3D RECONSTRUCTION OF THE DENSITY FIELD OF A JET USING SYNTHETIC BOS IMAGES

V. TODOROFF [1,C], A. PLYER [2], G. LE BESNERAIS [2], F. CHAMPAGNAT [2], D. DONJAT [1],
F. MICHELI[1], P. MILLAN [1]

[1] ONERA, DMAE, BP74025, 2 avenue Edouard Belin FR-31055 Toulouse CEDEX 4, France

[2] ONERA, DTIM, Chemin de la Hunière BP 80100 FR-91123 Palaiseau CEDEX, France

[c]**Corresponding** author: Tel.: +33562252525 (2229); Email: Violaine.Todoroff@onera.fr

KEYWORDS: 3D tomography, Background Oriented Schlieren, inverse problem, Graphic Processing Unit

ABSTRACT: In this paper, we investigate a direct method for 3D Background Oriented Schlieren (BOS) reconstruction of an instantaneous jet. The observed angles of deflection are expressed as the integral of the density gradient along the optical path. After discretization, one has to solve an inverse problem where the observation matrix is essentially the composition of a tomographic projection matrix with a 3D gradient operator. While 2D and 3D-axisymetric cases had been successfully solved using only one projection, the general 3D case requires several projections in order to define correctly the solution. Yet, in practical experimental setups the use of hundreds of projections, as used in conventional computed tomography (CT), is prohibited. Our objective is to reconstruct an instantaneous 3D density field with a good accuracy using only a limited number of views. Contrary to what has been done previously we do not first reconstruct the gradient field by tomography then integrate it but instead we use a direct approach and reconstruct directly the density field. The resulting inverse problem is not only ill-posed but severely under-determinate, hence its resolution requires some regularization process. This is done by minimizing a compound criterion combining a data term and a regularization term. We use a Variable Splitting (VS) approach to derive an alternate minimization over both terms of the global criterion. As conventional solvers do not seem adequate to such a huge problem, we propose a fast implementation on Graphic Processing Units (GPU). The resulting software is validated on synthetic BOS images of a 3D density field issued from numerical calculation. The effect of using a limited number of projections for various levels of noise is studied in order to evaluate the expected performance of the software on a real 3D BOS experiment.

1. INTRODUCTION

Density is a fluid property which is not easily measured from experiments. Schlieren methods, based on the deflection of light rays through a flow of different refractive indexes, (corresponding to different densities through the Gladstone-Dale relationship), are sensitive to density gradients. While enabling the visualization of density gradient such as conventional Schlieren, Background Oriented Schlieren (BOS) can as well quantify the value of the deflection angle of a light ray and provide indirect means of density measurement. This optical technique is fairly recent, primary work had been done by Meier [1] in 2000. His work has been followed by studies on the different application fields of the BOS technique, from supersonic jets [2] to vortices [3] and flames [4]. Besides giving a quantitative result, the BOS setup is rather unsophisticated and cheap. In fact, in its simplest form, it requires only a camera, a high frequency background and proper illumination. The BOS procedure consists in recording a reference frame, then another one with the presence of the flow between the camera and the background. The underlying principle is the following: a light ray, issued from the background is bent in presence of the flow and impacts the CCD of the camera on a point different from the reference ray. In order to evaluate the displacement, that is to say the difference, in pixels, between the impact point of two rays issued from the same point on the background, image correlation algorithms, similar to the ones used in PIV, are used. In our case the software FOLKI-SPIV [5] developed at ONERA is utilized to obtain the displacement field. From it, deviation angles can be derived and related to the gradient of density via the following equation,

$$\varepsilon_u = K \int \frac{\partial \rho}{\partial u} ds \quad (1) \quad \text{where } u \in \{x, y, z\}$$

where ε_u denotes the deflection angle along one of the three directions, ds the optical path and K the Gladstone-Dale constant, commonly taken at $0.23 \cdot 10^{-3}$ m³/kg for air at ambient pressure and temperature. It is then theoretically possible

to reconstruct the 3D density gradient or density field if more than one projection is available by solving the inverse problem. That is to say, finding the density field compatible with the measured resulting deflection angles.

The BOS inverse problem is close to the conventional computed tomography (CT) problem, tools from it were then used at first. At the very beginning most reconstructions had been conducted either on axisymmetrical flows, (in this case one projection is sufficient and the density field is computed from an Abel inversion transform [6]) or by using the Filtered Back Projection (FBP) algorithm [7]. This strategy, which is rather computationally efficient, requires a large number of projections to yield a satisfying result and is not suitable for a small number of projections or incomplete data [8]. Yet, the mean density field around a spike in a Mach 2 flow had been successfully reconstructed from 22 projections by using interpolated data where those were missing [9]. Algebraic techniques received less attention in the BOS community and were then, until recently, not used. Yet, confronted to incomplete and noisy data their behaviour tends to be better and they are therefore an interesting opportunity to consider. To our knowledge, the first work to use algebraic techniques in the BOS field is Ihrke's [10]. He reconstructed the density field above a bunsen burner flame from 16 projections, his work mainly focused on a visual approach rather than a quantitative one but he definitely sets the path for scientific applications of the method. Recently, this approach has also been chosen to reconstruct the mean density field around an asymmetric body in a Mach 2 flow from 19 projections [11]. Density gradients in the three directions were reconstructed with ART then integrated to obtain the density field.

