the grids that these techniques can use is limited by the amount of computational power available. Because turbulence in fluids extends over many scales, a direct simulation resolving all details requires a costly high resolution calculation. The high cost of a direct numerical simulation has lead to increasing interest in algorithms to synthetically generate turbulence for augmenting low resolution simulations.

As a consequence, many authors have developed algorithms that add noise or turbulence to the simulations. Unfortunately, current methods, especially complex ones that capture effects more accurately, often rely on strong assumptions about the production of turbulence, which is a very important factor that strongly determines the quality of the dynamics generated with the turbulence model. In our technique, we have designed our method to generate small scale fluid detail procedurally. To avoid costly computation, we only solve the Navier-Stokes equations for very coarse base simulations. We then use a full two equation energy transport model with physically plausible production terms. So, the large scales are computed using a low resolution fluid solver and a full energy function is used to compute details and exactly the production terms of turbulence model.

Our contributions are as follows:

- A scalable, incompressible turbulence function that can generate turbulent energy.
- An energy transport model base on $k-\omega$ model that realistically captures the turbulence production.
- A new and robust algorithm to synthetic turbulence from low resolution fluid solver.

II. Related Work

Fluid simulations have become popular in computer graphic when Stam [1] introduced semi-Lagrangian method for simulating stable fluids. Although this grid based technique successfully simulates stable fluids, the size of the grids is limited by the amount of computational power available. Many subsequent works have improved initial algorithm. Fedkiw [2] introduced vorticity confinement, which detecting and amplifying existing vortices to combat dissipation. Selle [3] used vortex particles for higher simulations. Back Forth Error Compensation and Correction (BFECC) was instroduced by Kim [4]. In [5], Selle presented semi-Lagrangian MacCormack methods. One can also apply higher order advection methods such as Molemaker [7] and Kim [8]. All these methods allow fluid features to be robustly resolved but they still have problems when increasing grid size.

Manuscript received November 2, 2012; revised January 5, 2013.

doi: 10.12720/lnit.1.1.39-43

Many authors have replaced basic fluid simulation with synthetic turbulence or noise. For example, Kolmolgorov noise in Stam [9] and Lamorlette [10], curl noise Bridson [11] can be used to enhance the visual fidelity of fluid simulations by coupling the noise to produce a more detailed flow. Kim [12] and Narain [13] decide where to add noise using information from the previous simulation and then add it as a post process, so noise can be added where it is best suited. In the other hand, Schechter [13] and Pfaff [14] determine where to add noise and couple the noise to the Navier-Stokes equations at the same time by using use energy transport models. These methods were able to represent anisotropic effects near obstacles. However, they cannot handle free stream turbulence such as rising smoke, and requires careful tuning for the particle decay in order to obtain turbulence. In Kim [15] and Pfaff [16], they use wavelet decomposition to determine local turbulence intensities. Our method is most similar to these methods, but we not only use wavelet turbulence for synthesis, but also improve the coupling with the fluid simulation and energy transport between different scales by using the complete $k-\omega$ model.

In order to handle very high resolution grids, Wicke [17] introduces a reduced order model that can handle large grids at a small cost. However, this method lacks the physical realism. In [18], Horvath introduces method that can run large scale two dimensional algorithms on GPU. Lentine [19] uses only a coarse grid projection for simulation high resolution grids with effectively reduces the amount of time required for the Poisson solve by using a coarse grid projection. McAdams [20] presents parallel multi grid Poisson solver method which can increase grid size but also increase memory requirement. All these methods successfully simulate very high resolution grids but they are not suitable for real time application and require lot of memory for solving equations during the simulation process.

III. Synthetic Turbulence

Because the computational effort increases strongly with the grid size, our approach is driven by a low resolution Eulerian fluid solver and a synthetic turbulence system as described in Fig. 1.

