To appear in the International Journal of Robotics Research,
special issue on Medical Robotics, 2009

Real-time motion stabilization with B-mode ultrasound using image speckle information and visual servoing

Alexandre Krupa[1], Gabor Fichtinger[2,3] and Gregory D. Hager[3]

[1] IRISA, INRIA Rennes-Bretagne Atlantique, Lagadic, F-35042 Rennes, France
E-Mail: alexandre.krupa@irisa.fr *
[2] School of Computing, Queen's University, Kingston, Canada
E-Mail: gabor@cs.queensu.ca
[3] Engineering Research Center, Johns Hopkins University, Baltimore, MD 21218, USA
E-Mail: hager@cs.jhu.edu

Abstract

We develop visual servo control to stabilize the image of moving soft tissue in B-mode ultrasound (US) imaging. We define the target region in a B-mode US image, and automatically control a robot to manipulate an US probe by minimizing the difference between the target and the most recently acquired US image. We exploit tissue speckle information to compute the relative pose between the probe and the target region. In-plane motion is handled by image region tracking and out-of-plane motion recovered by speckle tracking using speckle decorrelation. A visual servo control scheme is then applied to manipulate the US probe to stabilize the target region in the live US image. In a first experiment involving only translational motion, an US phantom was moved by one robot while stabilizing the target with a second robot holding the US probe. In a second experiment, large 6-DOF motions were manually applied to an US phantom while a 6-DOF medical robot was automatically controlled to compensate for the probe displacement. The obtained results support the hypothesis that automated motion stabilization shows promise for a variety of US-guided medical procedures such as prostate cancer brachytherapy.

1 Introduction

Quantitative ultrasound (US) guidance has great potential in supporting a wide range of diagnostic procedures and minimally invasive interventions. However, one of the barriers to wider application is the challenge of locating and maintaining targets of interest within the US scan-plane, particularly when the underlying tissue is in motion. Conventional wisdom might suggest that this problem could be effectively solved by applying known motion tracking techniques to 3D US images. However, current 3D US systems are prohibitively expensive, suffer from low voxel resolution, and, most importantly, they do not provide each access to real-time volumetric data stream to the user. Specialized hardware and privileged access is required to accomodate the huge volume of B-mode image data delivered by such systems, and accessing

*Corresponding author is Alexandre Krupa (alexandre.krupa@irisa.fr).

the raw RF signal volume in real-time is difficult with today's technology. However real-time access to the data stream is crucial for applications that control a robot directly from ultrasound images. In addition, tapping into the internal data stream falls outside the scope of current regulatory approvals of the US machines, which creates regulatory issues in scanning human subjects, even in laboratory setting.

A more practical approach is to achieve target tracking and stabilization with conventional 2D B-mode US imaging systems which are readily available in most clinics. Given the prevalence of conventional 2D US, a workable method operating on 2D US images could be exploited in a host of clinical applications. For example, in diagnostic US imaging, one could automatically move the US probe to maintain the optimal view of moving soft tissue targets. Or, in biopsies and localized therapy procedures, one could synchronize the insertion of needles or other surgical tools into a moving target observed in live US.

Although full 6-DOF ultrasound motion tracking and robotic image stabilization seems to lend itself to a wide spectrum of ultrasound-guided diagnostic and interventional procedures, introduction of an autonomous US probe manipulation robot into many of these procedures will represent major departure from current clinical practice. Therefore, it seems prudent to adapt robotic image stabilization first to a procedure where constrained mechanical US probe motion is part of standard practice, and motorizing the probe's motion will not create any new clinical hazards.

We have identified prostate cancer brachytherapy as one such pilot clinical application. The prostate is a walnut size organ situated in the pelvic floor, adjacent to the rectum. Prostate brachytherapy entails implanting radioactive pellets the size of a rice grain into the prostate through the perineum. This is performed under live transrectal ultrasound (TRUS) imaging guidance [Wallner *et al.*, 2001]. The radioactive pellets kill cancer by emitting radiation. A typical brachytherapy procedure requires the insertion of 20-40 needles, the actual number of needles depending on the size of the prostate. Penetration by the needle causes often severe dislocation, rotation and deformation of the prostate. The scanning motion of the TRUS probe has similar effects, though to a lesser degree, as the probe deforms the prostate gland through the rectum wall. As a result, in the TRUS image it is not unusual to lose sight of the target when the needle is being observed or to lose sight of the needle when the target is being observed. Worse yet, the target location is seldom characterized by any visible anatomical feature.

Since the desired target is invisible to the naked eye in B-mode ultrasound, US speckle based tracking methods are an appealing approach to synchronize the motion of the probe with the motion of the target. As described in [Wallner *et al.*, 2001], the TRUS probe is already mounted on a movable structure (called a probe stepper) that allows the physician to translate the probe inside the rectum and to rotate the probe about the axis of translation. Automated target tracking would allow us to automatically modify the probe's position with respect to the prostate through robotized motion of the probe controlled based on the ultrasound image. The modifications necessary to accomplish this are described in Section 6. In short, brachytherapy can significantly benefit from ultrasound-based motion tracking and robotic image stabilization, and this approach does not represent major departure from current clinical hardware and workflow. Thus, the transition to clinical trials can be achieved relatively quickly.

Over the past several years, a sizable body of research has been dedicated to US imaging in conjunction with medical robots for the purposes of image acquisition. For example, in [Pierrot *et al.*, 1999] a robotic system has been developed to automatically perform 3D US acquisition of cardiovascular pathologies by moving a 2D probe along a given trajectory. In [Martinelli *et al.*, 2007] a teleoperated master/slave is used to perform remote ultrasound examination in order to detect abdominal aortic and iliac aneurysms.

The use of the US imaging information in robot control has received much less attention. In [Abolmaesumi *et al.*, 2002], visual servoing was used for automatic centering of the aorta artery section in the observed ultrasound image in order to maintain it visible during a three-dimensional robotized ultrasound scan. In this

work, the 3 in-plane motions (2 translations and 1 rotation) of the probe were controlled directly from 2D visual features extracted after a 2D segmentation of the section image. The remaining 3 out-of-plane motions (1 translation and 2 rotations) were teleoperated by the user. However, no solution was proposed to control the out-of-plane motions of the 2D probe by visual servoing. In [Hong *et al.*, 2004], the authors present a robotic system including a motionless US probe and a 2 degree of freedom (DOF) needle manipulator. Automatic needle insertion into a soft sponge phantom was performed using US image-based visual servoing. However, in this work, the actuated needle had to lie in the US observation plane, as only 2 DOF inside the observation plane were controlled. In general, a conventional US probe provides a 2D B-scan image which therefore limits vision-based control to the 3 DOF contained in the plane (two translations, one rotation) using classic visual servoing techniques. In [Stoll *et al.*, 2006], a surgical instrument was positioned under 3D US visual servoing, but as we pointed out earlier, 3D US guidance for real-time applications is limited by a variety of commercial and regulatory considerations.

There are some recent studies that have investigated controlling degrees of freedom outside the US observation plane. In [Vitrani *et al.*, 2005], 4 DOF were controlled by visual servoing in order to automatically position a robotized laparoscopic instrument. In [Bachta and Krupa, 2006], a visual servoing technique was used to control 6-DOF motion of the US probe to target a targeted section of a tumor. These methods, however, depended on geometrical models of the objects of interests, i.e. the tool forceps in [Vitrani *et al.*, 2005] and a pre-operative tumor model in [Bachta and Krupa, 2006], as well as on extensive image processing to segment the objects in B-mode US images.

Our stabilization methods rely heavily on the properties of ultrasound speckle. Traditionally, ultrasound speckle has been considered to be noise, and much effort has been devoted to eliminating or reducing speckle in US images. Speckle, however, is not random noise. It results from coherent reflection of very small cells contained in soft tissue. As a result, it is spatially coherent and remains highly correlated over small motions of the US probe. In practice, focusing of the US beam is never perfect, especially in the elevation direction, i.e, orthogonal to the imaging plane, and so the US beam has a thickness of several millimeters. Thus, for small motions of the US probe, consecutive beams overlap in space. Perfect, or "fully developed," speckle created by the region of tissue in the intersection of two beams appears to be fixed in space. In principle, it follows that just three regions of perfect speckle are sufficient to locate the full 6 DOF pose of the US beam relative to the tissue. Unfortunately, in biological tissue speckle is seldom perfect and it is further diminished during the formation of B-mode images. Nonetheless, as we show below, B-mode images still possess enough coherence that we can exploit it to recover the full 6-DOF relative pose of B-mode US images, even in the elevation direction.

In prior work, speckle information was used to estimate multi-dimensional flow of 2D US image ([Bohs *et al.*, 2000]). Recently several authors ([Gee *et al.*, 2006; Chang *et al.*, 2003]) have published speckle decorrelation techniques for performing freehand 3D ultrasound imaging without the need of a position sensor to provide the location of the 2D US probe. A probalistic framework was also proposed in [Laporte and Arbel, 2007] to estimate elevational separation between US images over large image sequences from speckle information. These techniques depend on experimental pre-calibration of speckle decorrelation curves in real soft tissues and/or speckle mimicking phantoms. In [Boctor *et al.*, 2005], a method using speckle tracking was used for real-time intra-operative calibration of a tracked 2D B-mode probe used in image guided surgery applications. Speckle correlation is also widely used in sonoelastography imaging, to estimate the displacement field of biological scatterers caused by physical pressure [Boctor *et al.*, 2006].

In contrast to the motion tracking methods enumerated above, we present a method for fully automatic, real-time tracking and motion compensation of a moving soft tissue target, using a sequence of 2D B-mode US images. We track both in-plane and out-of plane motions by making direct use of the speckle information

contained in the US images. This is fundamentally different from prior techniques that relied on segmenting structures of interest, such as in [Abolmaesumi *et al.*, 2002; Hong *et al.*, 2004]. Much abridged descriptions of particular aspects of this project have appeared in [Krupa *et al.*, 2007*a*] and [Krupa *et al.*, 2007*b*]. Here we provide a wider survey of prior art, in-depth description of the tracking method, and extensive simulation and experimental results accompanied by an in-depth discussion and analysis.

The remainder of this paper is organized as follows: Section 2 presents the overall tracking problem and the motion decomposition we use to describe the full motion of the soft tissue target. Sections 2.1 and 2.2 present respectively the methods used to extract the in-plane and out-of-plane motion of the target B-scan image. A hybrid servo control approach is developed in section 3 to control the displacement of an US probe held by a robot in order to stabilize a moving B-scan target of soft tissue. Results obtained from simulations and a ex-vivo experiments are then presented and discussed in sections 4 and 5.

2 Motion estimation

Our problem is to control the motion of an US probe so as to minimize the relative offset between the observed B-scan denoted by a Cartesian frame $\{p\}$ and a target B-scan denoted by a Cartesian frame $\{t\}$. Since this relative offset will be close to zero during the active stabilisation process that we present in the next of this paper, we propose to approximate the 6 DOF target plane pose relative to the probe from the combination of two homogeneous transformations: ${}^{p}\mathbf{H}_t \approx {}^{p}\mathbf{H}_c \, {}^{c}\mathbf{H}_t$ where ${}^{p}\mathbf{H}_c$ and ${}^{c}\mathbf{H}_t$ describing the in-plane and out-of-plane displacement of the target, respectively as illustrated in Fig. 1.

Note that $\{c\}$ corresponds to the Cartesian frame attached to an intermediate "virtual" plane.
The in-plane displacement is described by the translations t_x and t_y along the X and Y axes of the observed B-scan plane $\{p\}$ and the angular rotation γ around the Z axis (orthogonal to the image), such that:

$$ {}^{p}\mathbf{H}_c = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & t_x \\ \sin(\gamma) & \cos(\gamma) & 0 & t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1} $$

We define the relative displacement caused by out-of-plane motion as an elevation of distance t_z along the Z axis of $\{c\}$ and two successive rotations α and β around the Y and X axes of $\{c\}$. This yields the following homogeneous transformation matrix between $\{c\}$ and $\{t\}$:

$$ {}^{c}\mathbf{H}_t = \begin{pmatrix} \cos(\alpha) & \cos(\alpha)\sin(\beta) & \sin(\alpha)\cos(\beta) & 0 \\ 0 & \cos(\beta) & -\sin(\beta) & 0 \\ -\sin(\alpha) & \cos(\alpha)\sin(\beta) & \cos(\alpha)\cos(\beta) & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2} $$

2.1 In-plane motion estimation

Fig. 2 shows the target image captured at time $t_0 = 0$ and an image obtained at a later time t after in-plane motion was applied. To extract the in-plane rigid motion between the two images, we use the image region tracking technique presented in [Hager and Belhumeur, 1998] which we briefly recall here.