The present work addresses 3D BOS and has three main original features. First, in contrast with previous works, we do not first reconstruct the gradient field by some tomographic method then integrate it but instead we propose a direct approach and reconstruct directly the density field from measured deviations. Second, the resulting inverse problem is cast into the minimization of a regularized criterion by an alternate scheme derived from a Variable-Splitting (VS) approach. Third, we exploit the fact that the problem is parallelizable by means of Graphics Processing Units (GPU) architecture to dramatically speed up the calculation.

The paper is organized as follows. The proposed approach is described in Section 2: the chosen direct formulation and VS strategy are presented in sub-Section 2.1; data term minimization in 2.2 and regularization term minimization in 2.3. The overall minimization is described in 2.4. Section 3 presents a simulation study on synthetic BOS images of a 3D density field issued from numerical calculation. Concluding remarks are given in Section 4.

2. SOLVING THE INVERSE PROBLEM

The chosen direct, or end-to-end, approach of the problem requires Eq. (1) to be discretized in the form,

$$A\rho = \varepsilon \quad (2)$$

where ρ is the density field, and ε is the concatenation of deflection angles computed from the displacement fields in the three directions. The observation matrix A is the composition of the ray tracing matrix of tomography and a derivation matrix, multiplied by the Gladstone-Dale constant.

Strictly speaking BOS inverse problem is not linear since the ray tracing matrix may change as a function of the reconstructed density. Yet in practice deflections are small and the paraxial approximation can be used [12].

2. 1. Criterion definition and variable splitting (VS) strategy

Dealing with instantaneous images, the noise inherent to any experiments complicates the reconstruction since it is not diminished by averaging several images. As this kind of problem is ill-posed, direct inversion of the observation matrix A in Eq. (2) would lead to a dramatic noise amplification in the reconstructed field. Some regularization is required. The idea besides the concept of regularization is that data fidelity is not enough and a priori information on the sought 3D density field needs to be added. We are then no longer trying to solve the linear system of Eq. (2) but to minimize the following functional,

$$J(\rho) = \left\| A\rho - \varepsilon \right\|^2 + \alpha R(\rho) \quad (3)$$

where the first term is the data fidelity term and the second one is the regularization term, which contains the a priori information on the volume. The balance between both terms is set by the regularization parameter α.

Numerous regularization functions can be found in the literature [13]. Quadratic regularization, derived from Tikhonov's regularization principles leads to well-behaved criteria but tends to oversmooth the result. More recent approaches, such as total variation [14], are more computationally demanding but can also better reconstruct discontinuities in the density field. In some cases, it might be necessary to work with regularization functions that are non strictly convex and then require a non trivial strategy to be minimized. Following recent works in image restoration [15], we address all these regularization settings in a common formulation based on Variable Splitting (VS). It consists in defining an augmented criterion:

$$J(\rho, v) = \|A\rho - \varepsilon\|^2 + \alpha R(v) + \beta \|\rho - v\|^2 \quad (4)$$

A new, intermediate variable v is introduced to "split" the problem into two sub-problems related to the data term and to the regularization term. The criterion of Eq. (4) is minimized subject to the constraint $\rho = v$ by a Lagrangian approach. The minimization algorithm that we use is derived from the algorithm SALSA developed in [15]:

VS algorithm

Set k=0, choose α>0, β>0, ρ_0, v_0 and λ_0.

Repeat

$$\rho^{(k+1)} = \min_x \|A\rho - \varepsilon\|^2 + \beta \|\rho - v^{(k)} - \lambda^{(k)}\|^2 \qquad (VS1)$$

$$v^{(k+1)} = \min_v \alpha R(v) + \frac{\beta}{2} \|\rho^{(k+1)} - v - \lambda^{(k)}\|^2 \qquad (VS2)$$

$$\lambda^{(k+1)} = \lambda^{(k)} - (\rho^{(k+1)} - v^{(k+1)}) \qquad (VS3)$$

$$k \leftarrow k+1$$

Until stopping criterion is satisfied.