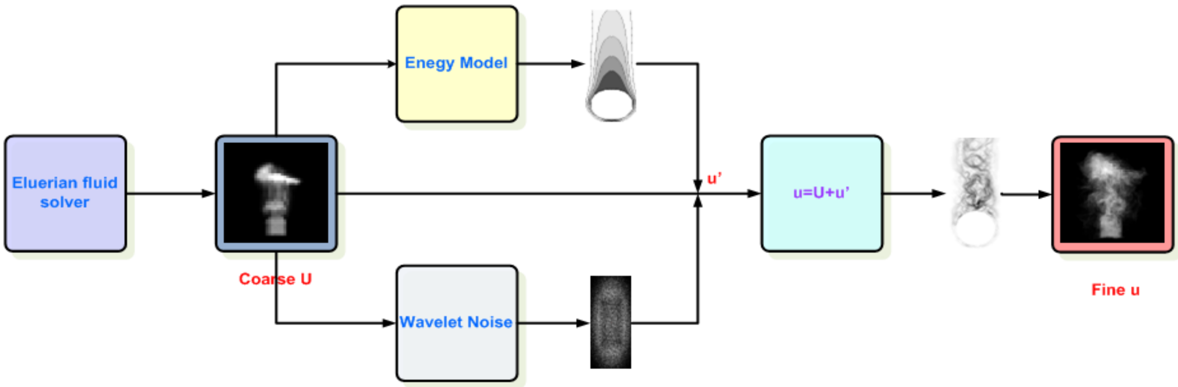

Figure 1. An overview of our method: A low resolution grid based simulation is generated by Eulerian fluid solver using semi-Lagrangian method. Procedural turbulence is added according to energy model and wavelet noise. The final velocity is given by the large scale velocity from Eulerian fluid solver and the small scale turbulent velocity.

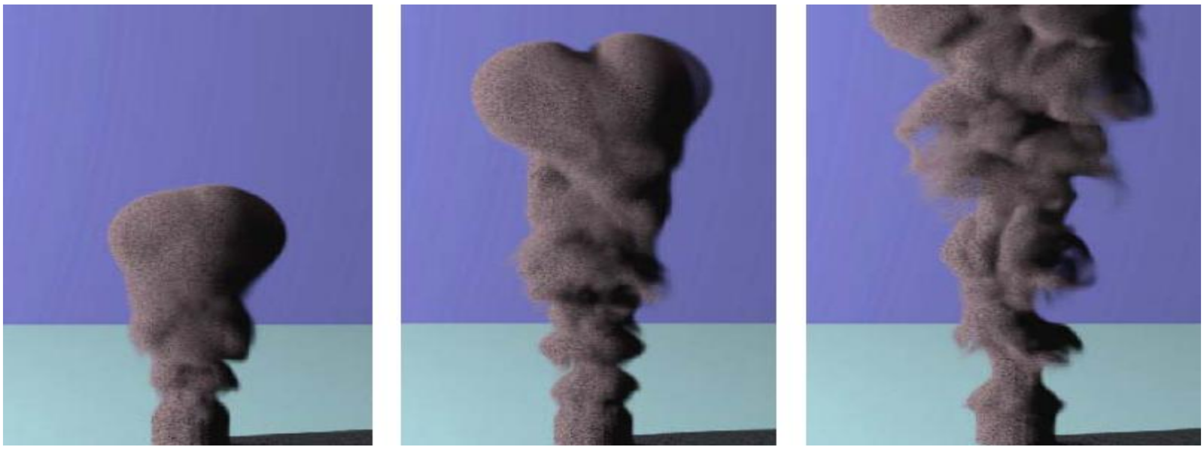

Figure 2. A free smoke simulation on 1280x2560x1280 resolution from only 128x256x128 grid size.

