A Novel Approach to Vision-Based Computed Torque Control of Parallel Robots

Omar Ait-Aider Flavien Paccot Nicolas Andreff Philippe Martinet

Abstract— **This paper proposes a novel vision sensing method to be used in vision-based dynamic identification and control of parallel robots. Indeed, it is shown that in the latter problem one requires to estimate (to the least) or measure (to the best) the end-effector pose and its time derivatives. The sensor takes advantage of the image deformations induced by rolling shutter in CMOS image sensors. We derive a method compensating for the rolling shutter deformations to deliver an accurate 3D pose and exploiting them to also estimate the full 3D velocity transforming the CMOS camera into an exteroceptive pose and velocity sensor. We present control schemes which illustrate how the developed visual sensor is particularly adapted for parallel mechanisms dynamic control. Experimental results with real data confirm the relevance of the approach and show the sensor good practical measurement accuracy.**

I. Introduction

This work spins off from parallel robot dynamic control. Indeed, it has been reported that high-speed vision could be used for dynamic control of serial robots [4]. Hence, we initially wanted to study whether vision-based control can also be applied to parallel manipulators. To do so, one first requires to identify the parameters of the dynamical model used for control [11]. Since vision has proven [19] an accurate tool for identifying the kinematic parameters of parallel robots, it may also be useful for dynamic identification.

However, using vision sensors in this framework requires the ability to capture clear images of objects undergoing high velocity motion without any distortion, blur nor smear. To achieve this task, there is a need for image sensors enabling very short exposure time of all the matrix pixels simultaneously. This functionality requires a particular electronic design that is not included in all camera devices. Indeed, Full Frame CCD sensors, without storage memory areas, require mechanical obturator or stroboscopic light source, introducing more complexity in the vision system. Similarly, Frame Transfer CCD sensors may not reach the desired frame rate or may be costly because of additional sillicium in storage areas [23]. Alternately, Standard CMOS Rolling Shutter sensors are considered as low cost and low power sensors. They are becoming more frequently used in cameras. They enable adequate exposure time without reducing frame rate thanks to overlapping exposure and readout. Their drawback is that they distort images of moving objects because the pixels are not all exposed simultaneously but row by row with a time delay defined by the sensor technology. This distortion may represent a major obstacle in tasks such as localization, reconstruction or default detection (the system may see an ellipse where in fact there is a circular hole). Therefore, CMOS Rolling Shutter cameras could offer a good compromise between cost and frame rate performances if the problem of deformations is taken into account.

Coming back to frame rate, the current affordable technology allows to reach very high frame rates by reducing the sensor resolution (about 250×250 pixels). The latter resolution reduces a lot the field of view of the camera for a given accuracy of the end-effector pose estimation [3], [12]. This is very restrictive since the workspace available for identification, which should be as large as possible to provide good excitation and high result accuracy [5], [21], is consequently much reduced. If one tries to enlarge the field of view, the frame rate falls down and the rolling shutter phenomenon becomes more perceptible. Faced with this huge drawback, we turned it into an advantage since this mere effect acts as an analog time derivator. With adequate modelling, we can thus propose a method where both the full end-effector pose and the full Cartesian end-effector velocity can be estimated simultaneously with a single image. Consequently, this not only suppresses the need for numerically inverting the inverse kinematic model to estimate the end-effector pose from the joint values, but also suppresses (or makes it more robust) the numerical time derivation of the end-effector pose in the dynamical model.

One may argue that doing so, we reduced the frame rate and thus impaired the dynamic identification which uses high-frequency signals. Nevertheless, let us recall that these signals are mainly used to be filtered and sub-sampled (down to a 100Hz) in order to make the identification process robust to high-frequency noise. If we can reach a 100Hz frame rate with 1 Mpixel resolution and rolling shutter, the measured end-effector poses and velocities should have an equivalent signal-to-noise ratio.