Complete theoretical justifications of this algorithm can be found in Ref. [15] and will not be discussed here. Let us only remark that the VS algorithm is the iterative minimization of two simpler problems. The first one (Eq. (VS1)) is made up of the original data term of Eq. (4) associated with a simple quadratic and separable regularization term, it is essentially a quadratic fit of the data. The second one (Eq. (VS2)) combines the regularization term associated with a simple observation equation and can be considered as a denoising, or smoothing step. Finally, the third equation (Eq. (VS3)) consists in the correction of λ, the Lagrangian parameter associated to the equality constraint $\rho = v$. Hence VS provides a rigorous framework for iterative fit/smoothing empirical strategies such as the one presented in [16]. Methods for solving the two main optimization sub-problems (VS1) and (VS2) are the subject of the two following sub-Sections.

2. 2. Data term minimization

In this problem as in general in computerized tomography, the matrix A can not be stored due to its size and the coefficients are then calculated on the fly. Here we first detail the relationship between the global algorithm presented above, which involves an abstract observation matrix A, and the operators which derive from the specific BOS observation Eq. (1). In such a "coefficient-oriented strategy" the direct problem we are solving can be modelled as

$$\varepsilon_{ui} = \sum_{m=1}^{N} w_{im} D_u \rho_m \quad (5) \quad \text{where} \quad u \in \{x, y, z\},$$

and where w_{im} represents the weight of the contribution of the gradient of the voxel ρ_m to the image of deviations measured by camera i, or the projection i, using the vocabulary of tomography. D_u is the derivation operator. In order to update the voxel value, one has to compute the difference between the measured projection and the current projection obtained from the current estimated volume by means of Eq. (5):

$$\Delta r_{ui}^{(k)} = \varepsilon_{ui} - \sum_{m=1}^{N} w_{im} D_u \rho_m^{(k)} \quad (6) \quad \text{where} \quad u \in \{x, y, z\}$$

where ε_i represents the measured ith projection.

Therefore the voxel value can be updated taking into account the difference previously calculated and the simple regularization term carried by the secondary variable, which appears in Eq. (VS1). This amounts to a steepest descent gradient update,

$$\rho_m^{(k+1)} = \rho_m^{(k)} + \gamma (\sum_i d\Delta r_i^{(k)} - 2\beta(\rho_m^{(k)} - v_m^{(k)} - \lambda_m^{(k)})) \quad (7)$$

where γ is the step of the steepest descent scheme and $d\Delta r_i$ is the update value.

Since only gradient information is involved in the angle of deflection value, $d\Delta r_i$ is calculated from the projections of the neighbouring voxels. The detail is presented in Eq. (8) below. For a voxel at a position (a,b,c) in the reconstructed volume,

$$d\Delta r_i^{(k)}(a,b,c) = -\frac{1}{2}(\Delta r_{xi}^{(k)}(a+1,b,c) - \Delta r_{xi}^{(k)}(a-1,b,c)) + (\Delta r_{yi}^{(k)}(a,b+1,c) - \Delta r_{yi}^{(k)}(a,b-1,c))$$
$$+ (\Delta r_{zi}^{(k)}(a,b,c+1) - \Delta r_{zi}^{(k)}(a,b,c-1)) \quad (8)$$

If conventional computer architecture was used to compute Eqs. (5) and (7), the reconstruction time would be huge since the large number of voxels of the reconstruction volume would be processed sequentially. Instead GPU architecture is used to speed up the calculation and exploits the full possibility of parallelization of the problem. In fact, ray trajectories are independent one from another, it is then possible to do simple operations on any of them simultaneously. GPU have a big number of processors and recently have opened a huge field of applications where data can be processed simultaneously. Eqs. (5) and (7) respectively denoted as the projection and backprojection operations, are especially suited to GPU architecture as it is presented in the figures below. Since gradient informations are required, more than one ray is needed in the backprojection step but for the sake of clarity only one is drawn.

The projection step (5)

For each projection, the direction of the ray issued from a pixel on the camera is computed knowing the camera calibration and model. The first intersection point with the volume is defined, then from there the projection measurement is calculated by accumulating the gradient information of each crossed voxel. This step is especially suited for the GPU architecture since several rays can be launched simultaneously.

Fig.1. Illustration of the projection step

The Backprojection step (7)

For each voxel in the object, pixels contributing to it are found by computing the intersection of the ray passing in this voxel and going towards the camera center with the image plane. Then the voxel value is updated according to the value of the hit pixels, or, in our case, according to an interpolation of the values of the neighbouring pixels. Once again, with a GPU architecture, several voxels can be updated simultaneously.

Fig. 2. Illustration of the backprojection step

2. 3. Regularization term minimization

While numerous regularization functions could be used we currently work in a quadratic regularization framework. Associated to a small cost of computation, this approach is known to smooth the results. The regularization function is expressed in Eq. (9),

$$R(v) = \left\| \nabla v \right\|^2 \quad (9)$$

With this function, the smoothing problem (VS2) is strictly convex, and its minimization is simply addressed with a steepest descent method:

$$v_m^{(k+1)} = v_m^{(k)} - \gamma(2\alpha\nabla.\nabla v_m^{(k)} + \beta(\rho_m^{(k+1)} - v_m^{(k)} - \lambda_m^{(k)})) \quad (10)$$

2. 4. Overall minimization

To complete the VS algorithm, the last equation is computed,

$$\lambda^{(k+1)} = \lambda^{(k)} - (\rho^{(k+1)} - v^{(k+1)}) \quad (11)$$

And the overall minimization process consists finally in computing Eq. (5), (6), (7), (10) and (11) sequentially until convergence. Convergence could be checked by monitoring the norm of the difference between two successive estimate of the density volume, but we usually prefer to use a fixed number of iterations, typically between 20 and 100 iterations.