A. Navier-Stokes and RANS Equations

On the simulation grid, we solve the incompressible Navier-Stokes equations given by

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}.\nabla\vec{u} + \frac{1}{\rho}\nabla p = \vec{F} + v\nabla.\nabla\vec{u} \tag{1}$$

where $\vec{u}$ is the velocity field, ρ is the density of the fluid, $\vec{F}$ are any external forces (such as gravity), v is the fluid viscosity, and p is the fluid pressure. In most cases, viscosity plays a minor role in the simulations, and thus we often drop it. The Navier-Stokes equations without viscosity are called the Euler equations:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}.\nabla \vec{u} + \frac{1}{\rho}\nabla p = \vec{F} \tag{2}$$

$$\nabla \cdot \vec{u} = 0 \tag{3}$$

In traditional simulations, we solve (2) and (3) over a fine simulation grid. But this method requires highly computational cost. Like other approaches in simulations using turbulences. We will use Reynolds Averaged Navier-Stokes (RANS) equations instead of Navier-Stokes equations.

In RANS equations, we break the velocities and pressure down into their mean and fluctuating parts:

$$u_i = U_i + u_i^{'} \; ; \qquad p = P + p' \tag{4}$$

Here u_i and p are the instantaneous variables we are decomposing, U_i and P are the mean flow values and $u_i^{'}$ and p' represent the turbulent fluctuations.

From (2), (3), and (4) we will have RANS equations:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{5}$$

$$u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(v\frac{\partial u_i}{\partial x_j} - u_i^{'} u_j^{'} \right) \tag{6}$$

where the part $u_i^{'} u_j^{'}$ in (6) is known as the Reynolds stress tensor, τ_{ij}, which represents the influence of the turbulent fluctuations on the mean flow.

The most common approach to compute Reynolds stress tensor is known as the Boussinesq approximation [21], [22]. The Boussinesq approximation assumes the Reynolds stress tensor is proportional to the mean flow stress tensor.

$$\tau_{ij} = v_T S_{ij} \tag{7}$$

where v_T is the turbulent viscosity and S_{ij} denotes the strain tensor given by,

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{8}$$

B. Energy Transport Model

The presented Reynolds tress model requires a high grid resolution. The turbulence of the fluid flow will be described as an energy representation. We use the complete $k-\omega$ model [23], [24] to simulate the energy dynamics that allows us to inject full energy for our simulations. A full discussion of different turbulence models used in CFD can be found in Wilcox [24] and Pope [30].

Unlike one-equation models that requires strong assumptions, $k-\omega$ is a complete two-equation model. While k represents the turbulent kinetic energy contained in the smaller scales, ω is commonly thought of as the characteristic frequency of the turbulent decay process or the time scale which dissipation of the turbulent energy occurs. Our method uses the most popular $k-\omega$ model from Wilcox [25] which is commonly referred to as the standard $k-\omega$ model:

$$\frac{Dk}{Dt} = \nabla\left(\frac{v_T}{\sigma_1}\nabla k \right) + P - k\omega \tag{9}$$

$$\frac{D\omega}{Dt} = \nabla\left(\frac{v_T}{\sigma_\omega}\nabla \omega \right) + C_{\omega 1}\frac{P\omega}{k} - C_{\omega 2}\omega^2 \tag{10}$$

The turbulent kinetic energy k is computed in (9) and the specific dissipation rate ω is computed in (10). The turbulent viscosity (or eddy viscosity) v_T is a virtual viscosity. It is not a property of the fluid but a property of the flow field and hence will vary throughout the flow domain. Turbulent viscosity describes the effect of small scale turbulent motion, in $k-\omega$ model, it is defined as:

$$v_T = \frac{k}{\omega} \tag{11}$$

The constants in (9), (10) can be found in Wilcox [25].

Like other two-equation models, in $k-\omega$ model the production P is the energy transfer from the large scale flow field to the small scale turbulence, is defined as:

$$P = 2v_T \sum_{ij} S_{ij}^2 \tag{12}$$

where S_{ij} given by (8)

In [24], Wilcox showed that the $k-\omega$ model not only performs well for free flows but also for more complicated adverse pressure gradient flows and separated flows.

C. Turbulence Synthesis

The production of turbulent energy is in the energy range of fluid, while the dissipation to heat grows stronger for small scales. Between these two lies is inertial subrange, which transporting the energy from large to small scales.

