# ROBUST REAL TIME MOTION COMPENSATION FOR INTRAOPERATIVE VIDEO PROCESSING DURING NEUROSURGERY

*M. Sdika* [1] *L. Alston*[1] *L. Mahieu-Williame*[1] *J. Guyotat*[2] *D. Rousseau*[1] *B. Montcel*[1]

[1] Université de Lyon, CREATIS; CNRS UMR5220;
Inserm U1044; INSA-Lyon; Université Lyon 1, France.
[2] Service de Neurochirurgie D; Hospices Civils de Lyon, Bron, France

## ABSTRACT

A motion compensation method dedicated to intraoperative RGB video imaging in neurosurgery is presented in this work. The dedicated motion model proposed is based on subspace learning of the patient brain motion. The resolution method uses keypoints for a sparse, fast and robust estimation of the brain motion. Our results, obtained from *in vivo* data, show that our method is as accurate as standard motion estimation method while being much faster. It is also very robust to unpredicted events that can happen in the operative room and opens the way to intraoperative real time hemodynamics map during neurosurgery on human subjects.

***Index Terms—*** motion compensation, image registration, learning, computer vision, neurosurgery, interventional image processing

## 1. INTRODUCTION

Brain intraoperative RGB video imaging is relevant in terms of interventionnal imaging in neurosurgery. It has been used for example on small animal to map hemoglobin changes on the cortex due to functional activation [1]. As far as we know, such interventional functional mapping has not been done yet on human in real time during the perioperative period. Indeed, beside the constraints due to experimentation in the operative room, brain motion occurs after opening the skull and the dura matter. These motions are mainly due to cardiac and breathing cycles, but also to any other motion of the patient or the operating table. This leads to the need of a dedicated real time motion compensation method for the problem of interventional RGB video imaging in neurosurgery.

Motion compensation relates to the vast field of image registration (see [2, 3] for recent reviews on these two subjects), and is most commonly addressed with optical flow equation. However, the specific application domain considered here has two important properties that can be advantageously used: the viewpoint changes are limited and the motion of the object recorded by the camera is repetitive. This leads us to consider subspace learning to provide priors to the transformation and to improve the computation speed.

Linear parameterization of the optical flow to model the transformation have been proposed in the context of mouth motion or articulated limb motion analysis [4]. The linear basis are either given or estimated from principal component analysis (PCA) and the warping coefficients are then estimated with a dense computationally expensive optimization procedure. In [5], PCA is also used to learn the transformation subspace, however, the PCA coefficients are estimated from only a sparse set of keypoints.

There are several contributions in this work. First, a transformation model adapted to the brain motion compensation problem in the context of neurosurgery is proposed. We also show that, the use of subspace learning and keypoints tracking is an effective method to solve this problem.Then an iterated re-weighted least square (IRLS) algorithm is proposed for a robust numerical resolution of the motion compensation. Compared to [5], the robust resolution stays at the keypoints level: the model validation does not go down at the pixel level and consequently, our method better scales with the image size. Our method allows compensating for the motion in real time; it is very robust and even allows the surgeon to move its tools in the field of view without losing the tracking.

## 2. METHODS

Two properties of the video obtained can be exploited to reduce the computation time and provides priors to the transformation model. First, the camera is always pointing at the same moving object, so if a viewpoint change occurs, it is limited. Second, as the brain deformation is repetitive, learning can be used to capture the main modes of the motion once it has been estimated on the first frames. Using these two remarks, a transformation model as well as a sparse and robust resolution method is proposed.

### 2.1. Motion Model

The transformation is decomposed into two components. The first one models the patient moves and slight motion of the operation table or the camera. It is global, large scale and unpredictable and is modeled as a time dependent affine trans-

978-1-4799-2349-6/16/$31.00 ©2016 IEEE

form. The second component, $T_d$, models the local nonrigid deformations of the brain:

$$T(x,t) = A(t)x + b(t) + T_d(x,t). \tag{1}$$

$T_d$ is mainly due to the cardiac pulsation and consequently, the same motion is repeated over time. A strict periodicity of the motion would be a too strong hypothesis as the patient cardiac pulse rate is not strictly constant over time. However, one can assume the deformation lies in a low dimensional affine space:

$$T_d(x,t) = T_\mu(x) + \sum_{k=1}^{K} \lambda_k(t)p_k(x). \tag{2}$$

$T_\mu$ is the average local deformation, $p_k$ are the stationary basis vectors of the deformation and $\lambda_k$ are the 2D time dependent deformation coefficients in this basis (product $\lambda_k(t)p_k(x)$ is the entrywise Hadamart product). If a composition of the deformable and affine part might be more appropriate, it would imply a nonlinear parameterization of the transformation. As we will see in the experiments, the additive model seems sufficient for our problem.