It is important to note that, since only gradient informations contribute to the angle of deflection value, the mean value of the density is not observable. Moreover, gradient-based regularization terms like Eq. (9) do not constraint the mean value either. The mean value is then defined by its (arbitrary) initial value. In our simulation study, we choose the mean value of the synthetic volume. In real data processing, some guess of the mean density should be used.

3. RESULTS

3. 1. Simulation data

In order to study the quality of our reconstruction algorithm we made the choice of reconstructing a synthetic density field prior to use it in a real experimental context. Synthetic BOS images were then computed from a calculation of coplanar jet issued from the CoJen project [17]. Slices of the volume are presented in figure 3.
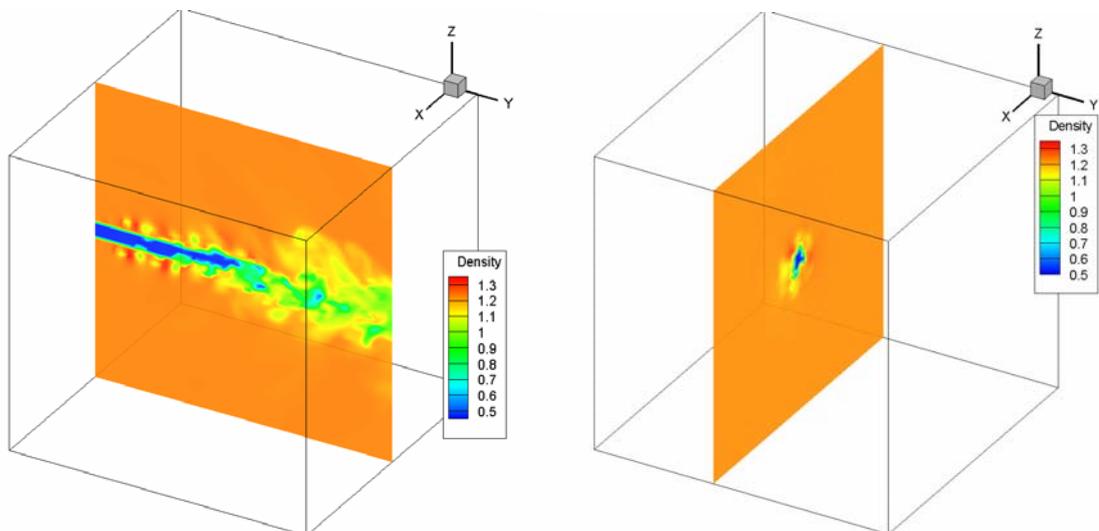
Fig. 3. Slices of the jet to be reconstructed

A pinhole camera model was then chosen to compute the ray directions. This model has the advantage of being in good agreement with camera real behaviour when distortions are removed with calibration. The direct problem strategy is taken from [10]: we solve the Eikonal equation using a Range Kutta 4 integration scheme to obtain the deflection angles in the three directions for each camera. Cameras are positioned all around the 2 meter side cube volume (cf. figure 4). Since the discretization varied from 80x80x80 to 240x240x240 voxels, the voxel size was either 25 mm or 8 mm.
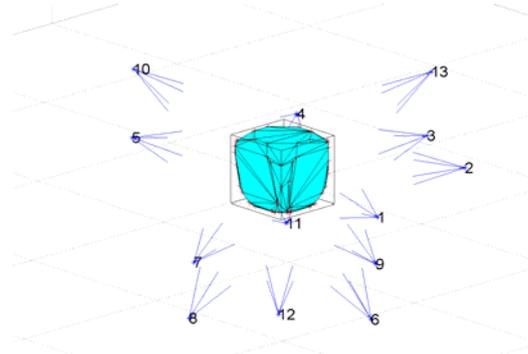
Fig. 4. Camera positions around the volume

In practice, deviations are measured by a window correlation process applied between a reference image of the background pattern and the deformed one. Full simulation of this step is very complicated, hence we have chosen to model the correlation error by an additive noise whose standard deviation, given in pixel, is chosen according to a reference displacement. The reference displacement of 1 pixel is associated to the mean density variation that we want to observe and to the geometric setting of the experiment. Two different noise levels are tested, one with a standard variation of 0.1 pixel and another one with a standard variation of 0.5 pixel. The associated projections are presented in figure 5. The first level of noise is relevant to the available software used to compute the displacement field. The second one seems rather noisy but is interesting to evaluate the robustness of our algorithm.

