



HAL
open science

SPIDAR calibration using Support Vector Regression

Pierre Boudoin, Hichem Maaref, Samir Otmane, Malik Mallem

► **To cite this version:**

Pierre Boudoin, Hichem Maaref, Samir Otmane, Malik Mallem. SPIDAR calibration using Support Vector Regression. 2nd International Conference on Image Processing Theory Tools and Applications (IPTA 2010), Jul 2010, Paris, France. pp.500–505, 10.1109/IPTA.2010.5586748 . hal-00639387

HAL Id: hal-00639387

<https://hal.science/hal-00639387v1>

Submitted on 3 Oct 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SPIDAR Calibration using Support Vector Regression

Pierre Boudoin, Hichem Maaref, Samir Otmame, Malik Mallem

¹ Laboratoire IBISC - Université d'Evry Val d'Essonne

e-mail: {firstname}.{name}@ibisc.fr

Abstract—This paper aims to present all the study done on the SPIDAR, which is a tracking and haptic device, in order to improve its accuracy on the given position. Firstly we proposed a new semi-automatic initialization technique for this device using an optical tracking system. Then, we propose to use Support Vector Regression (SVR) to calibrate the SPIDAR in order to reduce location errors. We obtained very good results with this calibration, since we reduced the mean error by more than 50%.

Keywords—calibration, support vector regression, SPIDAR, tracking, virtual reality,.

I. INTRODUCTION

Virtual reality is a domain which is highly dependant on tracking systems. Users interact in 3 dimensions, with virtual entities in digital environments. In order to provide the best user experience, it's very important that 3D interaction has to be without any interruption. This interaction relies on the translation of a real movement into an action in the virtual world. This work is done by a tracking solution. This tracking system has to be reliable and the most available as possible. This point is crucial in order to preserve data continuity and, so, data processing continuity and finally 3D interaction continuity.

The main device used in our system is an optical tracking solution, it's a very accurate device. On the other hand, it suffers from a huge defect: tracking-loss. That's a particular true defect when only one marker is used. So, it's essential to be able to switch to another device in these situations in order to compensate this defect. In our virtual reality system, we've got a SPIDAR [1] and we chose it to stand in for the optical tracking system.

SPIDAR [1], for *SPace Interaction Device for Augmented Reality*, is an electromechanical device, which has 8 couples of motor/encoder distributed on each vertex of a cubic structure. One string is attached to each motor via a pulley. These 8 strings converges to an effector. By winding their respective strings, each motor produces a tension on each string. The vectorial sum of these tensions produce the force feedback vector to be applied on the effector, allowing the user to feel on what he is stumbling or to feel the weight of an object. By observing the encoders values, the system can compute the 3D position of the effector. The SPIDAR tracking is always available, but it suffers from a weak accuracy and repeatability. So it's impossible to use raw position given by the SPIDAR without correction.

In our case, it's a huge problem, since we used a 3D interaction technique, called *Fly Over* [2], which needs a

continuous position vector. This technique is based on different interaction areas offering to the user a continuity in the interaction. Indeed, the least jump of position during the swing of a system, would be likely to pass the pointer of *Fly-Over* of a zone of interaction towards another. This phenomenon involves a behavior of the technique thus, not wished by the user and creating consequently a rupture of the continuity of the 3D interaction. Thus, it's important to propose measures in order to consider the position given by the SPIDAR so that it is closest to the position given by the optical tracking system, and so, minimizing effects on the 3D interaction.

This research work is presented as follow. First, we talk about similar works on virtual reality devices calibration and correction. Then, we briefly present the Support Vector Regression method. Finally, we speak about the correction of the SPIDAR position using SVR.

II. RELATED WORK

Since virtual reality systems use more and more devices, especially tracking devices, it's important to perform a good calibration of them. But not all tracking devices need a huge correction, thus infrared based optical tracking devices are accurate enough and so don't need to be corrected. On the other hand, it exists some mechanical, electromechanical or electromagnetic tracking devices which need to be calibrated and/or corrected.

Most of reseach works has been realized on the electromagnetic tracking devices because they suffers from electromagnetic distorsions when magnetical materials are placed into the tracking range. Moreover, the tracking accuracy falls off rapidly depending on the distance from the emitter and the power of the emitter [3]. These effects induce non-linear errors on the location. In order to correct them, it exists different ways.