The objective of this technique is to estimate the parameter vector μ of an appropriate parametric model function $f(\mathbf{x};\mu)$ which describes the geometrical transformation on the pixel coordinates $\mathbf{x} = (x \; y)^T$ from the reference to the observed image. For in-plane rigid displacement, the motion parameter vector is

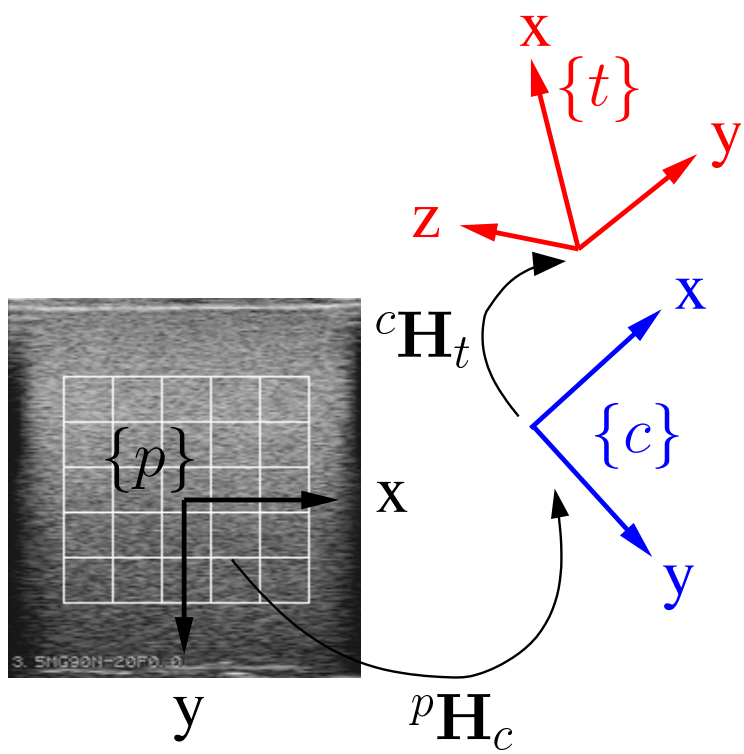

Figure 1: Decomposition of the target plane position by successive in-plane and out-of-plane homogeneous transformations

$\mu = (u_x\ u_y\ \gamma)^T$ where u_x, u_y are the pixel translations along X and Y axes of the reference image and γ is the rotation angle around the Z axis. Note that u_x and u_y are related to t_x and t_y by:

$$\begin{aligned} t_x &= u_x s_x \\ t_y &= u_y s_y \end{aligned} \tag{3}$$

where s_x and s_y are respectively the width and height of a pixel.
The vector form of the motion parametric model function is:

$$f(\mathbf{x}; u_x, u_y, \gamma) = \mathbf{R}(\gamma)\mathbf{x} + \mathbf{u} \tag{4}$$

where $\mathbf{R}(\gamma)$ is the 2×2 rotation matrix of angle γ and $\mathbf{u} = (u_x\ u_y)^T$ is the translation vector. The principle of the motion tracking method is to compute the motion parameter μ that minimizes the sum of squared differences of pixel intensities between the region of interest (obtained with the geometrical transformation (4) in the observed image) and the reference region of interest (fixed in the target image where $\mu = 0$.) Therefore the objective function to minimize is the following:

$$\mathcal{O}(\mu) = ||\mathbf{I}(\mu, t) - \mathbf{I}(0, t_0)||^2 \tag{5}$$

where $\mathbf{I}(0, t_0)$ is the vector containing the intensity values of the N pixels belonging to the reference target image at $t = 0$ and $\mathbf{I}(\mu, t)$ contains the intensity values of the N pixels in the image acquired at time

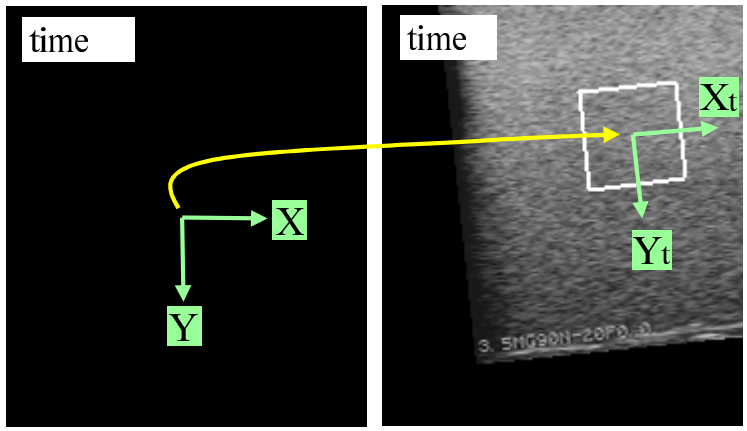

Figure 2: (Left) The reference image acquired at time $t_0 = 0$ with the region of interest to track. (Right) The observed image modified by the in-plane motion $f(\mathbf{x};\mu)$ with the estimated region of interest

t after resampling (warping) according to (4) using the most recent motion parameter $\mu(t)$ as given here:

$$\mathbf{I}(\mu,t) = \begin{bmatrix} I(f(\mathbf{x}_1,\mu),t) \\ \vdots \\ I(f(\mathbf{x}_N,\mu),t) \end{bmatrix} \tag{6}$$

By rewriting (5) in term of a vector of offsets $\delta\mu$ such that $\mu(t+\tau) = \mu(t) + \delta\mu$ from an image captured at time $t+\tau$:

$$\mathcal{O}(\delta\mu) = ||\mathbf{I}(\mu+\delta\mu, t+\tau) - \mathbf{I}(0,t_0)||^2 \tag{7}$$

and approximating it with a first order Taylor expansion, we obtain:

$$\mathcal{O}(\delta\mu) \approx ||\mathbf{M}\delta\mu + \mathbf{I}(\mu,t+\tau) - \mathbf{I}(0,t_0)||^2 \tag{8}$$

where $\mathbf{M}$ is the Jacobian matrix of $\mathbf{I}$ with respect to μ:

$$\mathbf{M}(\mu) = \begin{bmatrix} \nabla_{\mathbf{x}} I(\mathbf{x}_1,t_0)^T f_{\mathbf{x}}(\mathbf{x}_1,\mu)^{-1} f_\mu(\mathbf{x}_1,\mu) \\ \vdots \\ \nabla_{\mathbf{x}} I(\mathbf{x}_N,t_0)^T f_{\mathbf{x}}(\mathbf{x}_N,\mu)^{-1} f_\mu(\mathbf{x}_N,\mu) \end{bmatrix} \tag{9}$$

Here $\nabla_{\mathbf{x}} I(\mathbf{x},t_0)^T$ is the intensity gradient vector at pixel location $\mathbf{x} = (x\ y)^T$ in the target image and $f_{\mathbf{x}}$, f_μ are respectively the partial derivatives of $f(\mathbf{x};\mu)$ with respect to $\mathbf{x}$ and μ. By using $\mu = (u_x\ u_y\ \gamma)^T$ and the parametric motion model (4) we have:

$$f_{\mathbf{x}}^{-1} f_\mu = \begin{bmatrix} 1 & 0 & -y \\ 0 & 1 & x \end{bmatrix} \begin{bmatrix} \mathbf{R}(-\gamma) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \tag{10}$$

The solution of $\delta\mu$ is then obtained by setting the gradient of $\mathcal{O}(\delta\mu)$ to zero and solving which yields:

$$\delta\mu = -\mathbf{M}^+(\mathbf{I}(\mu,t+\tau) - \mathbf{I}(0,t_0)) \tag{11}$$

where $\mathbf{M}^+$ is the pseudo inverse of $\mathbf{M}$. The motion parameter vector is then:

$$\mu(t+\tau) = \mu(t) + \delta\mu \tag{12}$$

In practice, in order to obtain adequate convergence, we successively compute (11) and (12) during several iterations until $||\delta\mu||^2$ becomes lower than a small fixed threshold value ϵ. For more complete details on this method we invite the reader to refer to [Hager and Belhumeur, 1998].

Other methods based on the same principle are proposed in the literature, for example in [Benhimane and Malis, 2004] a second-order minimization technique is used for large motion tracking with fast convergence rate by using the mean value of the Jacobian $\mathbf{M}$ in the target image and the one in the observed image. An unifying framework is also presented in [Baker and Matthews, 2004] to compares the different approaches.

2.2 Out-of-plane motion estimation

We estimate the out-of-plane motion of the target US image plane $\{t\}$ with respect to the intermediate "virtual" plane $\{c\}$ obtained after applying the estimated in-plane motion transformation. The principle is to first use a speckle decorrelation technique to estimate the elevation distance of a grid of n patches that were fixed on the target image at time $t_0 = 0$, and then to fit a plane to this data.

2.2.1 Speckle decorrelation technique

An approximation of the speckle correlation function as a function of the orthogonal distance d between two B-mode scans $\mathbf{I}_1$ and $\mathbf{I}_2$ is given in [Gee *et al.*, 2006] by the Gaussian model function:

$$\rho(\mathbf{I}_1, \mathbf{I}_2) = \exp\left(\frac{-d^2}{2\sigma^2}\right) \tag{13}$$

where ρ is the correlation value of speckle included in two corresponding patches in the two images and σ is the resolution cell width along the elevation direction. In practice, this approximation works well when the grey level intensity of the image is defined on a linear scale. This is the case when we directly use the radio-frequency (RF) signal provided by the ultrasound imaging device. Unfortunately, this signal is not generally available on most standard US systems. Instead, the RF data is processed into B-mode images with intensity compressed on a logarithmic scale. As we deal with B-mode images, we first convert the intensity back to a linear scale by applying the relation given in [Smith and Fenster, 2000]:

$$I(i,j) = 10^{P(i,j)/51} \tag{14}$$

where $I(i,j)$ is the decompressed grey level intensity of the pixel located at image coordinates i, j and $P(i,j)$ is the measured intensity in the B-mode image.

In order to perform position estimation using decorrelation, it is necessary to experimentally calibrate speckle decorrelation curves from real soft tissues or from an ultrasound phantom simulating speckle. These curves are obtained by capturing a set of B-scan images at known distances along the elevation direction and measuring the normalized correlation coefficients $\rho(d)$. Let I_0, I_d correspond, respectively, to the pixel intensity array of a given patch of the B-scan image captured at $d = 0$ and the one of the corresponding patch in the image captured at distance d. Let $\overline{I_0}$, $\overline{I_d}$ denote the mean value intensity of these patches, and let m and n are their height and width. Then the normalized correlation coefficients are given by:

$$\rho(d) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(I_0(i,j)-\overline{I_0})(I_d(i,j)-\overline{I_d})}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(I_0(i,j)-\overline{I_0})^2 \sum_{i=1}^{m}\sum_{j=1}^{n}(I_d(i,j)-\overline{I_d})^2}} \tag{15}$$

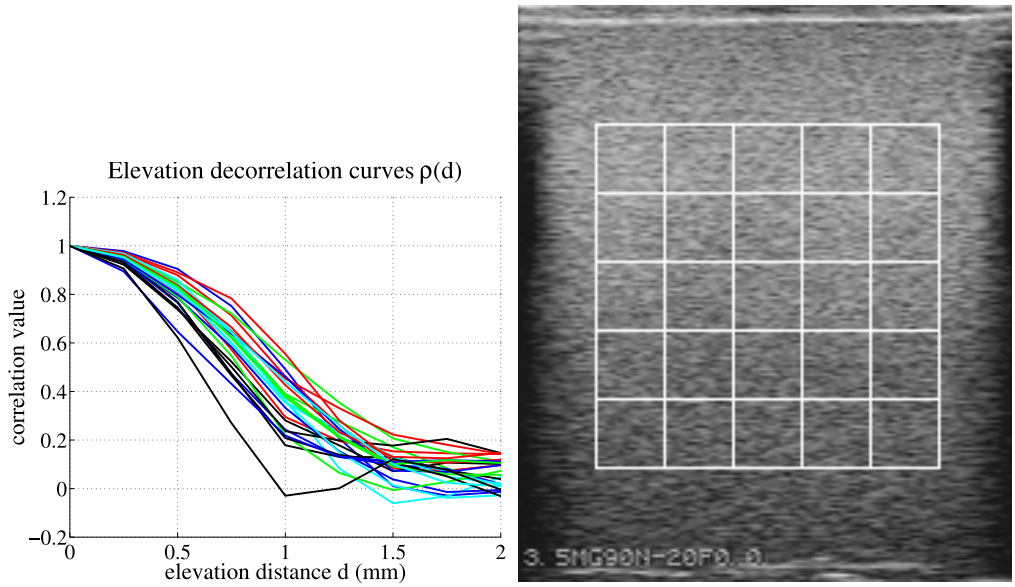

Figure 3: (left) Experimental decorrelation curves of the 25 patches considered in the (right) ultrasound image

These values are measured for several patches positioned in the images. Fig. 3 shows the decorrelation curves when we consider a grid of 25 patches in images taken from an US speckle phantom.

As described in (13), the observed decorrelation curves behave like Gaussian functions, but with different parameters σ. This is due to the fact that the resolution cell width σ is a function of the lateral and axial position of the patch in the image. Generally for sensorless freehand 3D ultrasound, a look-up table based on these calibrated decorrelation curves is used to provide an accurate estimation of the elevation distance from the considered measured inter-patch correlation value. In our motion stabilization application the objective is to minimize the relative position between the observed B-scan and a desired one, therefore we do not require high accuracy on the target plane position estimation. Consequently, we propose to estimate the inter-patch elevation distance directly from (13) by using:

$$\hat{d}(\rho) = \sqrt{-2\hat{\sigma}^2 \ln(\rho)} \tag{16}$$

where $\hat{\sigma} = 0.72$ mm is identified by averaging the experimental decorrelation curves and fitting the model function.