a   b
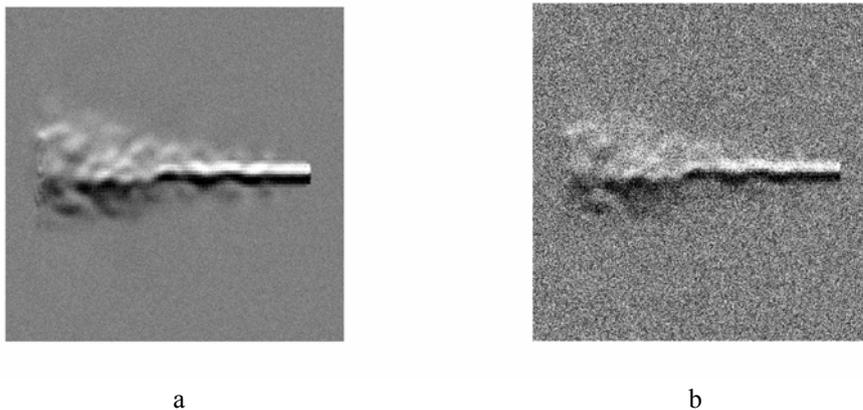
Fig. 5. Noisy projection (ε_x) STD error = 0.1 pix (a), STD Error = 0.5 pix (b)

3. 2. Reconstruction without regularization

The fully 3D characteristics of the jet need a lot of cameras to be captured and reconstructed correctly, figure 6 presents the reconstruction without noise with three different configurations: 6, 13 and 23 cameras.

a   b   c
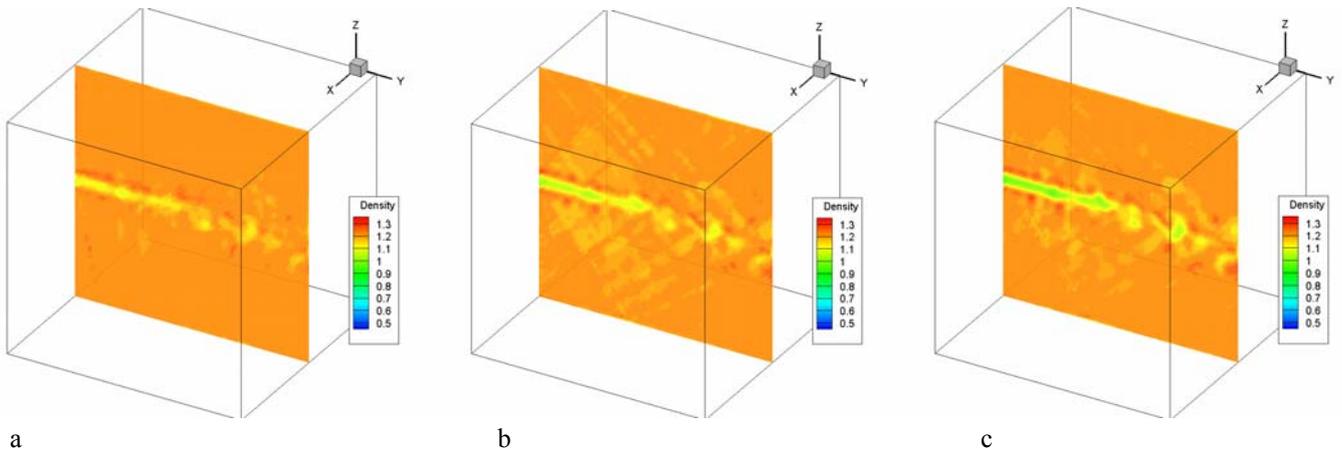
Fig. 6. Reconstruction with three configurations: 6 (a), 13 (b) and 23 (c) cameras

From this figure it appears that even in the 23 cameras configuration 3D features are not reconstructed. In fact, the information is spread along the rays and fails to focus completely in the centre of the jet. Numerous projections are then required. In order to reconstruct instantaneous fields it is necessary to have all these projections taken simultaneously, hence the need for many cameras. This approach does not seem appropriate to BOS experiments and we therefore need to reduce the number of required projection by adding some constraints on the problem. The simplest way to do that is to set a mask on the reconstructed volume according to the image support of the observed projections and to a model of the observed phenomenon. Here we use an approximation of a conic mask whose size is fitted on our projections. With this mask, the energy appears focused on the centre of the volume and the reconstruction with 13 cameras is already satisfactory (figure 7, the slices presented are the same than the initial volume).