All pose recovery methods in the vision community literature ([13], [22], [3], [14], [24]) make the assumption that all image sensor pixels are exposed simultaneously. The work done by Wilburn et al. [26] concerned the correction of image deformation due to rolling shutter by constructing a single image using several images from a dense camera array. Using the knowledge of the time delay due to rolling shutter and the chronograms of release of the cameras, one complete image is constructed by combining lines exposed at the same instant in each image from the different cameras. In [16] Meingast et al. modelled the projection in rolling shutter cameras with

Université Blaise Pascal Clermont Ferrand, LASMEA UMR 6602 CNRS firstname.lastname@univ-bpclermont.fr, WWW home page: http://www.lasmea.univ-bpclermont.fr, Tel: +334 73 40 72 45, Fax: +334 73 40 72 62. 24, Avenue des Landais 63172 Aubière France.

some approximations producing equations which are similar to those of Crossed-Slits cameras in the case of fronto-parallel motion. In [1], we presented an exact projection model for the perspective projection with rolling shutter and an algorithm for pose and velocity computation from a single view exploiting rolling shutter distortions.

In this paper we show how using a camera as an exteroceptive pose and velocity sensor is relevant for parallel mechanism dynamic control (and identification). Indeed, the approach may be used to avoid estimating the kinematic between successive views reducing the amount of data and the computational cost (one image is processed rather than several ones). Furthermore, it supresses the numerical derivation (and consequently signal filtering) classicaly used in velocity measurement. In section II, improvements expected from using a cartesian exteroceptive pose and velocity sensor for parallel mechanisms dynamic identification and control are discussed. In section III the process of image acquisition using a CMOS Rolling Shutter imager and a model for perspective projection of solid moving object are first described. Then, the problem of computing pose and velocity parameters is developed. Finally, experiments with real data are presented and analyzed in section IV.

II. WHY VISION IS WORTH COUPLING WITH DYNAMIC IDENTIFICATION AND COMPUTED TORQUE CONTROL IN PARALLEL ROBOT CASE ?

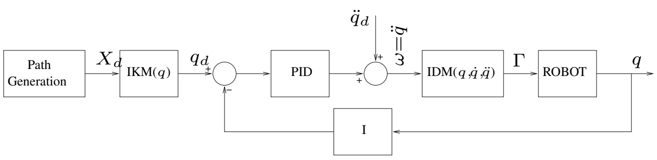

Fig. 1. *Computed Torque Control in joint space for serial robot [11]*

Implementing vision sensor in classical Computed Torque Control scheme could be a great improvement for parallel robot. Joint-space CTC scheme predominates for serial robot. It is presented in Figure 1, where inverse dynamic model is a function of joint variables and their time derivatives:

$$IDM(q, \dot{q}, \ddot{q}) = A(q)\ddot{q} + h(q, \dot{q}) \quad (1)$$

Parallel robot control is generally based on this scheme [8], [2]. This type of servoing seems to be unsuitable in this case. Indeed, controlling each actuators separately can create internal torques, because joint trajectory compatibility is not ensured. It can cause damage on structure. Servoing joint position instead of end-effector position does not ensure obtaining the desired platform pose, as explained in Figures 2 and 3. Actually, one joint configuration leads to several end-effector position [17], [9], so a pertubation can shift platform pose without changing joint variable configuration.

Serial and parallel robot are dual. Namely, parallel mechanisms have models expressed as a function of end-effector pose and its time derivatives instead of joint variable in serial

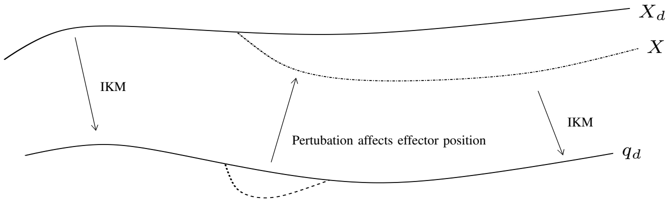

Fig. 2. *Joint-control can lead to loose desired cartesian trajectory tracking*

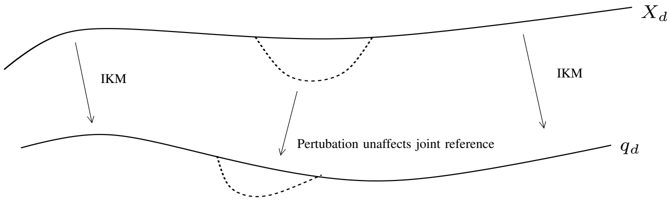

Fig. 3. *Cartesian control ensure a good tracking*

case. So used models in CTC (Figure 1) are expressed as $IKM(X)$ for inverse kinematic model and $IDM(X, \dot{X}, \ddot{X})$ [8], [10] where:

$$IDM(X, \dot{X}, \ddot{X}) = A(X)\ddot{X} + h(X, \dot{X}) \quad (2)$$

We can rewrite them as function of joint-variable, using :

$$\dot{X} = D_{inv}^{-1}(X)\dot{q} \quad (3)$$

and

$$\ddot{X} = D_{inv}^{-1}(X)\ddot{q} + \dot{D}_{inv}^{-1}(X, \dot{X})\dot{q} \quad (4)$$