The easiest method is the linear interpolation [4] but it doesn't correct non-linear systems, so it's very limited.

Polynomial fitting [5] allows to correct non-linear errors. But depending on the number of coefficients, it could be very difficult to use online because it will be a heavy load for the system. Moreover if the number of coefficient is too important, oscillations can appear, increasing errors rather than decreasing them. Moreover, these techniques often fail to capture small details in the correction. They are better in determining the overall shape of a non-linear function.

Kindratenko [6] and Saleh [7] worked on a neural network based calibration of electromagnetic tracking systems and they

obtained good results, better than with other methods.

An et Zan demonstrate the advantage of SVM for the identification of non-linear systems against the use of neural networks [8], [9]. They observed that learning time and results in generalization are better than these given by neural networks. But their results was for a mono-dimensionnal system.

Zhao used Smooth Support Vector Regression (SSVR) to reduce arithmetic complexity inherent to the SVR[10].

Cette réduction de la complexité permet d'appliquer les SVRs sur des jeux de données plus grands, car rappelez le, le temps d'apprentissage augmente fortement avec l'augmentation du nombre d'échantillons. Ils ont obtenus plus rapidement des meilleurs résultats avec leur méthode en l'appliquant sur un modèle d'état de turbine.

But all these techniques are based on interpolation and they need a valid set of data to be effective. This set of data highly is often given by a calibration grid. A calibration grid is a representation of a set of points. All these point have a known position and can be compared with the position given by the device that we want to calibrate. But when we're working in 3D space, it's very difficult to make use of it because it's difficult to place accurately a device on a 3D points. In order to realize that we can use another mechanical device, such a robot arm or a haptic arm [11]. Or we can place accurately passive sensors respecting a geometrical shape [12].

We focus on these methods because the SPIDAR suffers from same non-linear distorsions and 3D calibration problematic.

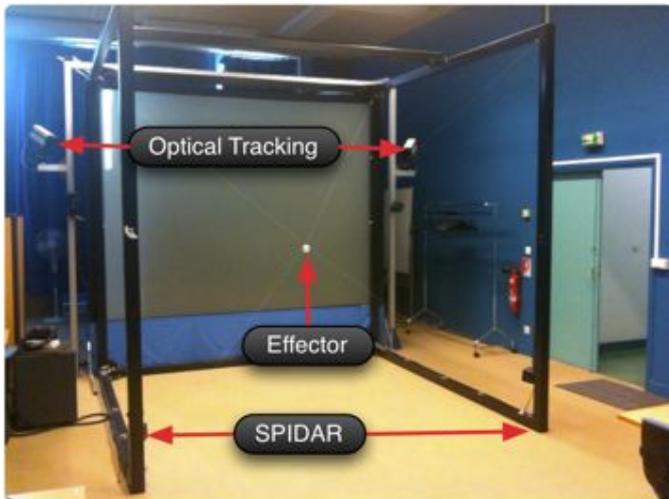


Fig. 1. SPIDAR and infrared based optical tracking in our virtual reality platform.

III. IDENTIFICATION OF THE SPIDAR

A. Context

In order to preserve the data continuity, it is essential to correct a well-known problem appearing with tracking systems: data loss. Data loss appears when the tracking is

unable to update the position calculation, conducting to a jump in the data when the system is re-enabled to update the position. This phenomena is often misled by occultation, especially in optical tracking system.

A data loss can be managed by three methods:

- 1) **Prediction:** We can predict the following data state by knowing the previous data state through mathematical method , such Kalman filter.
- 2) **Compensation:** A device tracking loss, don't forbid us to use another device. It's very important in this case that the data incoming from the different devices to be expressed in the same space representation (same referential). This is necessary in order to obtain a data continuity when the system switch from one device to another.
- 3) **Correction:** The last possibility is to correct data incoming from the most available device, in our case the SPIDAR. To perform the correction, we could use the a priori knowledge on the SPIDAR position through another device.