Let us now consider data with added observation noise. First, we need to mention that the iterations of the steepest descent update (Eqs. (5-7)) are stopped before convergence. Indeed, it is well known that the exact minimum of a non regularized criterion is not relevant when the data are noisy. Moreover, the resulting solution mainly depends on the minimization process. As GPU tends to promote the use of extremely simple algorithm rather than refined one, we use the steepest descent with a fixed step. This approach suffers of the fact that when reaching the minimum solution it will step away from it if the step is too large. It is therefore useless to do many iterations.
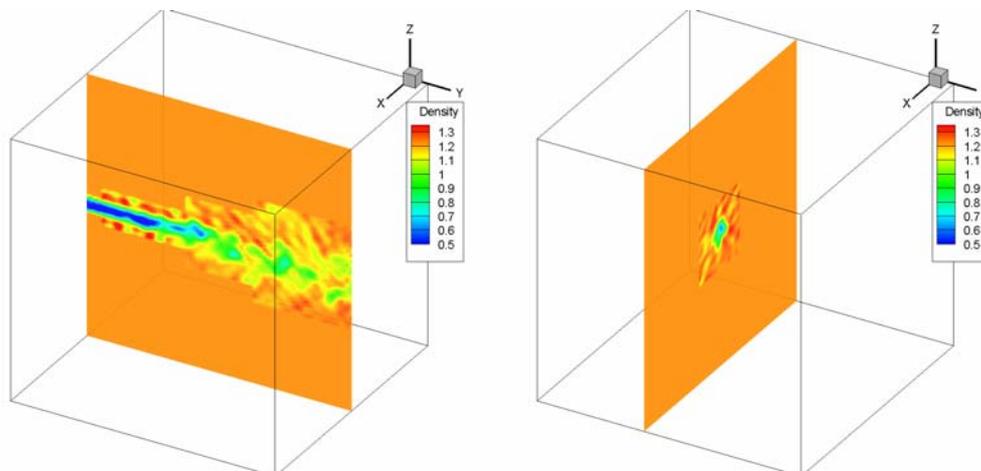
Fig. 7. Reconstruction with 13 cameras, no noise and adequate mask

a   b   c
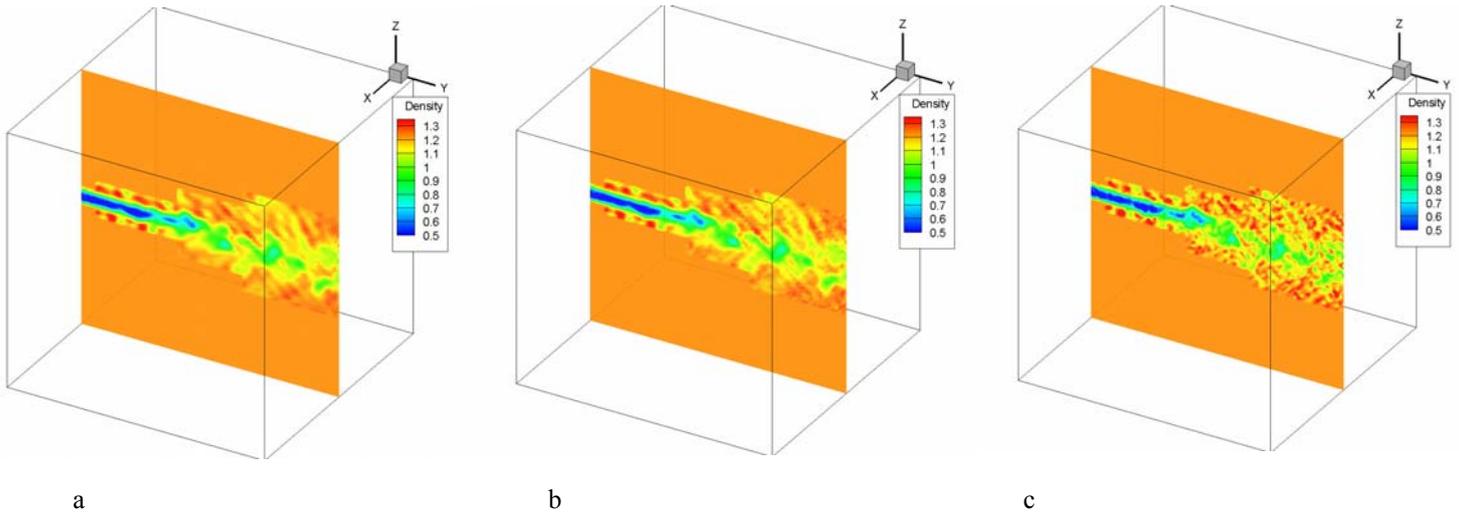
Fig. 8. Impact of the noise on the quality of the reconstruction, no noise (a), STD error = 0.1 pix (b), STD Error = 0.5 pix (c)

Noisy projections (cf. Section 3.1) are then given as inputs of our algorithm to evaluate its robustness. Reconstructions with no other regularization than the mask are presented in figure 8. The mask seems to be already a useful regularization tool since, with the first level of noise the reconstruction is almost not impacted, while, with the second level, the reconstruction tends to be more chaotic but is still recognizable.