2.2.2 Plane estimation

To estimate the target plane position, the 3D coordinates of a minimum of three non-collinear patches are needed. As (16) gives only absolute value d of the patch Z-coordinate, we must determine the correct sign of each elevation distance. If we first assume that the sign of each inter-patch distance is known, we can estimate the target plane $\{t\}$ position with respect to the intermediate plane $\{c\}$ by using the plane equation:

$$ax + by + cz + d = 0 \tag{17}$$

where x, y, z are the 3D coordinates of the center of a patch belonging to the target image plane with respect to the intermediate image plane $\{c\}$. x, y correspond to its 2D position fixed in the image grid (the same for

the intermediate and target image plane) and z is the signed elevation distance which can be estimated from (17) by:

$$\hat{z} = \sum_{j=1}^{3} \alpha_j f_j(x, y) \tag{18}$$

where $f_1(x, y) = 1$, $f_2(x, y) = x$, $f_3(x, y) = y$ depend on the coordinates x, y which are known and $\alpha_1 = -d/c$, $\alpha_2 = -a/c$, $\alpha_3 = -b/c$ are the parameters of the plane. By considering all the n patches of the grid, these parameters can be estimated by using a classical least-squares algorithm whose the cost function to minimize is the sum of squares of the differences between the estimated and observed elevation distances:

$$J = \sum_{i=1}^{n} (\hat{z}_i - z_i)^2 \tag{19}$$

and which gives solution:

$$(\alpha_1 \; \alpha_2 \; \alpha_3)^T = (\mathbf{M}^\mathsf{T} \mathbf{M})^{-1} \mathbf{M}^\mathsf{T} \mathbf{Z} \tag{20}$$

where the components of the $n \times 3$ matrix $\mathbf{M}$ are given by $M_{i,j} = f_j(x_i, y_i)$ with $i = 1 \ldots n$, $j = 1 \ldots 3$ and the vector $\mathbf{Z}$ contains the n observed elevation distances $Z_i = z_i$. The normal vector of the target plane expressed in the intermediate plane $\{c\}$ is then obtained by:

$$\vec{n} = (a \; b \; c)^T = \frac{(\alpha_2 \; \alpha_3 \; 1)^T}{||(\alpha_2 \; \alpha_3 \; 1)^T||} \tag{21}$$

and the elevation distance of the target plane $\{t\}$ with respect to the intermediate plane $\{c\}$ is $t_z = \alpha_1$.
As the third column of ${}^c\mathbf{H}_t$ in (2) corresponds to the Z axis of the target plane expressed in the the intermediate plane $\{c\}$ the out-of-plane angles α and β can be directly determined from the components of the estimated normal vector $\vec{n}$, with:

$$\begin{aligned} \alpha &= atan(a/c) \\ \beta &= -asin(b) \end{aligned} \tag{22}$$

However, this least-squares algorithm cannot be directly applied to estimate the plane position due to the sign ambiguity of the z_i distance of each patch. So we propose hereafter two methods to estimate the signed elevation distance of each patch.

2.2.2.1 Signed elevation distance: Small motion estimation method

The first method applies the iterative algorithm presented in Fig. 4 to rearrange sign of each distance measurement. The principle is first choose a random sign on each z_i, and to then compute an initial plane estimate and least-squares error using these signs. Then, we modify the sign of a patch and compute the new least-squares error. If the new error norm is lower than the previous one, then the sign is kept or otherwise it is discarded. This process is repeated for the n patches in a loop. At the end, if the resulting error norm is lower than the initial one then the initial error is set to the current one and the loop is repeated until the last resulting error is the same as the initial error. The algorithm will then stop when it converges to one of the two stable symmetric solutions as illustrated in Fig. 5. The first solution corresponds to the case when there is a positive elevation distance $t_z > 0$ between the target and observed plane and the second to the case for a negative distance $t_z < 0$. Note that from one solution we can easily determine the second. For the case presented in Fig. 5, the algorithm converges only with 50 iterations whereas there are, in principle, 2^n (with

```
affect random sign to each zi
compute least-squares error
do
    initial least-squares error = least-squares error
    for i = 1 to n,
        zi = -zi
        compute  new least-squares error
        if (new least-squares error < least-squares error)
            least-squares error = new least-squares error
        else
            zi = -zi
        end if
    end for
while (least-squares error < initial least-squares error)
```

Figure 4: Iterative algorithm for plane position estimation

$n = 25$) possible configurations of the signed distances. In fact, there are fewer than 2^n due to the planarity constraint; indeed this is why such a simple algorithm works.

The 2 solutions of ${}^c\mathbf{H}_t$ are then given by using:

$$\alpha = atan(a/c)\ ,\ \beta = -asin(b) \text{ if } t_z > 0$$
$$\alpha = atan(-a/c)\ ,\ \beta = -asin(-b) \text{ if } t_z < 0 \tag{23}$$

Note that, if $t_z = 0$ there is an ambiguity on the target plane orientation. This problem will be considered next.

Once a correct sign is known for the elevation plane, it is possible to develop a system for tracking it without the need for continual re-estimation. In order to resolve the remaining sign ambiguity and initiate tracking, we have developed a state-transition graph which memorizes the evolution of the sign and uses an intermediate B-scan image to reconstruct the target frame position ${}^c\mathbf{H}_t$ when $|t_z|$ is close to zero. In practice, the B-scan image target that is to be tracked will be chosen in some initial ultrasound image. This will be done after the user positions the probe held by a medical robot to see the target of interest. Therefore at the start, the most recent image and the target B-scan are superposed, so $t_z = 0$. We then propose to initially move the probe by a small control step in the negative elevation direction in order to obtain $t_z > s$ where s is a very low threshold value. This provides initialization for the state-transition graph presented in Fig. 6.

In particular, this first motion provides data for state 1 where the position of the target is given by ${}^c\mathbf{H}_t(t_z > 0)$. This state is maintained while $t_z > s$. If $|t_z|$ decreases below the threshold s due to the motion of soft tissues then an intermediate plane with Cartesian frame $\{s\}$ is set and frozen to the observed target B-scan position ${}^c\mathbf{H}_s = {}^c\mathbf{H}_t(s)$ and the state switches from 1 to 2. In this new state the position of the plane target is then given by ${}^c\mathbf{H}_t = {}^c\mathbf{H}_s(s){}^s\mathbf{H}_t(z_s > 0)$ where ${}^s\mathbf{H}_t(z_s > 0)$ is the homogeneous matrix from the fixed intermediate plane to the target plane computed from equations (20)-(22) with positive elevation distance z_s between these two planes.

This new state is maintained while $|t_z| < s$. Of course there is the possibility of going back to the state 1 if t_z increases when the transition $|t_z| \geq s$ & $|z_s| \leq |t_z|$ is validated. If now $|t_z| \geq s$ & $|z_s| \geq |t_z|$ which means that t_z is negative and is lower than $-s$ then the state goes in 3 where the target position is given

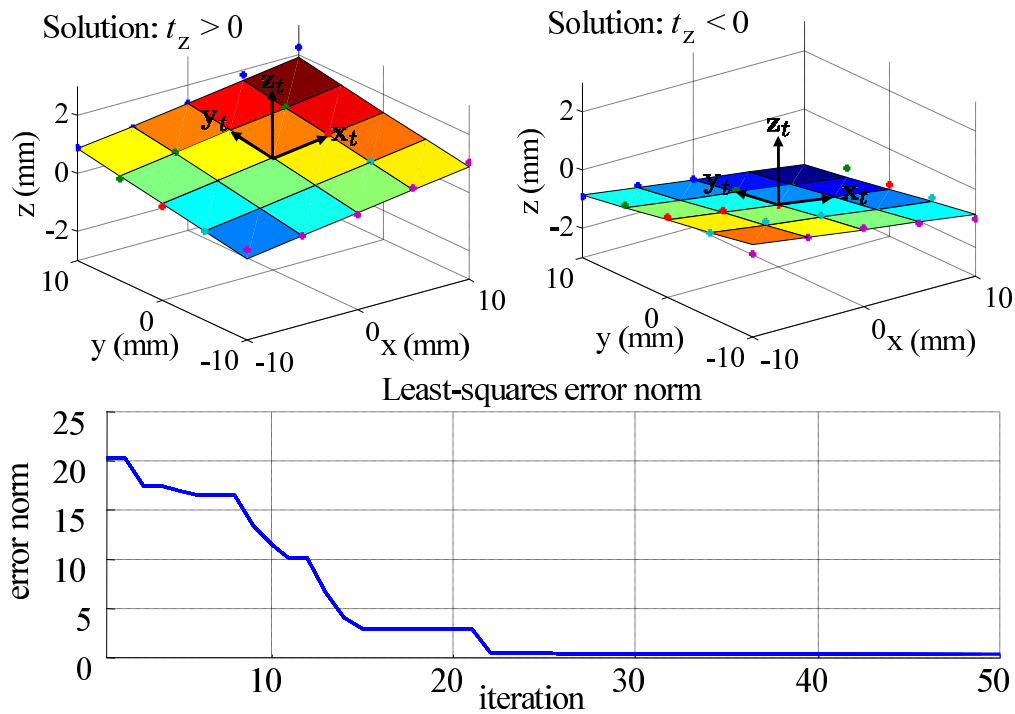

Figure 5: (top) Symmetric plane position solutions provided by the iterative algorithm. The points on the planes show the rearranged (signed) positions of patches after the algorithm convergence. (bottom) Plots of the decreasing least-squares error norm during the iterative algorithm process

directly by the solution with negative elevation distance ${}^c\mathbf{H}_t(t_z < 0)$. If afterwards $|t_z|$ becomes lower than the threshold, the intermediate plane is updated and frozen to the observed target position ${}^c\mathbf{H}_s = {}^c\mathbf{H}_t(-s)$ and the state goes to 4 with solution ${}^c\mathbf{H}_t = {}^c\mathbf{H}_s(-s){}^s\mathbf{H}_t(z_s > 0)$ where ${}^s\mathbf{H}_t(z_s > 0)$ is the transformation matrix from the recent updated intermediate plane to the target. The first state is then retrieved when $|t_z| \geq s$ & $|z_s| \geq |t_z|$. This method permits computation of the correct sign of the distance t_z by taking into account its evolution and avoiding ambiguous orientation case when $t_z = 0$. Moreover in order to obtain smooth transitions when the state switches, the following interpolation function is applied to give the target plane pose vector $\mathbf{p}$:

$$\mathbf{p} = (1 - (|t_z|/s)^2)\, \mathbf{p}_1 + (|t_z|/s)^2\, \mathbf{p}_2 \tag{24}$$

where $\mathbf{p}_1$ is the pose vector describing the reconstructed homogeneous matrix ${}^c\mathbf{H}_t$ obtained during state 2 or 4 and $\mathbf{p}_2$ is the pose vector describing the direct solution ${}^c\mathbf{H}_t$ during state 1 or 3. Note that this function gives no weight to the direct solution ${}^c\mathbf{H}_t$ when $t_z = 0$ in order to reject the unstable case. The components of the normal vector $\vec{n}$ of the B-scan plane and its orientation angles α, β are then retrieved using (2) and (23).

2.2.2.2 Signed elevation distance: Large motion estimation method

The previous method works only locally about the target region due to the rapid rate of speckle decorrelation with out-of-plane motion. Therefore, in order to increase the range of convergence, we propose a second approach that allows to estimate independently the signed elevation distance of each patch belonging to the target image plane for large out-of-plane displacement. The method is described hereafter for one

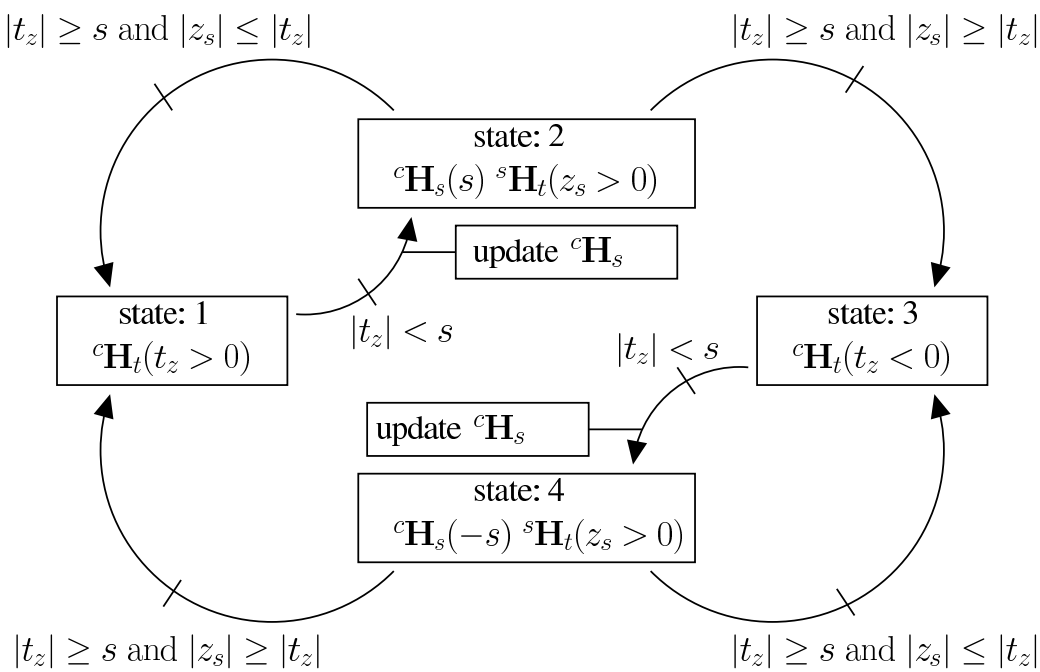

Figure 6: The state-transition graph used to track the sign of the elevation distance t_z and compute the relative position ${}^c\mathbf{H}_t$ between the observed and target planes

patch and is applied to all the patches before fitting the plane to the data.