B. Design problems

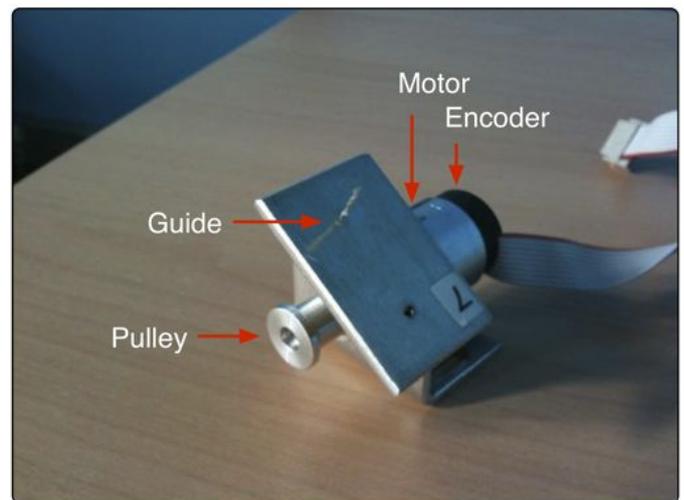


Fig. 2. Detailed view of a SPIDAR's motor and its winding guide.

SPIDAR is an electromechanical device and consequently it could suffer from design problems more or less awkward for computing the effector's position. These are problems we have identified:

- 1) **Encoders are directly mounted on the motor's axis.** This is an important problem because we must define the pulley's diameter in the configuration file of the SPIDAR's interface. However, this diameter is not constant, depending on the quantity of string winded. So, this information is skewed.
- 2) **Diameter of pulleys is too small.** The previous problem become more marked due to the small diameter of the pulley used. Thus, the diameter

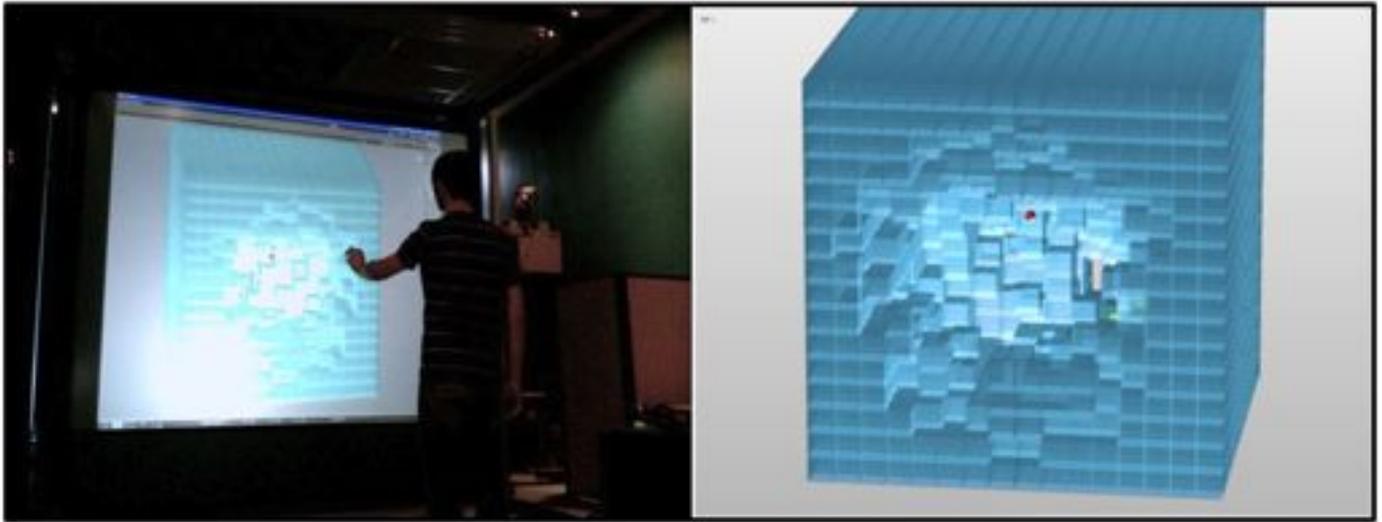


Fig. 3. On the left - A user using our virtual calibration grid in order to retrieve data for the SVR learning. On the right - Detailed representation of the virtual calibration grid.

being too small, it varies noticeably as strings being wound go along. This phenomena would be less marked if the diameter used was more important.

3) **Winding guides badly designed.**

The present design of the winding guides, don't prevent a string from missing the pulley. This phenomena appears when the effector is being moved fast and consequently, that motors have to wind an important quantity of string. This is a real problem, because the encoder count one revolution but the string doesn't be wound.