Calculation time

Note that the ray tracing procedure which computes the synthetic projections (Section 3.1) is coded in C. While the calculation with the Runge-Kutta scheme is a little bit more complex than what is implemented in the CUDA code, it is still useful to consider it as a reference computation time. With the C code, the projection step takes around 5 minutes to be completed for one camera. The whole iteration (projection step, backprojection step and regularization for all cameras) takes only 10 seconds in CUDA for the 80x80x80 voxels volume on a NVIDIA GeForce GTX 460, which has 336 processor cores with 1350 MHz processor clock and 1GB GPU memory. The 240x240x240 volume iteration time is 30 seconds and 10 iterations were needed to obtain the results presented above.

3. 3. Reconstruction with quadratic regularization

If one wants to improve the results quality, regularization is needed. Quadratic regularization presented previously is first tested. As expected, the results are smoothed with the unwanted consequence to decrease the overall intensity of the signal as can be seen in figure 9.
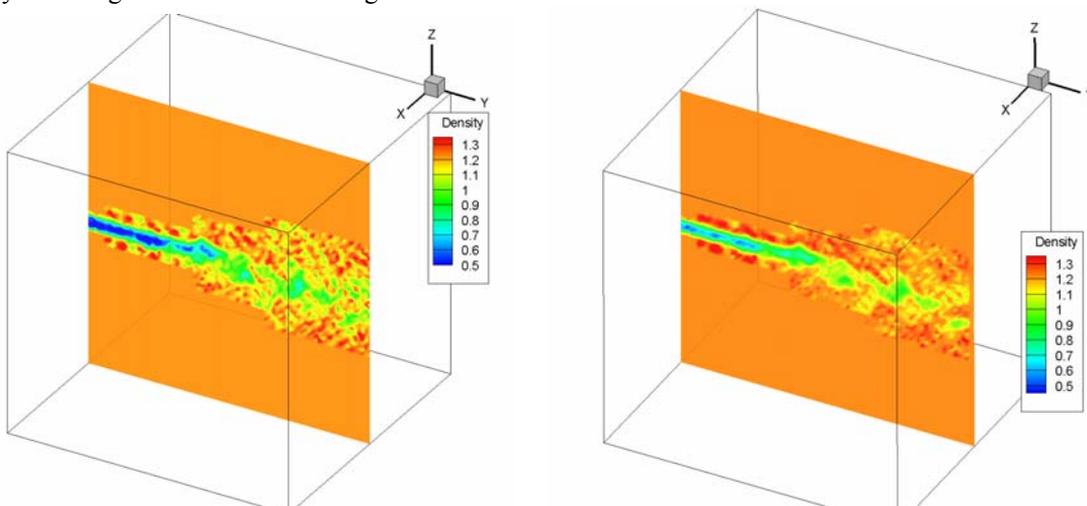
Fig.9. Impact of the regularization on the reconstruction, no regularization (a), quadratic regularization $\alpha=\beta=0.3$ (b)

Note that, in the regularized case, the algorithm is globally convergent and does not require to be stopped during the iterations. The regularization parameter value is chosen in an ad-hoc way, based on the calculation of the norm of the difference between the reconstructed volume and the true one. In this case, compared to the reconstruction with no noise presented in figure 7, the regularized reconstruction decreases the error by 50 % compared to the non-regularized reconstruction. Hence, while leading to oversmoothing, this approach improves the quality of the reconstruction.

4. CONCLUSION

Finally, we have seen that our reconstruction method is rather robust and with the required mask offers a satisfying quality of reconstruction. The quadratic regularization improves the result, but leads to oversmoothing. It is not perfectly suited to our purposes. This regularization does not succeed in reducing the oscillations due to the noise without decreasing the signal amplitude in the centre of the jet.

The choice of the regularization parameter is also an issue, in this case we were able to choose the regularization parameter by comparing the regularized solution with the true solution but in an experimental framework it will not be possible. For quadratic regularization, two methods, which are presented in [19], can determine the best value of the regularization parameter: Generalized Cross Validation (GCV) and the L-Curve method. These methods could be tested in future works. However, although satisfactory, our reconstruction algorithm reconstructs poorly some characteristics of the jet such as the strong gradient due to its linear behaviour. It is then needed to vary our regularization approach to be able to reconstruct all various particularities correctly.

Total variation regularization received a lot of attention particularly in image denoising [14] and recently in tomography [16] [18] [20]. This approach is really interesting for its efficiency in removing noise while preserving discontinuities. It may then help in focusing the energy of the signal inside of the jet. The regularization term is the following,

$$R(\rho) = \left| \nabla \rho \right| \quad (12)$$

Since this term is non differentiable in zero, its minimization requires non trivial strategies which will be implemented in the future in particular if we want to work with strong gradients such as shocks. In this case, the paraxial hypothesis which consists in approximating the ray trajectory with a straight ray might not be relevant and the non linear problem might be considered [10]. Hence the need for even more efficient minimization strategies.