First, at start time $t = 0$, when the observed patch and the target patch are superposed, the patch image is acquired in a memory array starting at index $k + p$ where k is the index corresponding to the target patch and $p = 0$ is an counter index that represents the number of intermediate patches that will be memorized in the array with positive elevation distance.

As in the previous method, we propose to initialise the sign of the elevation distance by moving the probe in the negative elevation direction. This time we do not apply a step motion but a constant velocity during a very short time period. Therefore, the positive elevation distance of the given target patch computed from the speckle decorrelation increases linearly. When it reaches the threshold value s, the index p is incremented and the positive elevation distance $d_{c[k+p-1]}$ of the target patch with respect to the observed patch is memorized in the array such that $d_{[k+p][k+p-1]} = d_{c[k+p-1]}$ and the observed image patch is stored as a first intermediate patch at array index $k + p$. Here we choose the notation $d_{c[i]}$ to define the signed elevation distance of the memorized patch at index i with respect to the observed patch called c and $d_{[i][j]}$ corresponds to the signed elevation distance of memorized patch at index j with respect to the memorized patch at index i. This is performed during the probe motion each time the distance of the last memorized intermediate patch with respect to the observed patch reaches the threshold value. When the probe motion stop after this initial procedure we obtained the patches "path" configuration shown in Fig. 7.
The relative distances between the memorized patches can then be expressed by the following vectorial system form:

$$\mathbf{Y} = \mathbf{D}\mathbf{P} \tag{25}$$

where $\mathbf{Y}$ is a vector of size $(\sum_{j=1}^{j=n} j)$ with $n = p$, containing the signed relative inter-patch elevation

of the array and the observed patch c. It is given by the following structure:

$$\mathbf{D} = \begin{bmatrix} |d_{c[i]}| & -|d_{c[i+1]}| & 0 & 0 & 0 & 0 & 0 \\ |d_{c[i]}| & 0 & -|d_{c[i+2]}| & 0 & 0 & 0 & 0 \\ 0 & |d_{c[i+1]}| & -|d_{c[i+2]}| & 0 & 0 & 0 & 0 \\ |d_{c[i]}| & 0 & 0 & -|d_{c[i+3]}| & 0 & 0 & 0 \\ 0 & |d_{c[i+1]}| & 0 & -|d_{c[i+3]}| & 0 & 0 & 0 \\ 0 & 0 & |d_{c[i+2]}| & -|d_{c[i+3]}| & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & |d_{c[i+n-1]}| & -|d_{c[i+n]}| \end{bmatrix} \tag{27}$$

with $i = k$ and $n = p$.
$\mathbf{P}$ is a vector of size $(n + 1)$ containing the sign of the distance of all the memorized patches with respect to the observed patch c. After the initialisation procedure it contains only positive signs such that:

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \dots & 1 \end{bmatrix}^T \tag{28}$$

Now, we consider that the soft tissue containing the target patch starts to move along the elevation direction with an unknown sign motion. Its signed elevation distance with respect to the observed patch can then be estimated by the following algorithm. The first step consists of estimating the elevation distance sign of each memorized patch with respect to the observed patch. This is done by minimizing the sum of squares of the differences between the estimated $\hat{\mathbf{Y}} = \mathbf{D}\hat{\mathbf{P}}$ and memorized $\mathbf{Y}$ inter-patch distances:

$$J(\hat{\mathbf{P}}) = (\mathbf{Y} - \mathbf{D}\hat{\mathbf{P}})^T(\mathbf{Y} - \mathbf{D}\hat{\mathbf{P}}) \tag{29}$$

The minimisation is performed by testing all possible sign configurations of vector $\hat{\mathbf{P}}$ and keeping $\hat{\mathbf{P}}$ that provides the lower cost function error J. Note that the possible configurations of $\hat{\mathbf{P}}$ are limited to circular sign sequences such (1,1,1,1,...1), (-1,1,1,1,...1), (-1,-1,1,1,...1), (-1,-1,-1,1,...1), (-1,-1,-1,-1,...1), (-1,-1,-1,-1,...-1), (1,-1,-1,-1,...-1), (1,1,-1,-1,...-1), (1,1,1,-1,...-1), (1,1,1,1,...-1), and are provided in practice by a shift register. All the signed distances $d_{c[j]}$ with $j = i...(i + n)$ are then affected with their estimated signs given by $\hat{\mathbf{P}}$. The second step consists of computing the elevation distance of the target patch with respect to the observed patch. In order to increase robustness of the estimation we perform a distance averaging which gives us the following distance estimate:

$$d_{c[k]} = \frac{1}{n+1} \sum_{j=i}^{j=i+n} d_{c[j]} + d_{[j][k]} \tag{30}$$

with $i = k$ and $n = p$.
These two steps of the algorithm are repeated at each iteration of the soft tissue tracking process.

The value of the estimated signed distance $d_{c[k]}$ is also used to control the evolution of the array of intermediate patches. If the distance becomes greater than its maximal value d_{max} previously achieved $d_{c[k]} > d_{max}$ and if the distance of the $k + p$ patch with respect to the observed patch reaches the threshold value s, $d_{c[k+p]} > s$, then the positive patches counter index p is incremented and a new intermediate patch is acquired in the memory array. In the opposite side if the distance of the target patch with respect to the observed patch goes belows to its minimal value d_{min} previously achieved $d_{c[k]} < d_{min}$ and if the distance of the $k - m$ patch with respect to the observed patch reaches the negative threshold value $-s$

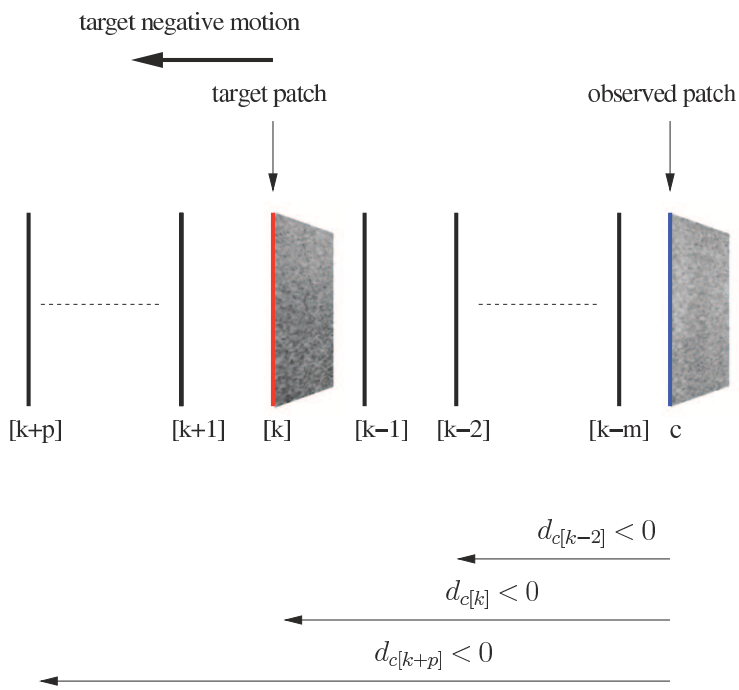

Figure 8: Configuration of the intermediate patches position when the target patch elevation distance is negative and increases in the negative direction

such as $d_{c[k-m]} < -s$, then a negative patches counter index m (initially set to 0) is incremented and a new intermediate patch is acquired in the memory array at index $k - m$. Note that the index m counts the patches of negative elevation distance in opposite to index p which counts the patches of positive distance. Fig. 8 illustrates the case when the target distance is negative and shows the different intermediate patches captured during the motion. Note that if $m > 0$ then we simply adapt the estimation algorithm by setting $i = k - m$ and $n = p + m$ in equations (26) to (30).

For the moment this second method only allows us to locally estimate the signed elevation distance of the target patch since all the memorized patches contained in the array have to be speckle correlated with the observed patch observed by the probe. Therefore to allow large displacement of the target we propose to use a sliding window as illustrated in Fig. 9 in order to include only the intermediate patches closest to the observed one in the estimation process. The sliding windows are centered on the patch l which is the closest to the observed one and whose index l is determined from elevation distance comparison. The estimation process is then performed by setting $i = l - w$ and $n = 2w$ in equations (26) to (30) where $(2w + 1)$ corresponds to the size of the window in term of number of patches.

Note that when the observed patch is far away from the target patch, they are not speckle correlated. This is not a problem if the sliding window is used. However the image region tracking algorithm described in section 2.1 needs a minimum of image correlation between the observed and target patch images to extract the in-plane motion. Therefore we propose to set the reference image used by the tracking algorithm to the image of the patch corresponding to the center of the sliding window (index l). In this way the reference image patch is automatically updated when the sliding window moves due to large target displacement.

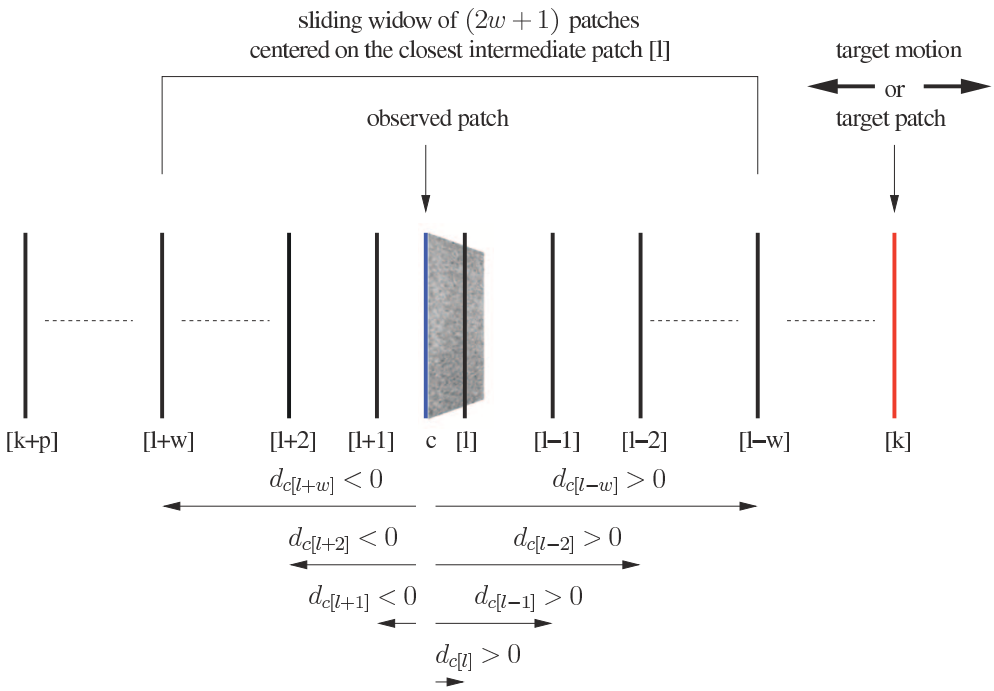

Figure 9: Configuration of the intermediate patches position for large target patch elevation distance estimation. A sliding window is centered on the memorized patch [l] which is the closest patch to the observed one.

Beside if the absolute elevation distance of the target patch decreases then the reference image of the region tracking algorithm is set to a previous memorized one until it retrieves the initial reference when the observed and target patch join together.
An overview of the algorithm is described by the Listings 1 and 2. The Listing 1 gives the successive steps performed to initialise the array of patches. The several steps used to estimate the signed elevation distance of the target patch with the sliding window are given in Listing 2. Note that the successive steps of Listing 2 are continuously iterated with the US stream.

```
   that is the closest to the observed one
imagePatchReference = PatchArray[l]; //uptade the image template reference
   used by the in-plane motions tracking algorithm
```

Listing 2: Estimation of the target patch signed elevation distance and patches array updating

This method is used to estimate independently the signed elevation distance of each patch belonging to the target plane. The signed elevation distance t_z and the out-of-plane angles α, β of the target plane $\{t\}$ with respect to the intermediate plane $\{c\}$ are then computed from equations (20)-(22).

3 Visual servoing

Now that the complete position of the B-scan target can be estimated, we present the control scheme used to control a medical robot holding the ultrasound probe in order to reach and stabilize a moving B-scan target. We propose a hybrid visual servoing approach that consists of independently controlling the in-plane 3-DOF and out-of-plane 3-DOF motions of the US probe, respectively, by a 2D image-based visual servoing algorithm and a 3D visual servoing algorithm.

3.1 Out-of-plane motion control

The out-of-plane motion stabilization is performed by a 3D visual servo control. We chose as the visual features $\mathbf{s}_1 = (a\ b\ c\ t_z)^T$ the 3 components of the normal vector $\vec{n}$ of the estimated target plane and its elevation distance t_z with respect to the observed B-scan. The desired visual feature vector to achieve is $\mathbf{s}_1^* = (0\ 0\ 1\ 0)^T$ which means that the final position of the normal vector of the target plane will be orthogonal to the observed image and that relative elevation distance will be null. The variation of the visual information $\mathbf{s}_1$ to the out-of-plane velocity $\mathbf{v}_1 = (v_z\ \omega_x\ \omega_y)^T$ of the probe is given by:

$$\dot{\mathbf{s}}_1 = \mathbf{L}_{s_1}\mathbf{v}_1 = \begin{bmatrix} 0 & 0 & -c \\ 0 & c & 0 \\ 0 & -b & a \\ -1 & 0 & 0 \end{bmatrix} \mathbf{v}_1 \tag{31}$$

where v_z is the probe translational velocity along the orthogonal Z axes of the observed image frame $\{p\}$ (attached to the center of the image) and ω_x, ω_y are respectively the rotational velocities around the X and Y axis. In visual servoing $\mathbf{L}_{s_1}$ is called the interaction matrix (see [Espiau *et al.*, 1992]) and is determined from the geometrical model of the considered system. In our case it depends only on the components of the normal vector $\vec{n}$ of the target plane. The visual servoing task can then be expressed as a regulation to zero of the task function $\mathbf{e}_1 = \mathbf{s}_1 - \mathbf{s}_1^*$. Usually, the control law is defined such as the task $\mathbf{e}_1$ decreases exponentially in order to behave like a first order system by using a proportional controller [Espiau *et al.*, 1992]. In this work we apply rather the second-order minimization technique introduced in [Malis, 2004] which uses the following control law to improve the trajectory for large displacement:

$$\mathbf{v}_1 = -2\lambda_1(\hat{\mathbf{L}}_{s1} + \mathbf{L}_{s1}^*)^+\mathbf{e}_1 \text{ with gain } \lambda_1 > 0 \tag{32}$$

where $\hat{\mathbf{L}}_{s1}$ is the interaction matrix estimated at each control iteration and $\mathbf{L}_{s1}^*$ is the one at the desired location (with $a = b = 0$ and $c = 1$).