4) **Size of encoders.**

Encoders' size is too small for counting the string quantity which must be wound. When an encoder overflows, the counter is resetted and the wound string quantity information is biased.

5) **Dimensions of the SPIDAR.**

More dimensions are important and more every problem cited previously is marked. Some problems that are inconsiderable when dimensions are small, become not inconsiderable when dimensions are huge.

C. *Experimental protocol*

We use what we called: a virtual calibration grid (see fig.3), which consists in the representation of a virtual scene, composed of many small cubes. Each cube corresponds to a sub-space of the SPIDAR workspace. This set of small cubes covers the whole SPIDAR workspace.

The use of virtual reality for calibration allows more flexibility and less complexity because we don't have to move the SPIDAR effector with constraints or to place the effector with a great accuracy on a set of calibration point.

This calibration grid, is represented Fig.3. We can identify the SPIDAR's problem with it, following these steps:

- 1) The user move the real effector (which is in his hand) in order to place the virtual effector (which is a red sphere in the virtual scene) in each cube represented.
- 2) Each time the virtual effector is in collision with a cube, we record the position given by the SPIDAR and the position given by the optical tracking.
- 3) Once these positions are recorded the cube disappears insuring that there will be only one point for this sub-space.

The great advantage of this protocol is the homogeneity distribution of the data set.

We recorded values, respecting this protocol in a workspace limited to $1 m^3$ splitted into 4096 sub-spaces ($16 \times 16 \times 16$).

D. *Identification Conclusion*

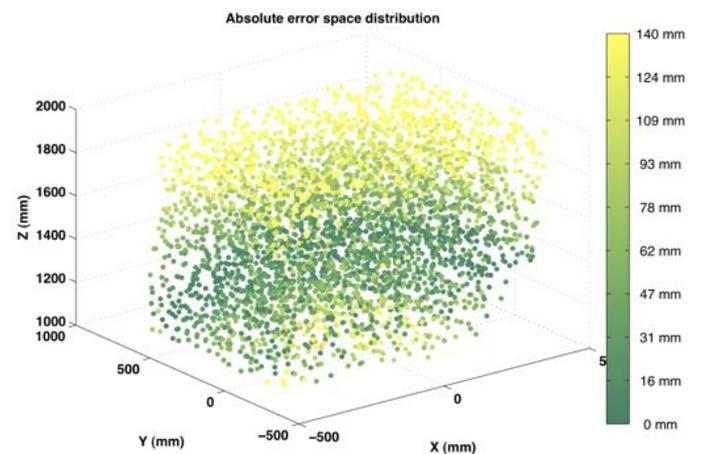


Fig. 4. Absolute error 3D distribution in the SPIDAR's workspace (Dark green is the best).

Figure 4 represents errors' space distribution. As we can see this is a onion skin distribution, meaning different spherical layers, the absolute error growing as the effector is going in outside layers.

This identification of the SPIDAR leads us to these observations:

- Firstly, the mechanical study tells us that too many design problems
- Moreover, it's hard to quantify the final effect of these mechanical problems.
- Another problem, is the loss of knowledge towards the mathematical model used by the SPIDAR to compute the effector's position.

Finally, the SPIDAR suffers from a set of problems, which have more or less known causes and for which we don't know very well the influence on the whole system. In order to enhance the SPIDAR's accuracy, it could be interesting to orient oneself to a solution capable of estimating/correcting the effector's position without any knowledge on the mathematical model. We choose to test that way using Support Vector Regression for their capacities to estimate a regression function without any information on the model.

IV. SUPPORT VECTOR REGRESSION

Support Vector Regression (SVR) is the Support Vector Machine applied to the regression problem [13]. This part briefly introduce how SVR works.

Firstly, we need to remind the classical regression problematic. This problem consists in the estimation of a parameter vector $\omega = [\omega_1, \omega_2, \dots, \omega_m]^T$ belonging to an unknown function $f(x_i, \omega)$. Only things we know about this function are the observed data y_i . So we have:

$$y_i = f(x_i, \omega) + e_i, \quad i = 1, 2, \dots, N \quad (1)$$

Where:

- x_i are input data,
- e_i are errors,
- N is the size of data.

The pair (x_i, y_i) define the training set, this represents the whole input data associated with their respective output values.

The regression function which approximate the unknown function is defined by:

$$f(x_i, w) = \sum_{j=1}^m \omega_j g_j(x_i) \quad (2)$$

Where:

- g_i define a set of nonlinear transformations.