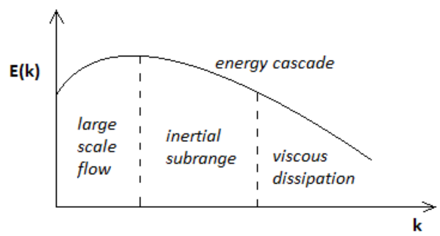

Figure 3. Energy cascade of fluid flow.

The statistical properties of turbulent energy can be described as a distribution spectrum. In Fig. 3, we show this spectrum which includes three sections:

- Large scale flow: In this region, large scales are dominant, the production of turbulence behavior is strongly dependent on the flow and very difficult to describe.

- Inertial subrange: In this region, the fluid flows start losing energy, although most of the aspects such as velocity, pressure, etc are very difficult to handle, Kolmogorov [26] famously showed that the slope of the energy spectrum is always -5/3, also known as five-thirds law.
- Viscous dissipation: In here, the main energy dissipation occurs due to viscosity.

Frisch [27] proposes a very reasonable approximation for Kolmogorov theory to compute total energy at band k :

$$e(k) = C\varepsilon^{\frac{2}{3}}k^{-\frac{5}{3}} \tag{13}$$

where C and ε are the Kolmogorov constant and the mean energy dissipation rate. The kinetic energy at a grid cell x is known as:

$$e(x) = \frac{1}{2}\left|u(x)\right|^2 \tag{14}$$

From (13) and (14) we have:

$$e(2k) = e(k)2^{-\frac{5}{6}},\ \left|u(x,2k)\right| = \left|u(x,k)\right|2^{-\frac{5}{6}} \tag{15}$$

The turbulence function is a series version of (15):

$$y(x) = \sum_{i=0}^{n}\left|u(x)\right|2^{-\frac{5}{6}i} \tag{16}$$

To enhance and make the simulation look more natural, like in Bridson [11], Kim [15] and Narain [13], we use Wavelet Noise [28] in construction an incompressible turbulence function. The noise is guaranteed to exist only over a narrow band and help the fluid flow look more turbulent but still does not affect much on the result of the simulations. The Wavelet Noise function $\wp$ is a scalar function, in 3D we have:

$$w(x) = \left(\frac{\partial\wp_1}{\partial y} - \frac{\partial\wp_2}{\partial z}, \frac{\partial\wp_3}{\partial z} - \frac{\partial\wp_1}{\partial x}, \frac{\partial\wp_2}{\partial x} - \frac{\partial\wp_3}{\partial y}\right) \tag{17}$$

Our final turbulence function includes both noise function and energy model is:

$$y(x) = \sum_{i=0}^{n} w(x)\left|u(x)\right|2^{-\frac{5}{6}i} \tag{18}$$

The total velocity u of the fluid flow is then given by the large scale flow velocity U , computed from the low resolution Eulerian solver, and the turbulence velocity as in (18):

$$u = U + \alpha\sum_{i=0}^{n} w(x)\left|u(x)\right|2^{-\frac{5}{6}i} \tag{19}$$

The parameter α in (19) encodes the shape of the assumed energy spectrum, and can be used to increase or decrease the strength of turbulence of the fluid flow.

IV. Implementation and Results

We have implemented our model to execute the Eulerian fluid solver for low resolution simulation and turbulence function for high resolution simulation. For the underlying Eulerian solver, we use semi-Lagrangian method as described in Stam [1] and Selle [29]. The turbulence function is computed base on our final equation for generating high resolution velocity field (19). The full algorithm is described below, and illustrated in Fig. 1:

---

1. *Initialize* simulation scenario
2. *For* each n do
3. // Grid-based fluid solver, semi – Lagrangian
4. *Advect:* $q^{n+1} = advect(q^n, \Delta t, \vec{u})$
5. *Project:* $U \leftarrow project(\Delta t, \vec{u})$
6. // Physics based model, synthetic turbulence
7. *Compute* turbulent viscosity: $v_T \leftarrow k / \omega$
8. *Compute* production terms: $P \leftarrow P = 2v_T\sum_{ij}S_{ij}^2$
9. *Integrate:* $k \leftarrow k + \Delta t(|P| - k\omega)$
10. *Integrate:* $\omega \leftarrow C_{\omega 1}P\omega / k - C_{\omega 2}\omega^2$
11. *Synthesize:* $u \leftarrow U + \alpha\sum_{i=0}^{n} w(x)\left|u(x)\right|2^{-\frac{5}{6}i}$
12. *Integrate:* $x \leftarrow x + \Delta tu$
13. *Advection* u
14. *End For*
15. *Render* simulation data

Although the complexity of our algorithm is still $O(n^3)$ due to we use standard fluid solver for low resolution simulation, we are able to simulate a very high effective resolution because a high-resolution grid is not required. In Fig. 2, we show a simulation of free smoke at very high resolution. Complete descriptions are available in the figure captions. Our method appears to resolve more high frequency detail and runs five times faster than the full solver.

The advantage of our method is wavelet noise successfully adds small detail to the overall flow that make our simulation look more natural and the complete $k-\omega$ model will handle the energy dissipation that ensure the simulation is correct. This make our algorithm appears to resolve more high frequency detail at a very low cost.

By design, all steps of our algorithm can be implemented parallel. That also suggests that the algorithm will perform very well on GPUs such as CUDA. One interesting avenue of future work would be to integrate our method with other CFD model. Our method provides an interesting twist in that it can be used as a prolongation operator for simulating a divergence free flow field.

References

[1] J. Stam, "Stable fluids," in *Proc. ACM SIGGRAPH*, 1999.
[2] R. Fedkiw, J. Stam, and H. Jensen. "Visual simulation of smoke," in *Proc. ACM Siggraph*, 15–22. 2001.

[3] A. Selle, N. Rasmussen, and R. Fedkiw. “A vortex particle method for smoke, water and explosions,” in *Proc. Siggraph*, 2005.

[4] B. Kim, Y. Liu, I. Llamas, and J. Rossignac. “Flowfixer: Using BFECC for fluid simulation,” in *Proce. Eurographics Workshop on Natural Phenomena*, 2005.

[5] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac. “An unconditionally stable maccormack method,” *Journal of Scientific Computing*, vol. 35, no. 2-3, pp. 350-371, 2008.

[6] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, and M. Miller, “Rotation, scale, and translation resilient public watermarking for images,” *IEEE Trans. Image Process.*, vol. 10, no. 5, pp. 767-782, May 2001.

[7] J. Molemaker, J. M. Cohen, S. Patel, and J. Noh. “Low viscosity flow simulations for animation,” in *Proce. ACM SIGGRAPH / Eurographics Symp. on Comp. Anim*., 2008, pp. 9–18.

[8] D. Kim, O. Young Song, and H. S. Ko. “A semilagrangian cip fluid solver without dimensional splitting,” *Comput. Graph. Forum(Proc. Eurographics)*, vol. 27, no. 2, pp. 467–475. 2008.

[9] J. Stam and E. Fiume. “Turbulent wind fields for gaseous phenomena,” in *Proc. Siggraph*, 1993, pp. 369–376.

[10] A. Lamorlette and N. Foster. “Structural modeling of flames for a production environment,” *ACM Trans. Graph. (Siggraph Proc.)* vol. 21, no. 3, pp. 729–735, 2002.

[11] R. Bridson, J. Houriham, and M. Nordenstam. “Curl-noise for procedural fluid flow,” *ACM Trans Graph*, vol. 26, no. 3, pp. 46. 2007.

[12] H. Schechter and R. Bridson. “Evolving sub-grid turbulence for smoke animation,” in *Proc. ACM Siggraph/Eurographics Symp. on Comput. Anim*, 2008, pp. 1–7.