3.2 In-plane motion control

To control the in-plane motion of the probe we implement an image-based visual servoing algorithm where the visual features $\mathbf{s}_2 = (t_x\ t_y\ \gamma)^T$ are directly the translation t_x, t_y and the rotation γ extracted and expressed in the observed image by using the method described in section 2.1. The corresponding desired feature vector to reach is $\mathbf{s}_2^* = (0\ 0\ 0)^T$ and the interaction matrix $\mathbf{L}_{\mathrm{s}2}$ related to $\mathbf{s}_2$ such that $\dot{\mathbf{s}}_2 = \mathbf{L}_{\mathrm{s}2}\mathbf{v}_2$, is simply a 3×3 identity matrix. The control velocity $\mathbf{v}_2 = (v_x\ v_y\ \omega_z)^T$ to apply to the probe in order to obtain an exponentially decreasing visual error $\mathbf{e}_2 = \mathbf{s}_2 - \mathbf{s}_2^*$ is then obtained by:

$$\mathbf{v}_2 = -\lambda_2(\mathbf{L}_{\mathrm{s}2})^{-1}\mathbf{e}_2 \text{ with gain } \lambda_2 > 0 \tag{33}$$

where v_x, v_y are the translational velocities of the probe along the X and Y axis of the reference frame $\{p\}$ attached to the observed image, and ω_z is the rotational velocity around its Z axes.

The 6-DOF control needed to track the full motion of the target B-scan is finally performed by applying to the probe the screw velocity $\mathbf{v} = (v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z)^T$ whose components are given by the two independent control laws (32) and (33).

4 Simulation results

4.1 Ultrasound imagery simulator

We first apply the algorithms described above to simulated ground truth data to analyze how the system performs under ideal circumstances. We then gradually introduce systemic and random errors into the data and the tracking system, thereby gradually approaching realistic scenarios, before an experimental validation on real data (especially on human data) is attempted. To this end, we developed an ultrasound simulator software which allows us to position and move a 2D virtual probe and simulate a moving 3D US volume. We composed an US volume from 100 parallel real B-mode US images of 180×210 pixels resolution with a pixel size of 0.2×0.2 mm, captured from an ultrasound speckle phantom at elevation intervals of 0.25 mm.

The simulator was built with the Visualization ToolKit (VTK) software system [Schroeder *et al.*, n.d.] and the Visual Servoing Platform (ViSP) [Marchand *et al.*, 2005], both freely available as open source resources, implemented as C++ routines and libraries. We use VTK to render the 3D view of the US volume, as shown in Fig. 10 and to generate the observed 2D US image with cubic interpolation, as if was generated by a virtual US probe. We also use ViSP to implement the target B-scan motion extraction from the resliced US volume and to compute the visual servo control law applied to the probe.

4.2 Stabilization robotic task results

We simulated the 6-DOF motion of the volume by applying 6 sinusoidal signals with same period of 5 seconds to the position of a Cartesian frame $\{\mathrm{o}\}$ attached to the volume and initially superposed to the ultrasound plane frame $\{p\}$ such that $\{o(t = 0)\} = \{p(t = 0)\}$. The translational magnitudes were set to 10 mm along the X, Y and 12 mm along the Z axes of $\{o\}$ and the rotational magnitudes were set to 10 deg around the X, Y axis and 8 deg around the Z axes. We used a grid of 25 patches (25×25 pixels for each patch) and a threshold elevation distance s of 0.1 mm to extract the out-of-plane motion. A patch of 50×50 pixels centered in the grid was employed to extract the in-plane motion.

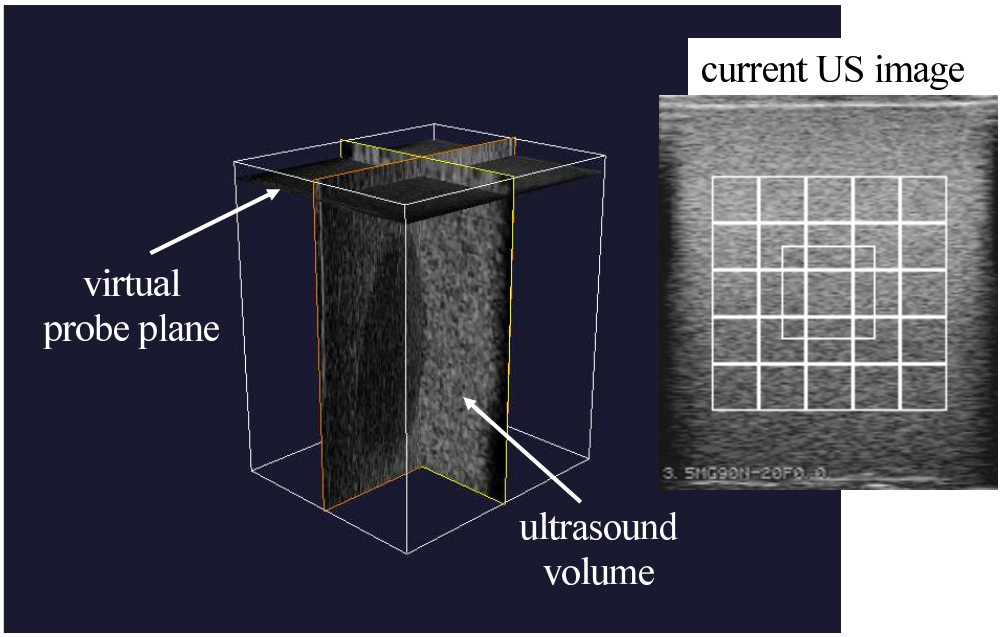

Figure 10: Ultrasound simulator: 3D view of the ultrasound volume and the initial ultrasound image observed by the virtual probe with the 25 speckle patches (grid) and the in-plane tracking region of interest (biggest box)

First, we tested the motion stabilization task using the out-of-plane small motion estimation method described in section 2.2.2.1 and the decoupled control scheme proposed in section 3. The gain of the control laws (32) and (33) were both fixed to $\lambda_1 = \lambda_2 = 10$.
Fig. 11 shows the time responses of the out-of-plane and in-plane positioning errors during the full motion stabilization task. The components of the out-of-plane error correspond to the α and β angles and the elevation distance t_z of the target B-scan plane with respect to the observed B-scan. Their values are linked to the visual feature $\mathbf{s}_1$ by the relation (22) whereas the in-plane error corresponds directly to the visual feature vector $\mathbf{s}_2$. Fig. 11 also shows the evolution of the volume position and probe position with respect to a fixed base frame. We can see that the task is performed well since only tracking errors lower than 0.8 mm for the translation and 0.6 deg for rotation components are measured. Fig. 12 shows the control velocity screw applied to the probe and the evolution of the inter-patch speckle correlation values between the observed and target B-scan images. The figure also presents the evolution of the plane estimation least-squares error norm and the cycle of the state-transition graph performed to track the elevation distance sign. As we can see, correlation values are decreasing due to the tracking error and reach the minimal value of 0.25.

In a second simulation we test the motion stabilization task using the out-of-plane large motion estimation method presented in section 2.2.2.2 with a sliding window set to 7 intermediate patches such that $w = 3$. Fig. 13 and 14 shows the results when the same decoupled control scheme is used with $\lambda_1 = \lambda_2 = 10$. We can note that the tracking errors are the same than the first simulation. However the speckle correlation values between the patches of the observed image and the patches of the intermediate plane, which is fixed at the center of the sliding window, do not go below the minimal value of 0.9 as we can see on Fig. 14. This means that the out-of-plane large motion estimation method will be more robust to large tracking error. To

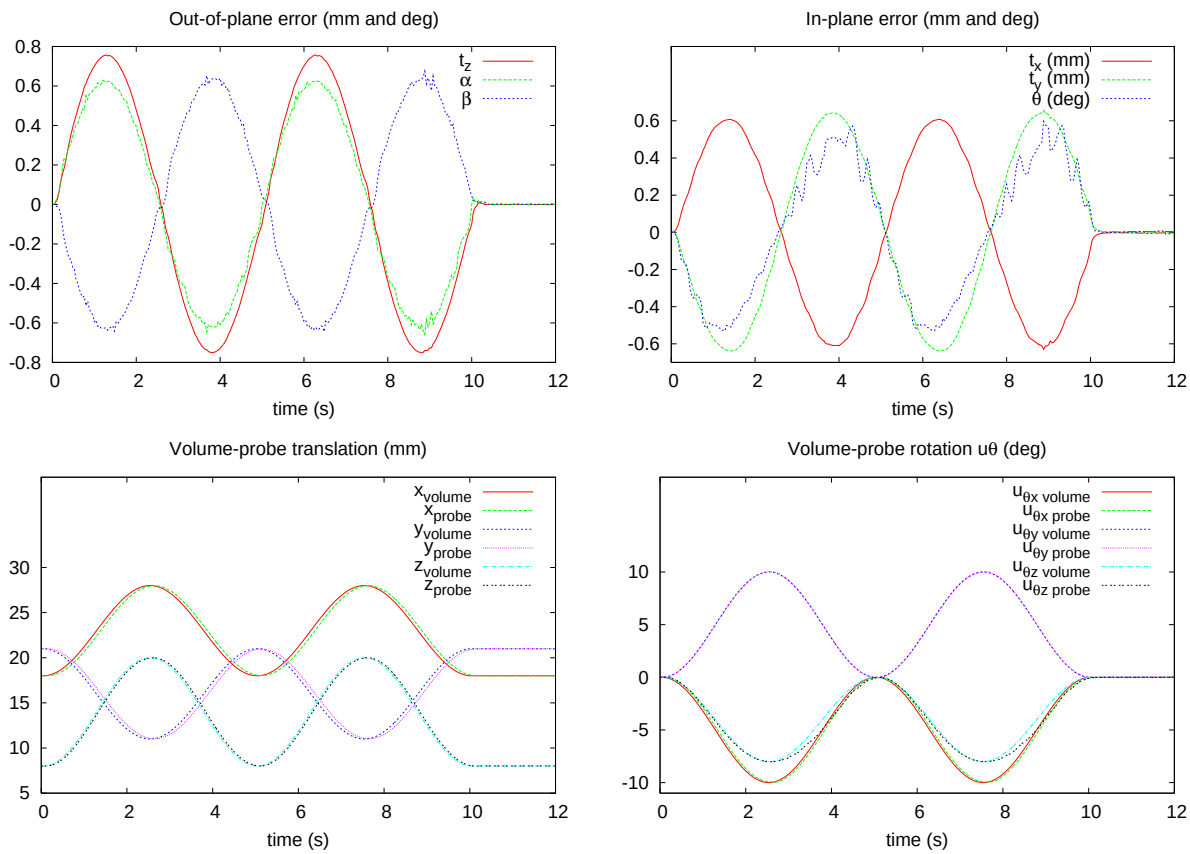

Figure 11: (top) Out-of-plane and in-plane tracking positioning errors - (bottom) Position and orientation ($\mathbf{u}\theta$ representation) of the volume and the ultrasound probe with respect to a fixed base frame