But, $D_{inv}^{-1}(X)$ has to be computed. So these numerical transformations make classical methods based on joint position unsuitable for identification and CTC.

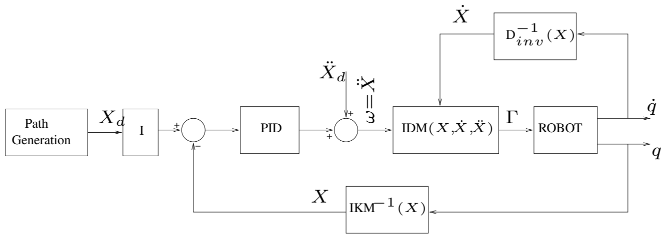

Fig. 4. *Computed Torque Control in cartesian space for parallel robot [11]*

As said, serial robot and parallel robot are dual, so why not using CTC in cartesian space (see Figure 4). This servoing is not common for serial robot, because a transformation between end-effector acceleration and joint-variable acceleration is needed , asking heavy computation:

$$\ddot{q} = D^{-1}(q)(\ddot{X} - \dot{D}(q)\dot{q}) \quad (5)$$

In parallel robot case, this transformation is useless since inverse dynamic model is a function of effector acceleration. So cartesian control seems to be more natural. As torques are computed in regards with effector position error, joints have only compatible moves and there are no internal torques. In addition, tracking desired cartesian pose is ensured in presence of perturbation (see Figure 3).

But there is still a default: the measure of end-effector pose. For parallel robots, end-effector pose is computed from joint variable measures, with the forward kinematic model, contrary to serial robot whose FKM is analytically defined. Obtained by a non-linear optimization, this numerical model leads to multiple solutions (40 for a Gough-Stewart platform [9]) and is influenced by singularities. This issue results in a lack of accuracy and a decrease in servoing speed. To get rid with this, special achitectures with analytical forward kinematic model have been developped (Orthoglide [25], T3R1 [7]). In general case, pose measurement can be done using redundant informations obtained with exteroceptive measurement means [15]. An other point is the end-effector velocity determination, by derivation of computed poses or by mean of forward jacobian, which is numerical is parallel robot case. This measures decrease time perfomance and accuracy.

The proposed vision sensing method could resolve these problems by allowing a measure of end-effector pose and velocity, with more precision and easier implementation than other exteroceptive sensors, for instance an expensive laser tracker. Consequently, it seems to be perfectly relevent for dynamic identification with high accuracy than classical methods [6], [18], [20]. If a high sampling rate can be obtained (1 kHz), computed torque control could be archieved with proposed control scheme in Figure 5, high performances and accuracy could be expected. In this conditions, sensors on actuators would be useless.

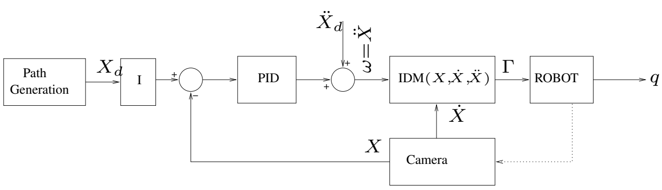

Fig. 5. *Computed Torque Control with proposed vision sensor*

III. Pose and velocity computation using a single rolling shutter image

A. Definition of rolling shutter image acquisition mode

In digital cameras, an image is captured by converting the light from an object into an electronic signal at the photosensitive area (photodiode) of a solid state CCD or CMOS image sensor. The amount of signal generated by the image sensor depends on the amount of light that falls on the imager, in terms of both intensity and duration. Therefore, an on-chip electronic shutter is required to control exposure. The pixels are allowed to accumulate charge during the integration time. With a CMOS image sensor with rolling shutter, the rows of pixels in the image are reset in sequence starting at the top and proceeding row by row to the bottom. The readout process proceeds in exactly the same fashion and the same speed with a time delay after the reset (Fig.6). In this case, if the object is moving during the integration time, some artifacts may appear.