### 2.2. Learning the Brain Deformation Basis

The $p_k$ basis vectors must be learned for each new experiment. This basis indeed depends on the camera positioning, the patient specific anatomy and the brain deformation pattern. Consequently, the first $N$ frames are dedicated to the learning of this basis: for $i \leq N$, the transformation $T(x,t_i)$ between the initial and the $i^{th}$ frame is estimated from any standard motion estimation routine. The affine part $A(t_i), b(t_i)$ is then estimated as the solution of the following linear least square problem:

$$\min_{A(t_i),b(t_i)} \sum_x \|A(t_i)x + b(t_i) - T(x,t_i)\|_2^2 . \tag{3}$$

Local brain deformations in the learning frames are then obtained by subtraction:

$$T_d(x,t_i) = T(x,t_i) - A(t_i)x - b(t_i).$$

A PCA is finally run to estimate the low dimensional affine space where the local deformation components lie while correctly capturing the variability: $T_\mu$ is the average of $(T_d(x,t_i))_{i\in[1,N]}$ and the basis vectors $p_k$ are the $K$ first eigenvectors of their covariance matrix.

### 2.3. Motion Estimation

Once the brain deformation basis is known, the motion between the initial and the current frame is given by the time dependent parameters: $A(t)$, $b(t)$ and $\lambda_k(t)$. The number of parameters to estimate is reduced from twice the number of pixels to $2K + 6$. To estimate such a low number of parameters, it is not necessary to use all the pixels. We select and track a sparse set of $L$ keypoints $(x_l)_{l\in[1,L]}$ and use them to recover the whole transformation field. The keypoints are chosen as the well-known Harris keypoints [6] on the initial frame. Using the algorithm described in [7] the transformation $T(x_l,t)$ can be estimated on these pixels only.

The linearity of the equation

$$T(x_l,t) = A(t)x_l + b(t) + T_\mu(x_l) + \sum_{k=1}^{K} \lambda_k(t)p_k(x_l) \tag{4}$$

with respect to $A(t)$, $b(t)$ and $\lambda_k(t)$ now allows to estimate these parameters from a simple linear least square fit.

Most of the time, this least square fit is sufficient to accurately estimate the motion. However, a more robust fit procedure is necessary when some keypoints violate the low dimensional transformation model. This is the case if $T(x_l,t)$ is not correctly estimated for some keypoints and happens in practice if, for example, a surgical tool appears in the field of view. In this case, we found that IRLS can be used to robustly find $A(t)$, $b(t)$ and $\lambda_k(t)$: after a first unweighted least square fit, parameters are iteratively estimated from weighted fits: the weight for $x_l$ is $g_\sigma(r_l)$ where $r_l$ is the current residual and $g_\sigma$ a Gaussian with a standard deviation $\sigma$ which is halved at each iteration.

To summarize, once the learning of the $p_k$ basis is done, the transformation between the initial and the current frame is estimated with the following algorithm:

- estimate $T(x_l,t)$ on the $L$ keypoints using the sparse Lucas and Kanade method [7]
- find $A(t)$, $b(t)$ and $\lambda_k(t)$ using IRLS on the problem 4
- compute $T(x,t)$ using the equation 1 and 2.

The complexity of the registration algorithm is then: $L$ keypoints transformation estimations for the first step and few linear system solving on a $(K+3) \times L$ matrices for the second step. This is very low when compared to the application of a standard registration or motion estimation method which involves the iterative resolution of a nonlinear problem with a cost function using all the pixels in the image.

## 3. NUMERICAL EXPERIMENTS

Several criteria have been used to assess the performance of our method. First, the affine model has been validated by comparing the transformations obtained from a standard optical flow motion estimation and our learning based low dimensional model. We also measure the temporal standard deviation images of the videos. The robustness of our method is finally assessed visually.

The proposed methods are denoted as ULS when unweighted least square fit is used and IRLS when the fit is done with iterated re-weighted least square. $N$ and $K$ should be large enough to sample the transformation space correctly and capture its variability. However too large values do not improve the results while increasing the CPU time. They were set to $N = 25$ and $K = 4$. Few hundreds keypoints were detected for each video.

The Farnebäck optical flow routine (GF) described in [8] has been used for comparison and for the learning step of both ULS and IRLS.

Seven videos from three patients has been used in the experiments. Each video lasts between 30 and 90 seconds, has a frame rate of 25 fps and a frame size of either $511 \times 388$ or $720 \times 576$. On the last video (denoted as V7), the surgeon placed a surgical tool in the field of view to evaluate the robustness of the method.