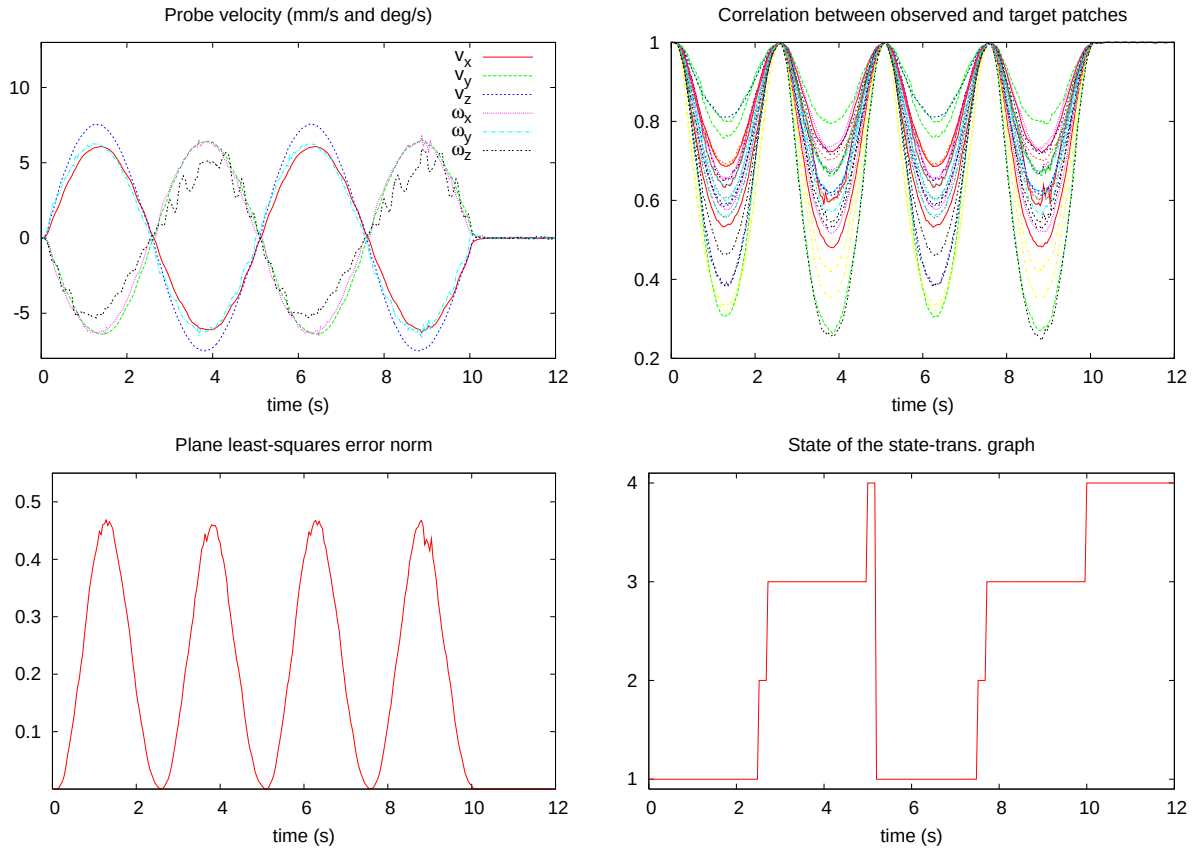

Figure 12: (top) Velocity control screw applied to the virtual ultrasound probe and speckle correlation values of the patches between the observed and target image plane - (bottom) Target plane least-squares error norm and state value of the state-transition graph use to extract the elevation sign

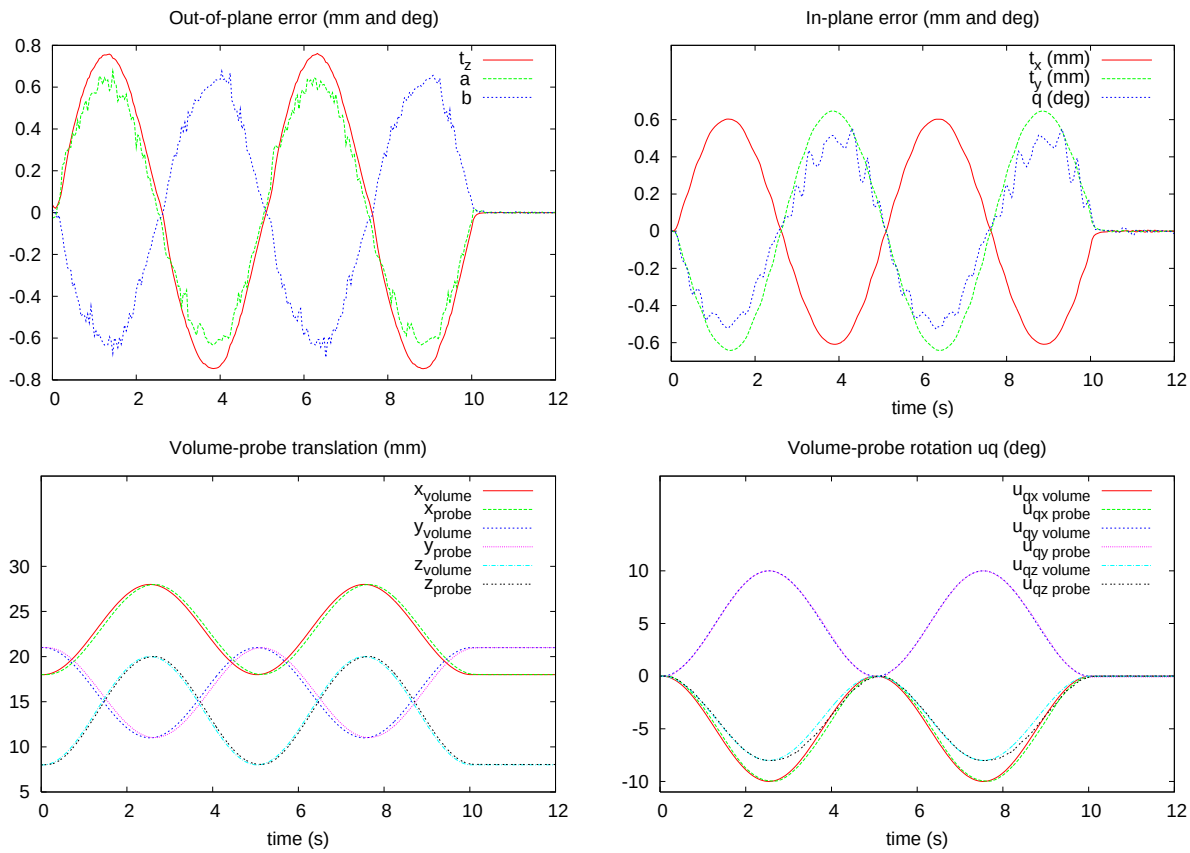

Figure 13: (top) Out-of-plane and in-plane tracking positioning errors. (bottom) Position and orientation (u representation) of the volume and the ultrasound probe with respect to the base frame

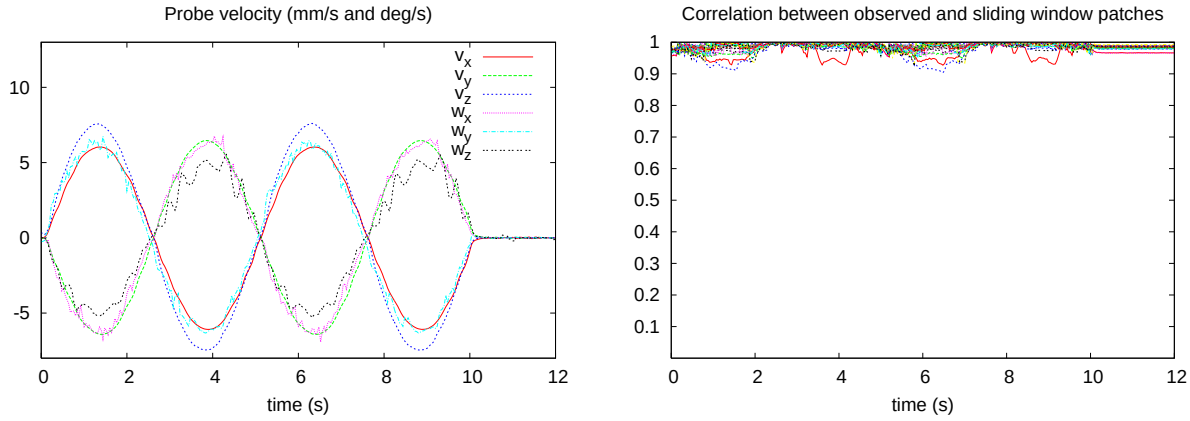

Figure 14: (left) Velocity control screw applied to the virtual ultrasound probe. (right) speckle correlation values of the patches between the observed image plane and the image plane at the center of the sliding window

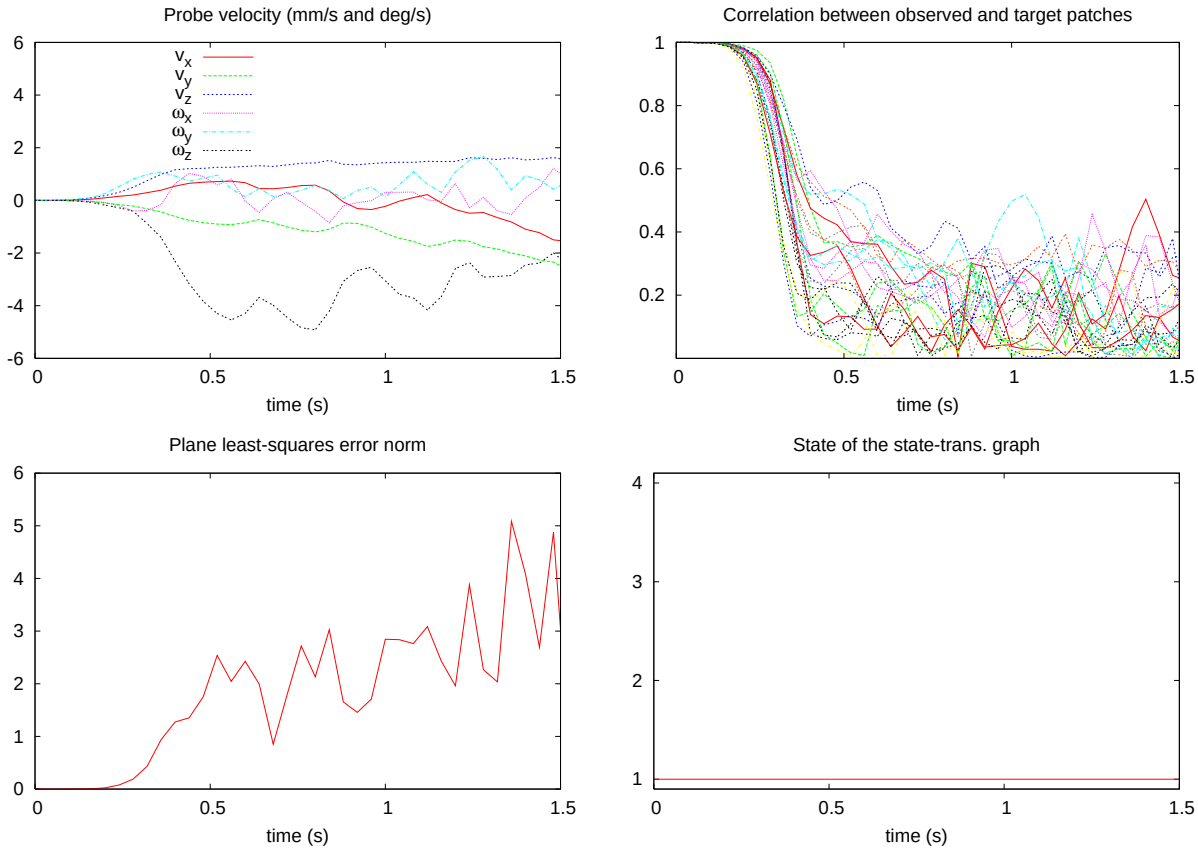

Figure 16: (top) Velocity control screw applied to the virtual ultrasound probe and speckle correlation values of the patches between the observed and target image plane - (bottom) Target plane least-squares error norm and state value of the state-transition graph use to extract the elevation sign

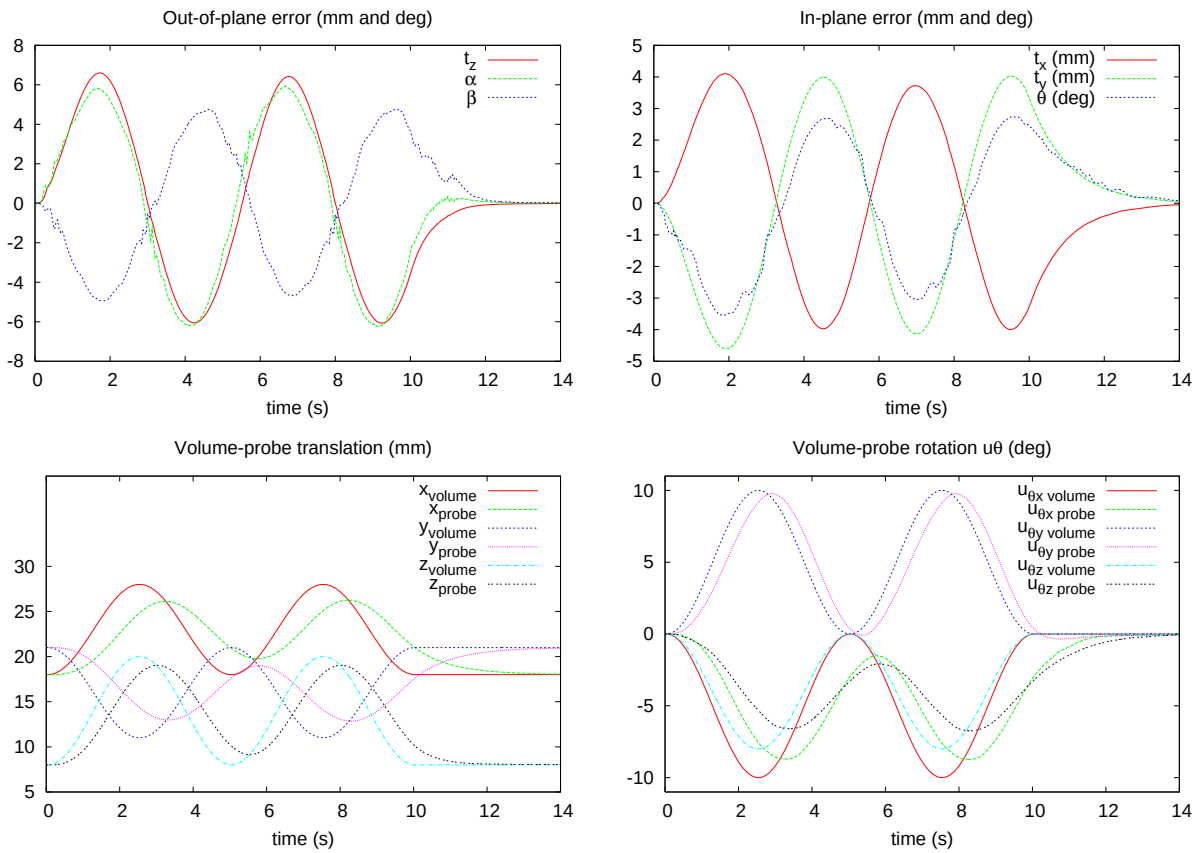

Figure 17: (top) Out-of-plane and in-plane tracking positioning errors - (bottom) Position and orientation ($\mathbf{u}\theta$ representation) of the volume and the ultrasound probe with respect to a fixed base frame