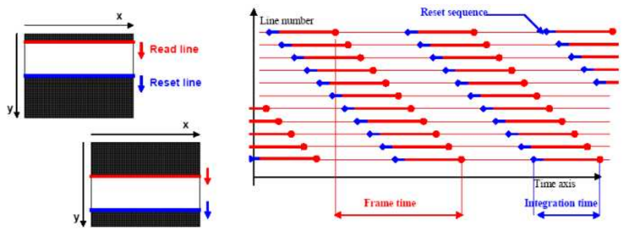

Fig. 6. Reset and reading chronograms in rolling shutter sensor (SILICON IMAGING documentation).

B. Projecting a point with a rolling shutter camera

Let us consider a classical camera with a pinhole projection model defined by its intrinsic parameter matrix [24]. Let $\vec{P} = [X, Y, Z]^{\mathrm{T}}$ be a 3D point and $\vec{R}$, $\vec{T}$ the rotation matrix and the translation vector between the object and the camera frames. The homogeneous perspective projection of $\vec{P}$ on the image is $s\tilde{\vec{m}} = \vec{K}\begin{bmatrix}\vec{R} & \vec{T}\end{bmatrix}\tilde{\vec{P}}$, where s is an arbitrary scale factor.

Assume now that an object of a known geometry modelled by a set of n points $\vec{P_i} = [X_i, Y_i, Z_i]^{\mathrm{T}}$, undergoing a motion with instantaneous angular velocity Ω around an instantaneous axis of unit vector $\vec{a} = [a_x, a_y, a_z]^{\mathrm{T}}$, and instantaneous linear velocity $\vec{V} = [V_x, V_y, V_z]^{\mathrm{T}}$, is snapped with a rolling shutter camera at an instant t_0. Thus, the light from the point $\vec{P_i}$ will be collected with a delay τ_i proportional to the image line number on which $\vec{P_i}$ is projected. Therefore, to obtain the projection $\vec{m}_i = [u_i, v_i]^{\mathrm{T}}$ of $\vec{P_i}$, the pose parameters are corrected by integrating the motion during the time delay τ_i. Since all the lines have the same exposure and integration time, we have $\tau_i = \tau v_i$ where τ is the time delay between two successive image line exposure. Thus $\tau = \frac{fp}{v_{max}}$ where fp is the frame period and v_{max} is the image height. Assuming that τ_i is short enough to consider uniform motion during this interval, the object rotation during this interval is obtained thanks to the Rodrigues formula:

$$\delta\vec{R_i} = \vec{a}\vec{a}^T\left(1 - cos\left(\tau v_i \Omega\right)\right) + \vec{I}cos\left(\tau v_i \Omega\right) + \hat{\vec{a}}sin\left(\tau v_i \Omega\right)$$

where $\vec{I}$ is the 3×3 identity matrix and $\hat{\vec{a}}$ the antisymetric matrix of $\vec{a}$. The translation during the same interval is:

$$\delta\vec{T_i} = \tau v_i \vec{V}$$

Thus, the projection equation is rewritten as follows:

$$s\,\tilde{\vec{m}_i} = \vec{K}\begin{bmatrix}\vec{R}\delta\vec{R_i} & \vec{T} + \delta\vec{T_i}\end{bmatrix}\tilde{\vec{P_i}} \tag{6}$$

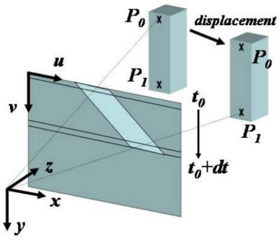

Fig. 7. Perspective projection of a moving 3D object: due to the time delay, points $\vec{P}_0$ and $\vec{P}_1$ are not projected from the same object pose

where $\vec{R}$ and $\vec{T}$ represent now the instantaneous object pose at t_0. Equation 6 is the expression of the projection of a 3D point from a moving solid object using a rolling shutter camera with respect to object pose, object velocity and the parameter τ.