### 3.1. Evaluation of the transformation model

To evaluate our low dimensional transformation model, the transformation $T$ obtained with our method is compared to the transformation $T_{\mathrm{GF}}$ obtained using GF. The difference image $D$ between the two results is measured as:

$$D(x) = \frac{1}{n} \sum_{t} \| T_{\mathrm{GF}}(x,t) - T(x,t) \| ,$$

where $n$ is the number of frames in the video. This measure is an indication that our model is adequate for our problem and that we are able to obtain similar transformations as the ones obtained by algorithms with much more degree of freedom.

| | V1 | V2 | V3 | V4 | V5 | V6 |
|---|---|---|---|---|---|---|
| max | 2.01 | 2.46 | 1.92 | 4.63 | 6.95 | 9.34 |
| avg | 0.09 | 0.34 | 0.35 | 1.51 | 0.50 | 0.68 |

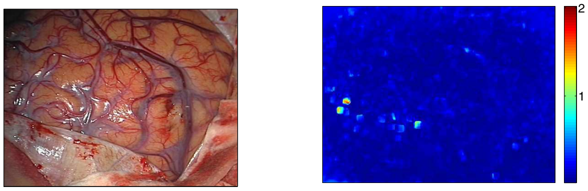
**Table 1**. Maximum and average difference in video V1 to V6.

In the table 1, the maximum and average of $D$ are given for videos V1 to V6 when IRLS is used. As visible, the average difference is subpixel for all the video. The maximum difference is also very low for V1-3. It seems somewhat important for V4-6 but one has to remember that in uniform regions and regions where the intensity gets saturated, the motion is not reliably estimated even for standard algorithms.

The difference image of V1 is shown on figure 1. For the most part of the image, the mean difference is subpixel. The bright spots in difference images are usually located where the intensity gets saturated during the course of the video.

### 3.2. Intensity Variation Based Validation

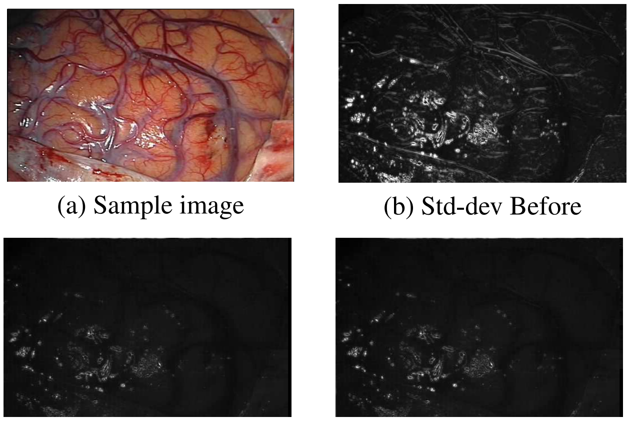
In this section, the temporal standard deviation (TSD) averaged on the three RGB channels of the videos is used to assess that the registration went well. TSD images have been inspected for the video V1 to V6 and they are all similar. In figure 2, TSD images for V1 is presented before motion compensation, using GF and using IRLS. The intensity scaling is the same for the three TSD images.



(a) Sample image

(b) Difference image $D$

**Fig. 1**. Difference image in pixel between the motion estimated using GF and our method.



(a) Sample image

(b) Std-dev Before

(c) Std-dev After, GF

(d) Std-dev After, Our Method

**Fig. 2**. Intensity temporal standard deviation of the recorded video before and after motion compensation using standard optical flow routine and IRLS.

The average TSD was reduced from 15% to 40% depending on the video with a slightly higher reduction for GF.

As expected, the motion compensation reduces the global range of the TSD images. It also removes brain structures visible on the TSD image of the raw video, meaning the alignment is correct in regions with high spatial gradients. Results obtained with GF and IRLS are visually similar. The main remaining bright spots are located for both methods where the intensity gets saturated during the video. This may be due to an incorrect motion estimation but also to the saturation itself.

### 3.3. Robustness

The robustness of the method is assessed using the V7 video on which a surgical tool is moved in the field of view during the recording. Snapshots of motion compensated videos are presented on figure 3.

As one can see, the presence of the tools strongly affects the motion compensation with GF or ULS, resulting in highly artefacted videos. Contrarily, the stability of the IRLS fit results in a visually correct video.