In order to use SVM in a regression problem, a new type of loss function has been defined by Vapnik. This loss function is called ε -insensitive and it's defined by:

$$L_\varepsilon(y_i, f(x_i, \omega)) = \begin{cases} 0, & \text{if } |y_i - f(x_i, \omega)| \leq \varepsilon \\ |y_i - f(x_i, \omega)| - \varepsilon, & \text{else} \end{cases} \quad (3)$$

With:

ε controls the number of support vectors and determines the width of the insensitive zone.

First problem:

The first problem spotted by the SVR is to find the parameter vector ω , which minimizes the empirical risk function R_{emp} .

$$\text{Min } R_{emp}(\omega) = \frac{1}{N} \sum_{i=1}^N L_\varepsilon \left(y_i, \sum_{j=1}^m \omega_j g_j(x_i) \right) \quad (4)$$

With the constraint:

$$(\omega \cdot \omega) \leq c_k \quad (5)$$

This problem can be transposed into another problem where parameters α_i^* and β_i^* must maximize the function $Q(\alpha, \beta)$:

$$\text{Max } Q(\alpha, \beta) = -\varepsilon \sum_{i=1}^N (\alpha_i + \beta_i) + \sum_{i=1}^N y_i (\alpha_i - \beta_i) - \frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \beta_i)(\alpha_j - \beta_j) K(x_i, x_j) \quad (6)$$

With constraints:

$$\sum_{i=1}^N \alpha_i = \sum_{i=1}^N \beta_i, \quad 0 \leq \alpha_i, \beta_i \leq C, \quad i = 1, \dots, N \quad (7)$$

Where:

- K is a kernel function,
- C is a regularization parameter,
- ε defines an insensitive zone.

This second problem can be resolved using quadratic programming (QP) which gives us the α_i^* and β_i^* , allowing us to construct the regression function:

$$f(x) = \sum_{i=1}^N (\alpha_i^* - \beta_i^*) K(x_i, x) \quad (8)$$

V. CALIBRATING THE SPIDAR USING SVR

A. SVR learning

For the learning step, we use the data used for the SPIDAR identification. So, we've got 4096 measure points covering the whole SPIDAR workspace.

B. SVR settings

In the support vector regression method, two parameters play an important role for the smoothness and the noise of the regression function. These parameters are ε and C . These parameters are chosen *a priori*, in an empirical way.

We find that we've got the best performance with ε -SVR method when: $\varepsilon = 0.1$ and $C = 1$.

C. SVR performances

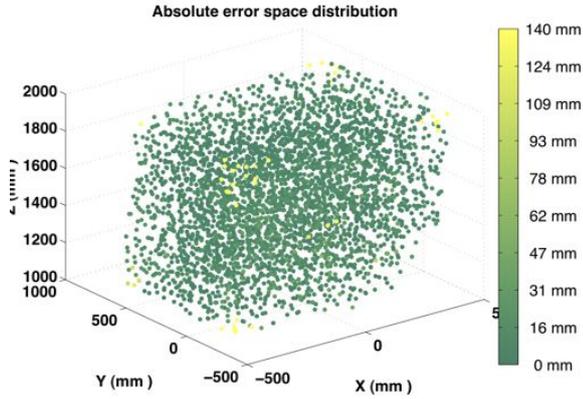


Fig. 5. Mean absolute error representation in the SPIDAR's workspace - full error range (Dark green is the best).

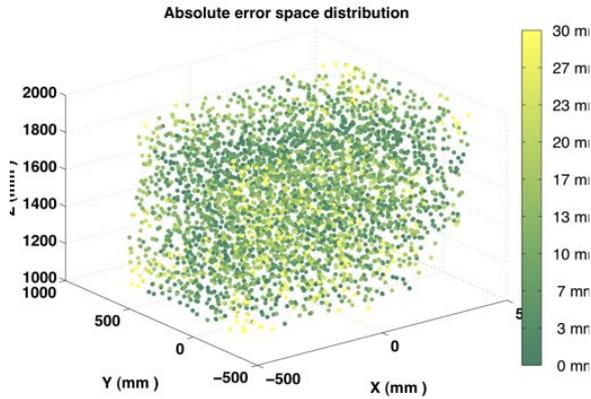


Fig. 6. Mean absolute error representation in the SPIDAR's workspace - reduce error range (Dark green is the best).