[13] R. Narain, J. Sewall, M. Carlson, and M. C. Lin. “Fast animation of turbulence using energy transport and procedural synthesis,” *ACM SIGGRAPH* Asia papers, Article 166, 2008.

[14] T. Pfaff, N. Thuerey, A. Selle, and M. Gross. “Synthetic turbulence using artificial boundary layers,” *ACM Transactions on Graphics*, vol. 28, no. 5, 121:1–121:10, 2009.

[15] T. Kim, N. Thuerey, D. James, and M. Gross. “Wavelet turbulence for fluid simulation,” in *Proc ACM SIGGRAPH*, August 2008, vol. 27, no. 3.

[16] T. Pfaff, N. Thuerey, J. Cohen, S.Tariq, and M. Gross “Scalable fluid simulation using anisotropic turbulence particles,” in *Proc ACM SIGGRAPH Asia (ACM Transactions on Graphics),* Seoul, Korea, December 15-18, 2010, pp. 174:1-174:8.

[17] M. Wicke, M. Stanton, and A. Treuille. “Modular bases for fluid dynamics,” *in SIGGRAPH ’09: ACM SIGGRAPH*, New York, NY, USA, 1–8. 2009.

[18] C. Horvath and W. Geiger. “Directable, high-resolution simulation of fire on the gpu,” *ACM Trans. Graph.* Vol. 28, no. 3, pp. 1–8, 2009.

[19] M. Lentine, W. Zheng, and R. Fedkiw, “A novel algorithm for incompressible flow using only a coarse grid projection,” *SIGGRAPH ACM TOG*, vol. 29, no. 4, 2010.

[20] A. McAdams, E. Sifakis, J. Teran. “A parallel multigrid Poisson solver for fluids simulation on large grids,” *ACM SIGGRAPH/Eurographics Symposium for Computer Animation.*2010.

[21] J. Boussinesq, "Essai sur la théorie des eaux courantes," *Mémoires Présentés par Divers Savants à l'Académie des Sciences XXIII*, 1, pp. 1-680, 1877.

[22] F. G. Schmitt, "About Boussinesq’S turbulent viscosity hypothesis: Historical remarks and a direct evaluation of its validity," *Comptes Rendus Mécanique*, vol. 335, no. 9-10, pp. 617-627, 2007.

[23] A. N. Kolmogorov, “Equations of turbulent motion of an incompressible fluid,” *Technical Report 1-2, Izvestia Academy of Sciences, USSR*; Phusics, 1942.

[24] D. C. Wilcox, “Turbulence Modeling for CFD,” 2nd Edition. Griffin Printing, California, 1998.

[25] D. C. Wilcox, "Reassessment of the scale-determining equation for advanced turbulence models," *AIIA Juarnal*, pp.1299- 1310, 1988.

[26] A. Kolmogorov, “The local structure of turbulence in incompressible viscous fluid for very large reynolds number,” *Dokl. Akad. Nauk* SSSR 30, 1941

[27] U. Frisch, “Turbulence: The Legacy of A. N. Kolmogorov” Cambridge University Press, 1995.

[28] R. Cook, T. Derose, “Wavelet noise,” in *Proc. ACM Siggraph*, 2005.

[29] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac. “An unconditionally stable MacCormack method,” *Journal of Scientific Computing*, 2008.

[30] S. POPE, “Turbulent Flows,” *Cambridge University Press*, 2000.

Anh Q. Nguyen was born in Hai Phong City, Vietnam 1989. He got the Bachelor of Science degree with honor 1st class from University of Science in 2011.
His research interests are computer graphic: fluid simulation, machine learning, and robot vision. In later 2011, he joined NICTA Lab at Australian National University as an internship student. He is now a member of PCL to develop Point Cloud Library.

Bac H. Le is Associate Professor at University of Science, Vietnam. He got PhD degree in 1999 at University of Science.
His research interests are computer graphic, computer vision, data mining and intelligent systems. He has published widely many papers on these areas. He is also a program committee member of many conferences such as RIVF 2012, ICTACS 2011.