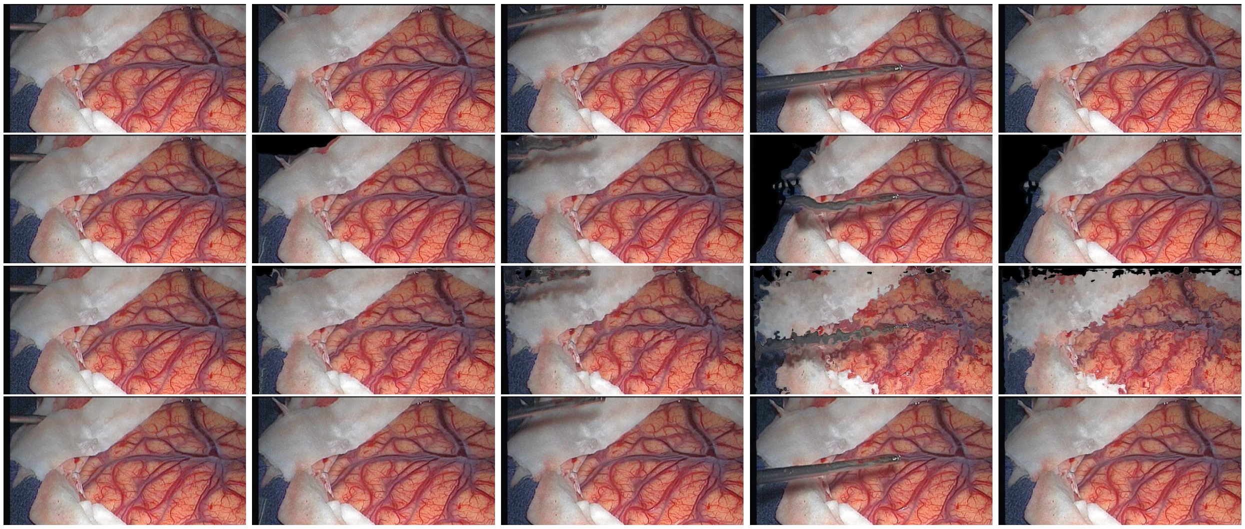


**Fig. 3**. Snapshots of the V7 video at several timepoints (in column). In line are presented, from top to bottom, the raw video and motion compensated video using GF, ULS and IRLS.

### 3.4. CPU Time Consideration

| Frame size | GF | ULS | IRLS |
|---|---|---|---|
| $511 \times 388$ | 0.2 | 0.01 | 0.02 |
| $720 \times 576$ | 0.45 | 0.04 | 0.09 |

**Table 2**. CPU time per frame in seconds (fps). Real time constraints is satisfied when fps $< 0.04$s.

CPU time measurements using a single core of an Intel Xeon E5620 2.40GHz are reported on table 2. Compared to GF, the reduction achieved by both ULS and IRLS is very important. ULS is always compatible with the real time constraint. For IRLS, the real time constraint is satisfied for moderate frame size and can easily be satisfied with the use of multiple cores for large frame size.

## 4. CONCLUSION

In this work, a motion compensation method dedicated to intraoperative video processing has been proposed. Compared to standard methods, the complexity is reduced by reducing the number of unknowns (from twice the number of pixels to $2K + 6$), the cost function complexity (from the number of pixels to $L$) and the nature of the resolution (from a nonlinear problem to few linear system resolutions). Our IRLS method also enables robust motion compensation even when unpredicted events breaking the optical flow brightness constancy assumption occur during the course of the video. Our results show that robust and real time motion compensation can be achieved for neurosurgery applications.

## 5. REFERENCES

[1] A. Steimers, M. Gramer, B. Ebert, M. Fchtemeier, G. Royl, C. Leithner, J. P. Dreier, U. Lindauer, and M. Kohl-Bareis, "Imaging of cortical haemoglobin concentration with RGB reflectometry," in *Proc. SPIE*, 2009, vol. 7368.

[2] D. Fortun, P. Bouthemy, and C. Kervrann, "Optical flow modeling and computation: A survey," *Computer Vision and Image Understanding*, vol. 134, pp. 1 – 21, 2015.

[3] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *Medical Imaging, IEEE Trans. on*, vol. 32, no. 7, pp. 1153–1190, 2013.

[4] D. J. Fleet, M. J. Black, Y. Yacoob, and A. D. Jepson, "Design and use of linear models for image motion analysis," *International Journal of Computer Vision*, vol. 36, no. 3, pp. 171–193, 2000.

[5] R. Roberts, C. Potthast, and F. Dellaert, "Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies," in *Computer Vision and Pattern Recognition*, 2009, pp. 57–64.

[6] C. Harris and M. Stephens, "A combined corner and edge detector.," in *Alvey vision conf.*, 1988, vol. 15, p. 50.

[7] J.Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, pp. 1–10, 2001.

[8] Gunnar Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, vol. 2749 of *Lect. Notes in Comp. Sci.*, pp. 363–370. 2003.