In order to study the SVR performances, we represent the same absolute error 3D spatial distribution as previously but after calibration using SVR. As we can see, figures 5 and 6, shows that SVR is quite effective and greatly improve the SPIDAR accuracy in comparison with the figure 4. Support Vector Regression performs a good calibration in the whole workspace except in its corners. Fig.7 shows that after calibration the SPIDAR accuracy is less dependant of the distance from its middle.

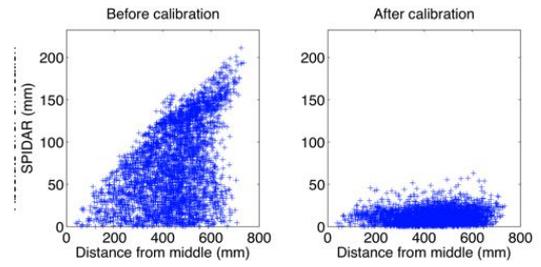


Fig. 7. Absolute error on location versus distance from the middle of the SPIDAR before and after calibration.

We also give errors bar graphs before and after using SVR (see Fig.8) and we put representative data in arrays comparing them with raw location (Raw), SVR calibrated location (SVR) and linear interpolation calibrated location (PF1) for information purpose only. Each bar graph is coupled with an array resuming characteristic values of the error distribution, where *mean* is the empirical mean error, *std* is the standard deviation of the data set and *max* is the maximum error.

As we can see with Fig.8 and table V-C, before calibration (Raw) the mean absolute error reaches 72.87 mm and the standard deviation is 47.06 mm , which mean that the raw SPIDAR position suffers of a bad accuracy and an important statistical dispersion. After calibration using SVR the mean absolute error is below 12 mm and the standard deviation is equal to 6 mm , meaning a higher accuracy and a lower statistical dispersion.

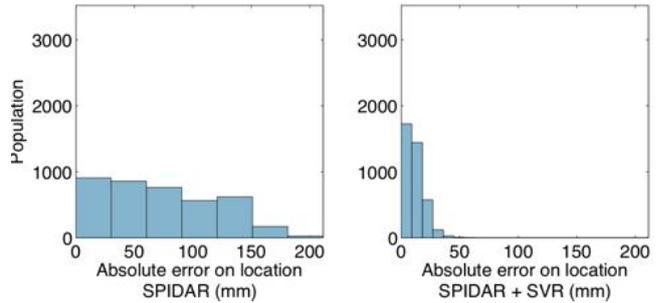


Fig. 8. Absolute errors distribution bar graph before and after calibration by the SVR method.

Absolute error	Raw	SVR	PF1
Mean (mm)	72.87	11.47	15.31
Std (mm)	47.06	6.00	40.75
Max (mm)	155.19	42.19	198.32

D. Generalization

In order to evaluate SVR performances, we need to observe this response output with unknown data sets. This step is called *generalization*. Figures 9 and 10 show results obtained with two generalization data sets. These data sets have been recorded during two time-splitting measure campaigns.

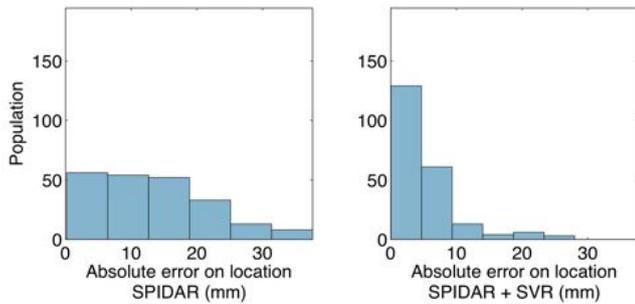


Fig. 9. Absolute errors distribution bar graph before and after calibration using SVR on the 1st data set.

Absolute error	Raw	SVR	PFI
Mean (mm)	13.23	5.06	9.67
Std (mm)	8.41	5.14	7.53
Max (mm)	37.60	24.04	48.25

1) *Results on the 1st data set:* We can observe that the SVR method has a good response to the generalization. Thus, the absolute mean error on the position has been reduced by 2.5 times, going from 13.3 mm to 5.06 mm.