Another branch for improvement of the algorithm lies in the selection of the minimization algorithm. Currently, a method of steepest descent with fixed step is used, while really cheap to compute, its global rate of convergence is also really slow. Steepest descent with recalculation of the step at each iteration or conjugate gradient methods could be an alternative. Yet in the GPU framework it might not be relevant to increase the complexity of the algorithm, hence its computation time per iteration, to decrease the overall number of iterations. The difference between the simple algorithm which requires a huge number of iterations and the complex one with a small number of iterations should be evaluated to ensure the shortest reconstruction time.

Finally, we have shown that iterative methods which are robust to incomplete and noisy data are to be considered with attention for the reconstruction of instantaneous fields in the BOS area. We focused on the practical aspect of the algorithm and provided a first simulation study. Our next step will be to evaluate it in a real experimental context.

References

1. G.E.A. Meier. *Computerized background-oriented schlieren.* Experiments in Fluids, (33) :181–187, 2002.
2. S. Raghunath, D.J. Mee, T. Roesgen, and P.A. Jacobs. *Visualization of supersonic flows in shock tunnels, using the background oriented schlieren technique.* In 2004 AIAA Australian Aerospace Student Conference, 2004.

3. Kolja Kindler, Erik Goldhahn, Friedrich Leopold, and Markus Raffel. *Recent developments in background oriented schlieren methods for rotor blade tip vortex measurements.* Experiments in Fluids, 43 :233–240, 2007.

4. E.D. Iffa, A.R.A. Aziz, and A.S. Malik. *Gas flame temperature measurement using background oriented schlieren.* Journal of Applied Sciences, 2011.

5. F. Champagnat, A. Plyer, G. Le Besnerais, B. Leclaire, S. Davoust, and Y. Le Sant. *Fast and accurate PIV computation using highly parallel iterative correlation maximization.* Experiments in Fluids, 2011, 50.

6. L. Venkatakrishnan and G.E.A. Meier. *Density measurements using the background oriented schlieren technique.* Experiments in Fluids, (37) :237–247, 2004.

7. L. Venkatakrishnan and P. Suriyanarayanan. *Density field of supersonic separated flow past an afterbody nozzle using tomographic reconstruction of bos data.* Experiments in Fluids, 47 :463–473, 2009.

8. A. C. Kak and Malcolm Slaney, *Principles of Computerized Tomographic Imaging,* IEEE Press, 1988.

9. Sourgen, F.; Leopold, F. and Klatt, D. *Reconstruction of the density field using the Colored Background Oriented Schlieren Technique (CBOS)* Optics and Lasers in Engineering, 50, 29-38, 2012.

10. I. Ihrke. *Reconstruction and Rendering of Time-Varying Natural Phenomena.* PhD thesis, Max-Planck-Institut für Informatik, 2007.

11. Ota, M.; Hamada, K.; Kato, H. and Maeno, K. *Computed-tomographic density measurement of supersonic flow field by colored-grid background oriented schlieren (CGBOS) technique* Measurement Science and Technology, 22, 104-111, 2011.

12. Atcheson, B.; Ihrke, I.; Heidrich, W.; Tevs, A.; Bradley, D.; Magnor, M. & Seidel, H.-P. *Time-resolved 3D Capture of Non-stationary Gas Flows* ACM Transactions on Graphics (Proc. SIGGRAPH Asia), 2008, 27, 132

13. J. Idier, *Regularization tools and models for image and signal reconstruction.* Proceedings of the 3rd International Conference on Inverse Problems in Engineering: Theory and Practice, Washington, USA, 1999

14. A. Chambolle. *An algorithm for total variation minimization and applications.* Journal of Mathematical Imaging and Vision, 20 :89–97, 2004.

15. Afonso, M. V. Bioucas-Dias, J. and Figueiredo, M. A. T. *Fast image recovery using variable splitting and constrained optimization* IEEE Transactions on Image Processing, 2010.

16. Y. Pan, R.T. Whitaker, A. Cheryauka, and D. Ferguson. *Regularized 3d iterative reconstruction on a mobile c-arm ct. In Proceedings of* the First CT meeting, Salt Lake City, 2010.

17. F. Vuillot, N. Lupoglazoff and G. Rahier, *Double stream nozzles flow and noise computations and comparisons to experiments.* 46 th AIAA Aerospace Sciences Meeting and Exhibit, 2008.

18. J. H. Joergensen, *Knowledge-based tomography algorithm.* Master of Science, Technical University of Denmark, 2009

19. J. Idier, Ed., *Bayesian Approach to Inverse Problems*, ISTE Ltd and John Wiley & Sons Inc, avr. 2008

20. G T Herman and R Davidi. *Image reconstruction from a small number of projections.* Inverse Problems, 24 (4), 2008.