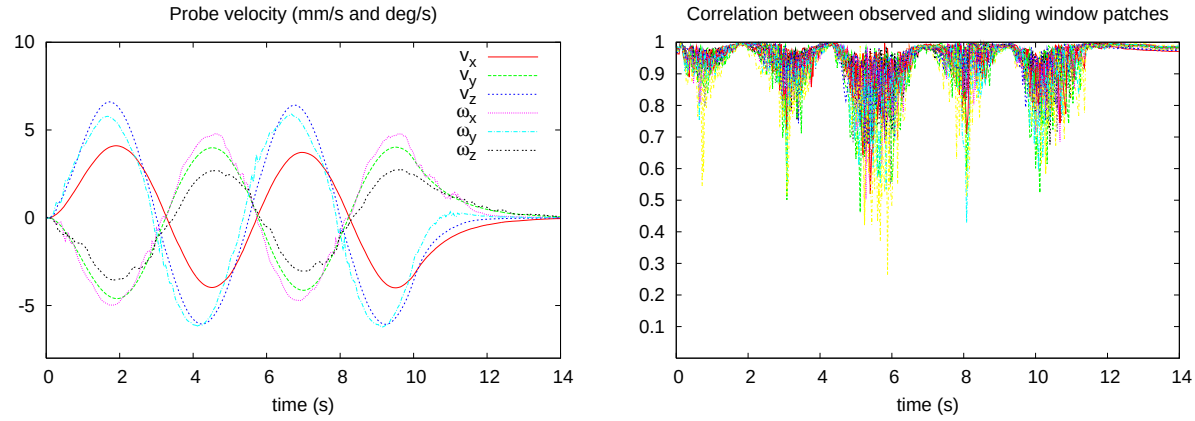

Figure 18: (left) Velocity control screw applied to the virtual ultrasound probe - (right) speckle correlation values of the patches between the observed image plane and the image plane fixed at the center of the sliding window

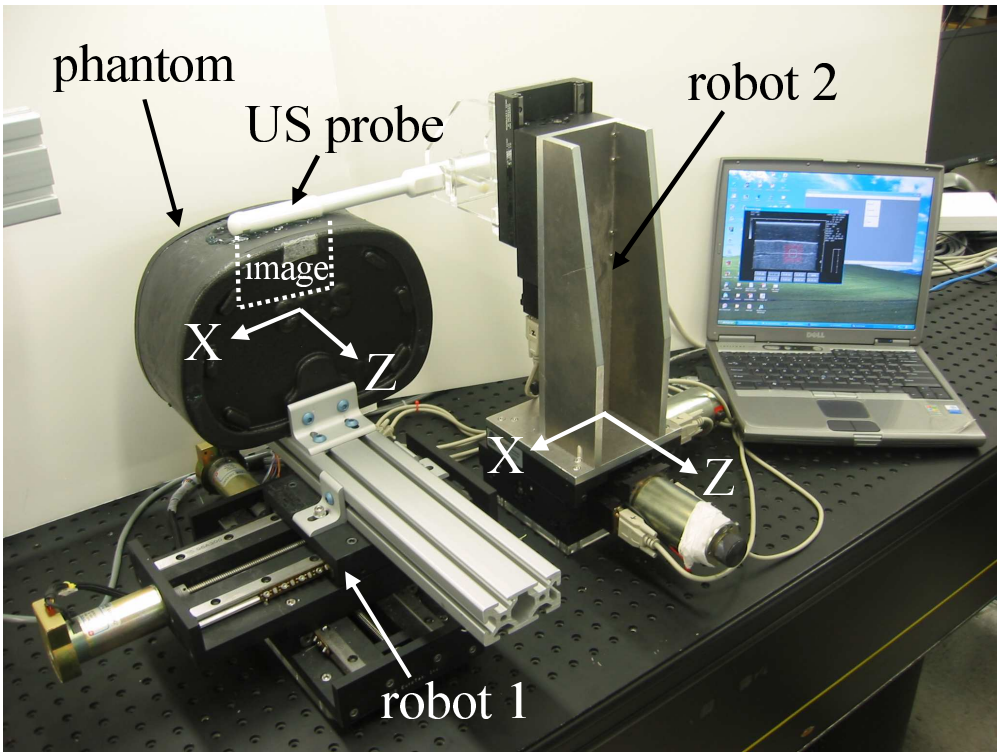

Figure 19: Experimental setup for 2-DOF motions compensation

5 Experimental results

5.1 2-DOF motions compensation

In a first step, we tested the motion stabilization method on 2-DOF motions combining a translation along the image X axis (in-plane translation) and elevation Z axis (out-of-plane translation). The experimental, setup, shown in Fig. 19, consists of two X-Z Cartesian robots fixed and aligned on an optical table. The first robot provides a ground truth displacement for an US speckle phantom. The second robot holds a transrectal 6.5 Mhz US transducter and is controlled as described above to stabilize a moving B-scan target. The US image is 440×320 pixels with resolution of 0.125 mm/pixel. A laptop computer (Pentium IV 2 Ghz) captures the US stream at 10 fps, extracts the target plane position by using a grid of 25 patches (25×25 pixels size) and computes the velocity control vector applied to the probe holding robot. For this experiment we implemented the out-of-plane large motion estimation method introduced in section 2.2.2.2. The plots in Fig. 20 show the evolution of the robots positions and the tracking error when sinusoidal motions (magnitude of 30 mm on each axis) were applied to the phantom. The dynamic tracking error was below 3 mm for in-plane translation and 3.5 mm for the elevation translation. This error is attributed the dynamics of the target motion, time delays in the control scheme, and the dynamics of the probe holding robot. In order to determine the static accuracy of the tracking robotic task, we applied a set of 140 random positions to the phantom by using ramp trajectories while tracking the target plane by the robotized probe. When the probe stabilized at a position, the phantom was held motionless for 2 seconds and the locations of the two robots were recorded. We recorded a static error of 0.0219 ± 0.05 mm (mean $\pm$ standard deviation)

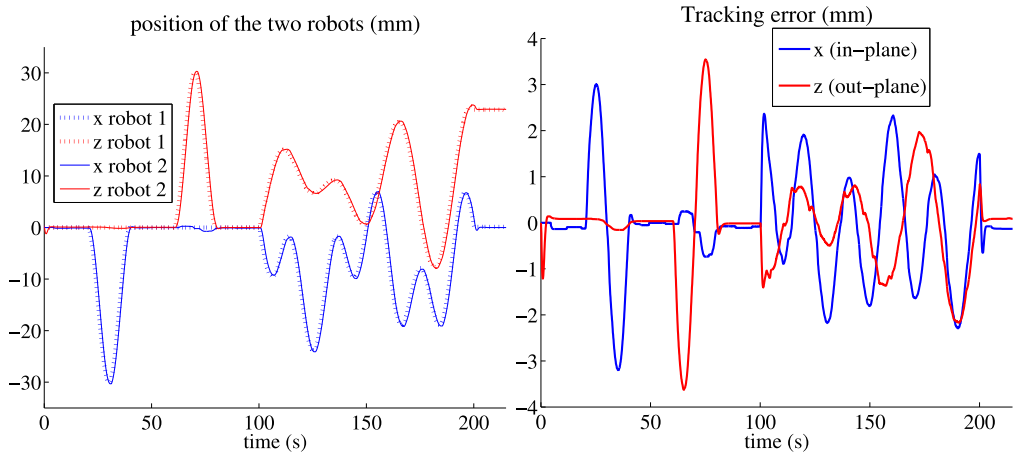

Figure 20: (left) Evolution of the robots positions - (right) Tracking error

for the in-plane positioning and 0.0233±0.05 mm for the out-of-plane positioning, which is close to the positioning accuracy of the robots (± 0.05 mm).

5.2 6-DOF motions compensation

In a second step, we tested our motion stabilization approach by considering 6-DOF rigid motions that were manually applied to the ultrasound phantom. The experimental setup is shown in Fig. 21. It consists of a 6-DOF medical robot equiped with a force sensor, similar to the Hippocrate system [14], that holds a broadband 5-2 MHz curved array usually used for general abdominal imaging. In order to keep the transducer in contact with the phantom, the probe velocity component along the Y axis of the observed image was directly constrained by a classical closed-loop force control scheme in such a way to keep a contact force of 2N along the Y axis direction. The remaining 5 DOFs of the probe include 2 in-plane motions (1 translation along the X axis and 1 rotation around the Z axis of the observed image), and 3 out-of-plane motions (1 translation along the Z axis and 2 translations around the X and Y axes of the observed image). These 5 DOFs were actuated by our motion stabilization approach using only the speckle information. Since the 6-DOF motions are applied manually (from hands) to the ultrasound phantom, we have no accurate ground truth related to its 3D pose in opposite to the first experimental setup where 2 robots were used. Nevertheless, a ground truth can be provided by using an external vision system that measures the phantom and the object respective 3D poses. In our case, we use a remote calibrated camera that observes 2 patterns of visual dots that are attached respectively on the phantom and the ultrasound probe as shown in Fig. 21 and perform pose computation by using the Dementhon approach [Dementhon and Davis, 1995]. The US image stream of 384×288 pixels with resolution of 0.58 mm/pixel was captured at 12 fps and the out-of-plane motion of the target B-scan image was estimated by using a grid of 9 patches (25×25 pixels size). In a first experiment we tested the out-of-plane small motion estimation method introduced in section 2.2.2.1. Unfortunately, the motion stabilization failed few time after we started to move manually the ultrasound phantom. This was due to the phantom jerky motion whose frequency component induced by the hands tremor was too high in comparison to the low bandwidth (12 Hz) of the robotic system. Therefore it resulted a large tracking error with a loss of speckle correlation between the observed and target B-scan. In a second experiment we tested the out-of-plane large motion estimation method introduced in section 2.2.2.2

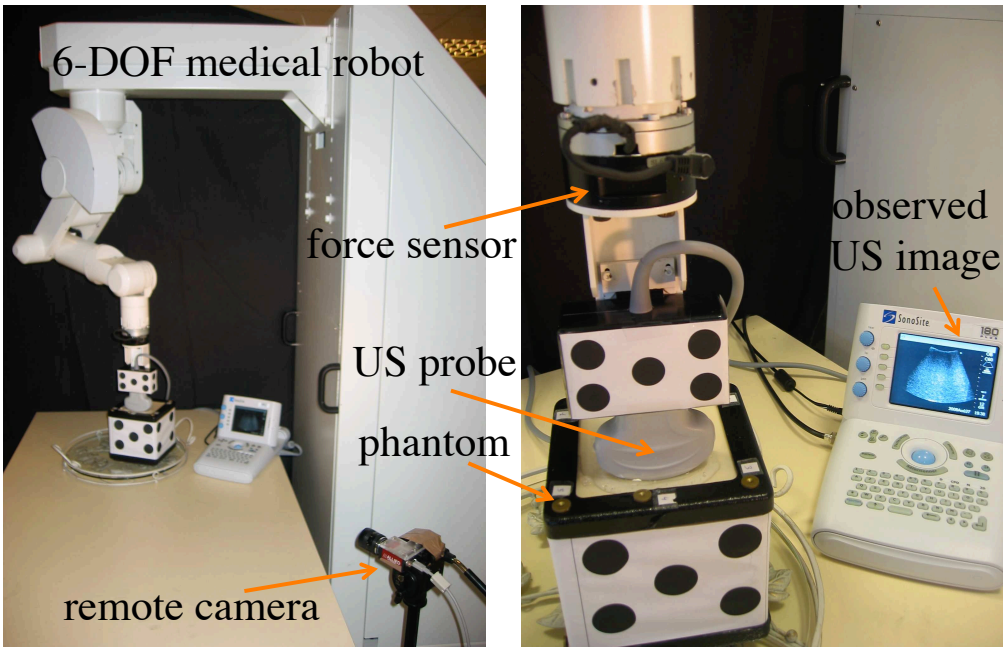

Figure 21: Experimental setup for 6-DOF motions compensation

which is based on the use of memory array of intermediate patches. The plots in Fig. 22 present the time evolution of the 3D poses of the US phantom and US probe both expressed in the remote camera frame and the positioning error of the probe with respect to the phantom during the test. We can see that the US probe automatically follows the motions of the phantom with tracking errors lower than 1.4 cm for the translation and 3 deg for rotation components. Note that this error also combines the pose estimation error inherent to the camera localisation system. These results validate the concept of our automatic stabilisation approach in the case of a rigid motion including both translations and rotations. The tracking error could be reduced if a prediction of its variation is introduced into the control law by some methods such as Kalman filter or generalized predictive controller [Ginhoux *et al.*, 2005]. Adopting recent methods [Rivaz *et al.*, 2006] for more accurate and efficient identification of fully developed speckle patches should also improve on tracking performance and may allow estimation of relative motion between different soft tissue elements.