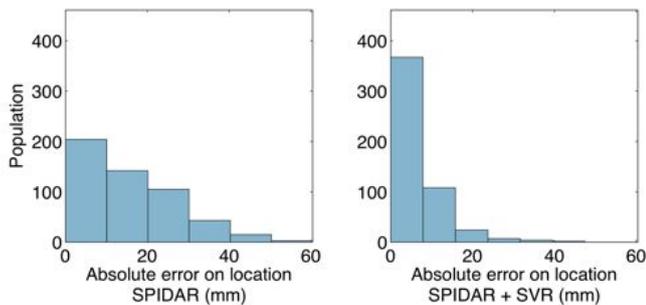


Fig. 10. Absolute errors distribution bar graph before and after calibration using SVR on the 2nd data set.

Absolute error	Raw	SVR	PFI
Mean (mm)	15.79	6.59	9.67
Std (mm)	11.64	6.45	7.53
Max (mm)	54.53	47.50	48.25

2) *Results on the 2nd data set:* With this data set, the use of SVR keeps good performances since the absolute mean error has been reduced by a bit more than a half.

3) *Interpretation:* Calibration using SVR is robust at any data as soon as the data belongs to the SPIDAR's workspace. Such results aren't surprising since the learning protocol was extremely rigorous and covered all the SPIDAR's workspace guaranteeing an efficient learning, so SVR method was able to perform a good calibration.

VI. CONCLUSION

In this paper we propose a method to calibrate SPIDAR using Support Vector Regression (SVR), which is a quite new

method for doing system identification. We also proposed a rigorous learning protocol using virtual reality to perform what we call a *virtual calibration grid* exempting us to constraint movement in order to perform the calibration of a 3D tracking device. We obtain good results and our whole calibration procedure is quite robust even if we reset the SPIDAR.

REFERENCES

- [1] Seahak Kim, Masahiro Ishii, Yasuharu Koike, and Makoto Sato, "Development of tension based haptic interface and possibility of its application to virtual reality", *Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 199–205, 2000.
- [2] Pierre Boudoin, Samir Otmame, and Malik Mallem, "Fly over, a 3d interaction technique for navigation in virtual environments independent from tracking devices", *Proceeding of the Virtual Reality International Conference*, pp. 7–13, Feb 2008.
- [3] M Nixon, B McCallum, W Fright, and N Price, "The effects of metals and interfering fields on electromagnetic trackers", *Presence*, vol. 7, no. 2, pp. 204–218, Jan 1998.
- [4] M Ghazisaedy and D Adamczyk, "Ultrasonic calibration of a magnetic tracker in a virtual reality space", *Proceedings of the Virtual Reality Annual International Symposium*, Jan 1995.
- [5] M Ikits, J Brederson, Charles D Hansen, and John M Hollerbach, "An improved calibration framework for electromagnetic tracking devices", *Proceeding of the IEEE Virtual Reality Conference*, Jan 2001.
- [6] VV Kindratenko, "A survey of electromagnetic position tracker calibration techniques", *Virtual Reality*, vol. 5, no. 3, pp. 169–182, 2000.
- [7] T Saleh, V Kindratenko, and W Sherman, "On using neural networks to calibrate electromagnetic tracking systems", *Proceedings of the IEEE Virtual Reality annual international symposium*, Jan 2000.
- [8] J An, Q Yang, and Z Ma, "Study on svm on-line function regression method for mass data", *Machine Learning and Cybernetics*, vol. 5, pp. 2773–2777, Jan 2007.
- [9] M Zhang, W Yan, and Z Yuan, "Study of nonlinear system identification based on support vector machine", *Proceedings of the International Conference on Machine Learning and Cybernetics, 2004*, vol. 5, pp. 3287–3290, Jan 2004.
- [10] C Zhao and P Han, "Model identification of thermal object based on smooth support vector regression", *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition*, vol. 3, no. 1388–1391, Jan 2007.
- [11] M Ikits, C Hansen, and C Johnson, "A comprehensive calibration and registration procedure for the visual haptic workbench", *Proceedings of the workshop on Virtual environments*, vol. 39, pp. 247–254, Jan 2003.
- [12] D Leotta, "An efficient calibration method for freehand 3-d ultrasound imaging systems", *Ultrasound in medicine & biology*, Jan 2004.
- [13] V Vapnik, "The nature of statistical learning theory", *Springer-Verlag New York, Inc. New York, NY, USA*, Jan 2000.