6 Conclusion

This paper has presented an estimation and control method to automatically stabilize the 6-DOF motion of a conventional 2D ultrasound probe with respect to a moving 3D ultrasound volume by tracking the displacement of a B-scan image relative to a reference target. The out-of-plane motion was extracted from the speckle information contained in the ultrasound image, and an image region tracking method was used to extract the in-plane motion. Two approaches were considered to estimate the out-of-plane motion and compared from simulation and experimental results. A hybrid visual control scheme was proposed to automatically move the probe in order to stabilize the full motion of the target B-scan. The method was first validated in simulation by controlling a virtual probe interacting with a static ultrasound volume acquired from a medical phantom.

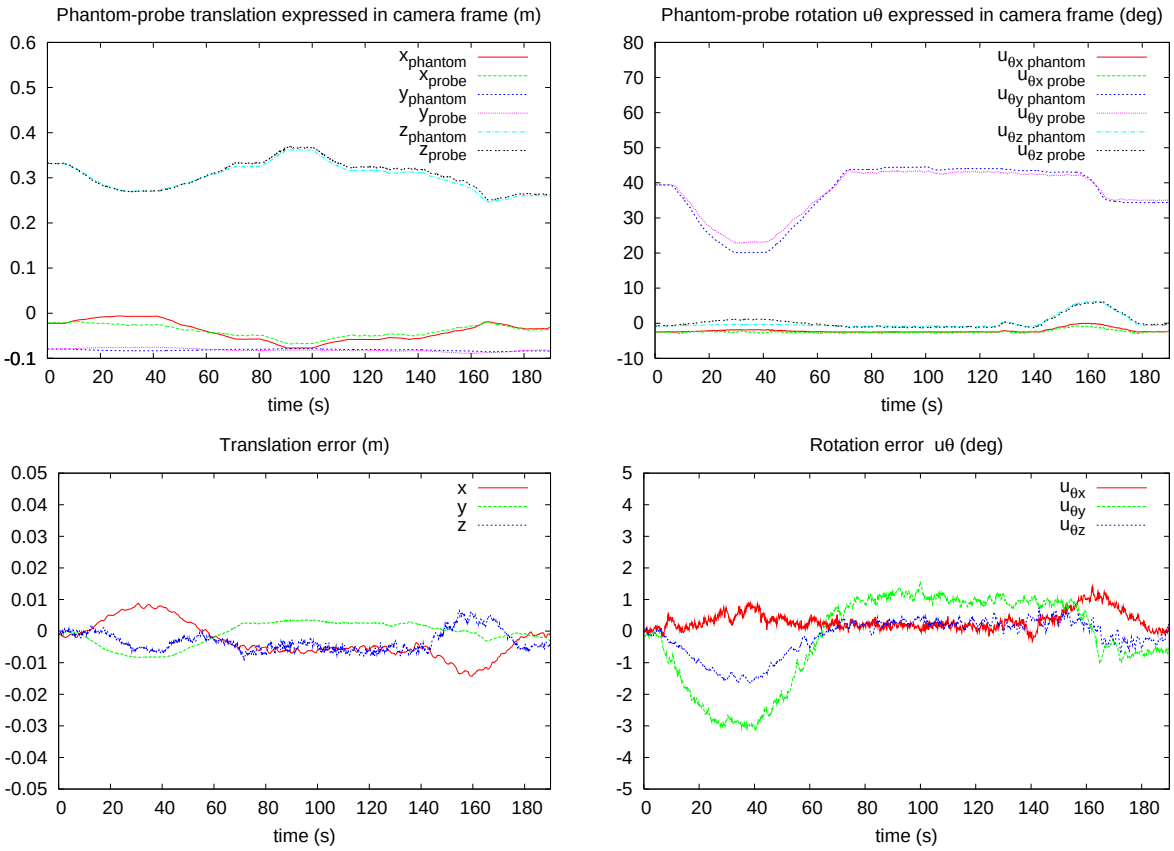

Figure 22: (top) Translation and orientation ($\mathbf{u}\theta$ representation) of the phantom and the ultrasound probe with respect to the remote camera frame - (bottom) Translation error and orientation error ($\mathbf{u}\theta$ representation) of the probe with respect to the phantom

The approach was then demonstrated on two different experimental setups. The first one consisted of an ultrasound speckle phantom, a 2-DOF robot for simulating tissue motion, and a 2-DOF robot controlling the ultrasound probe directly from the speckle information. The results demonstrate in a first step the validity of our approach for 2-DOF motions combining a translation along the image X axis (in-plane translation) and elevation Z axis (out-of-plane translation). In a second experiment we also demonstrated the approach for both translational and rotational motions by using an experimental setup consisting of a 6-DOF medical robot actuating the probe and an ultrasound speckle phantom that we manually moved.

In the introduction, we identified prostate brachytherapy as a clinical application of this work. We are currently addressing several challenges in adapting our work to prostate brachytherapy. First and foremost, we must not alter clinical setup and workflow. In current practice, the probe is moved in 2-DOF by the mechanical stepper under manual actuation, but our motion tracking will work in full 6-DOF. We can encode and actuate existing DOF-s of the stepper, but further modifications are prohibitive. To this end, several extensions will be necessary to our current tracking and servoing techniques. Most contemporary TRUS probes have two perpendicularly arranged transducers: one crystal provides a transverse image perpendicular to the transition axis and a second crystal gives a sagittal image across the rotation axis. In essence, the transverse crystal maps the prostate in Cartesian space while the sagittal crystal works in a cylindrical frame of reference. Therefore we will adapt our automatic stabilization approach to the mixed Cartesian-cylindrical scheme used in TRUS imaging. Second, we will attempt to track the target and needle at the same time with a single TRUS probe. We expect that some target and needle motions can be compensated for, and the remaining misalignments will have to be represented visually. Such a mixed scheme will undoubtedly lead to an extensive investigation of human machine interface techniques as well. Finally, in a later phase, we will integrate the resulting 6-DOF motion tracking and 2-DOF TRUS image stabilization with an existing needle placement robotic system ([Fichtinger *et al.*, 2008]). Altogether, the work presented here has launched us on a challenging and clinically important trajectory of research.

Acknowledgments

The authors acknowledge the supports of the National Science Foundation under Engineering Research Center grant EEC-9731748 and the French INRIA Institute. The authors thank Dr. Emad Boctor for providing the 3DUS data needed by the ultrasound imagery simulator and Dr. Ankur Kapoor and Dr. Iulian Iordachita for assistance in designing of the first experimental setup at the Johns Hopkins University. We also thank Dr. Danny Y. Song (Johns Hopkins University Hospital) and Dr. Everette C. Burdette (Acoustic MedSystems, Inc.) for assistance in systems concept development and expert advice in clinical brachytherapy instrumentation.

References

Abolmaesumi, P., S. E. Salcudean, W. H. Zhu, M. Sirouspour and S. DiMaio (2002). Image-guided control of a robot for medical ultrasound. *IEEE Trans. on Robotics and Automation* **18**(1), 11–23.

Bachta, W. and A. Krupa (2006). Towards ultrasound image-based visual servoing. In: *IEEE Int. Conf. on Robotics and Automation, ICRA'2006*. Orlando, Florida.

Baker, S. and I. Matthews (2004). Lucas-kanade 20 years on: A unifying framework. *Int. Journal of Computer Vision* **56**(3), 221–255.

Benhimane, S. and E. Malis (2004). Real-time image-based tracking of planes using efficient second-order minimization. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*. Sendai, Japan. pp. 943–948.

Boctor, E., I. Iordachita, G. Fichtinger and G. D. Hager (2005). Real-time quality control of tracked ultrasound. In: *8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, MICCAI'2005*. Palm Springs, California, USA. pp. 621–630.

Boctor, E., M. deOliveira, M. Choti, R. Ghanem, R. H. Taylor, G. D. Hager and G. Fichtinger (2006). Ultrasound monitoring of tissue ablation via deformation model and shape priors. In: *9th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, MICCAI'2006*. Copenhagen, Denmark. pp. 405–412.

Bohs, L. N., B. J. Geiman, M. E. Anderson, S. C. Gebhart and G. E. Trahey (2000). Speckle tracking for multi-dimensional flow estimation. *Ultrasonics* **28**(1), 369–375.

Chang, R. F., W. J. Wu, D. R. Chen, W. M. Chen, W. Shu, J. H. Lee and L. B. Jeng (2003). 3-d us frame positioning using speckle decorrelation and image registration. *Ultrasound in Medicine and Biology* **29**(6), 801–812.

Dementhon, D. and L. Davis (1995). Model-based object pose in 25 lines of code. *Int. Journal of Computer Vision* **15**, 123–141.

Espiau, B., F. Chaumette and P. Rives (1992). A new approach to visual servoing in robotics. *IEEE Trans. Robotics and Automation* **8**(3), 313–326.

Fichtinger, G., J.P. Fiene, C.W. Kennedy, G. Kronreif, I. Iordachita, D.Y. Song, E.C Burdette and P. Kazanzides (2008). Robotic assistance for ultrasound-guided prostate brachytherapy. *Medical Image Analysis* **12**(5), 535–545.

Gee, A. H., R. J. Housden, P. Hassenpflug, G. M. Treece and R. W. Prager (2006). Sensorless freehand 3d ultrasound in real tissues: Speckle decorrelation without fully developed speckle. *Medical Image Analysis* **10**(2), 137–149.

Ginhoux, R., J. Gangloff, M. de Mathelin, L. Soler, M. M. Arenas Sanchez and J. Marescaux (2005). Active filtering of physiological motion in robotized surgery using predictive control. *IEEE Transactions on Robotics* **21**(1), 67–79.

Hager, G. D. and P. N. Belhumeur (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(10), 1025–1039.

Hong, J., T. Dohi, M. Hashizume, K. Konishi and N. Hata (2004). An ultrasound-driven needle insertion robot for percutaneous cholecystostomy. *Physics in Medicine and Biology* **49**(3), 441–455.

Krupa, A., G. Fichtinger and G. D. Hager (2007*a*). Full motion tracking in ultrasound using image speckle information and visual servoing. In: *IEEE Int. Conf. on Robotics and Automation, ICRA'2007*. Roma, Italia. pp. 2458–2464.

Krupa, A., G. Fichtinger and G. D. Hager (2007*b*). Real-time tissue tracking with b-mode ultrasound using speckle and visual servoing. In: *10th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, MICCAI'2007*. Vol. 2. Brisbane, Australia. pp. 1–8.

Laporte, C. and T. Arbel (2007). Probabilistic speckle decorrelation for 3d ultrasound. In: *10th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, MICCAI'2007*. Vol. 1. Brisbane, Australia. pp. 925–932.

Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. In: *IEEE Int. Conf. on Robotics and Automation, ICRA'2004*. New Orleans, USA.

Marchand, E., F. Spindler and F. Chaumette (2005). Visp for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine* **12**(4), 40–52.

Martinelli, T., J.L. Bosson, L. Bressollette, F. Pelissier, E. Boidard, J. Troccaz and P. Cinquin (2007). Robot-based tele-echography; clinical evaluation of the ter system in abdominal aortic exploration. *Journal of Ultrasound in Medicine* **26**(11), 1611–1616.

Pierrot, F., E. Dombre, E. Degoulange, L. Urbain, P. Caron, S. Boudet, J. Gariepy and J. Megnien (1999). Hippocrate: A safe robot arm for medical applications with force feedback. *Medical Image Analysis (MedIA)* **3**(3), 285–300.

Rivaz, H., E. Boctor and G. Fichtinger (2006). Ultrasound speckle detection using low order moments. In: *IEEE International Ultrasonics Symposium*. Vancouver, Canada.

Schroeder, W., K. Martin and B. Lorensen (n.d.). *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics*. Inc. publishers, 3rd Edition, ISBN 1-930934-12-2 Kitware.

Smith, W. L. and A. Fenster (2000). Optimum scan spacing for three-dimensional ultrasound by speckle statistics. *Ultrasound in Medicine and Biology* **26**(4), 551–562.

Stoll, J., P. Novotny, R. Howe and P. Dupont (2006). Real-time 3d ultrasound-based servoing of a surgical instrument. In: *IEEE Int. Conf. on Robotics and Automation, ICRA'2006*. Orlando, Florida.

Vitrani, M. A., G. Morel and T. Ortmaier (2005). Automatic guidance of a surgical instrument with ultrasound based visual servoing. In: *IEEE Int. Conf. on Robotics and Automation, ICRA'2005*. Barcelona, Spain.

Wallner, K., J. Blasko and M. Dattoli (2001). *Prostate brachytherapy made complicated*. Seattle (WA): SmartMedicine Press, 2nd edition.

Contents

1 Introduction **1**

2 Motion estimation **4**
2.1 In-plane motion estimation 4
2.2 Out-of-plane motion estimation 7
2.2.1 Speckle decorrelation technique 7
2.2.2 Plane estimation 8

3 Visual servoing **18**
3.1 Out-of-plane motion control 18
3.2 In-plane motion control 19

4 Simulation results **19**
4.1 Ultrasound imagery simulator 19
4.2 Stabilization robotic task results 19

5 Experimental results **27**
5.1 2-DOF motions compensation 27
5.2 6-DOF motions compensation 28

6 Conclusion **29**