C. Computing the instantaneous pose and velocity of a moving object

In this section, we assume that a set of rigidly linked 3D points $\vec{P}_i$ on a moving object are matched with their respective projections $\vec{m}_i$ measured on an image taken with a rolling shutter camera. We want to use this list of 3D-2D correspondences to compute the instantaneous pose and velocity of the object at instant t_0. The scale factor of equation 6 can be removed as follows:

$$\begin{array}{l} u_i = \alpha_u \frac{\vec{R^{(1)}}_i \vec{P}_i + T^{(x)}}{\vec{R^{(3)}}_i \vec{P}_i + T^{(z)}} + u_0 = \xi_i^{(u)}\left(\vec{R}, \vec{T}, \Omega, \vec{a}, \vec{V}\right) \\ v_i = \alpha_v \frac{\vec{R^{(2)}}_i \vec{P}_i + T^{(y)}}{\vec{R^{(3)}}_i \vec{P}_i + T^{(z)}} + v_0 = \xi_i^{(v)}\left(\vec{R}, \vec{T}, \Omega, \vec{a}, \vec{V}\right) \end{array} \tag{7}$$

where $\vec{R^{(j)}}$ and $T^{(x,y,z)}$ are respectively the j^{th} row of $\vec{R}_i = \delta\vec{R}_i\vec{R}$ and the components of $\vec{T}_i = \vec{T} + \delta\vec{T}_i$. Subsiding the right term from the left term and substituting u_i and v_i by image measurements, equation 7 can be seen as an error function with respect to pose and velocity (and possibly τ) parameters:

$$\begin{array}{l} u_i - \xi_i^{(u)}\left(\vec{R}, \vec{T}, \Omega, \vec{a}, \vec{V}\right) = \epsilon_i^{(u)} \\ v_i - \xi_i^{(v)}\left(\vec{R}, \vec{T}, \Omega, \vec{a}, \vec{V}\right) = \epsilon_i^{(v)} \end{array}$$

We want to find $\vec{R}$, $\vec{T}$, Ω, $\vec{a}$ and $\vec{V}$ that minimize the following error function:

$$\epsilon = \sum_{i=1}^{n} \left[u_i - \xi_i^{(u)}\left(\vec{R}, \vec{T}, \Omega, \vec{a}, \vec{V}\right)\right]^2 + \left[v_i - \xi_i^{(v)}\left(\vec{R}, \vec{T}, \Omega, \vec{a}, \vec{V}\right)\right]^2$$

This problem with 12 independent unknowns can be solved using a non-linear least square optimization if at least 6 correspondences are available. This can be seen as a bundle adjustment with a calibrated camera.

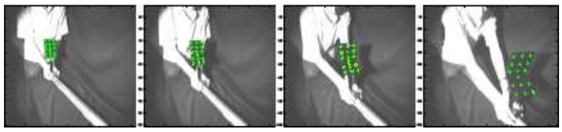

Fig. 8. Image samples of pure translational motion.

IV. EXPERIMENTS

The aim of this experimental evaluation is first to illustrate our pose recovery algorithm accuracy in comparison with classical algorithms under the same acquisition conditions, and second, to show its performances as a velocity sensor. The algorithm was tested on real image data. A reference 3D object with white spots was used. Sequences of the moving object at high velocity were captured with the Silicon Imaging CMOS Rolling Shutter camera SI1280M-CL, calibrated using the method described in [12]. Acquisition was done with a 1280×1024 resolution and at a rate of 30 frames per second so that $\tau = 7.15 \times 10^{-5}$ s. The pose and velocity parameters were computed for each image using first our algorithm, and compared with results obtained using the classical pose recovery algorithm described in [12] where an initial guess is first computed by the algorithm of Dementhon [3] and then the pose parameters are accurately estimated using a bundle adjustment technique.

Figure 8 shows image samples from a sequence where the reference object was moved following a straight rail forcing its motion to be a pure translation. In the first and last images of the sequence the object was static. Pose parameters corresponding to these two views were computed accurately using the classical algorithm. The reference object trajectory was then assumed to be the 3D straight line relating the two extremities. Pose recovery results are shown in figure 9. The left-hand side of this figure shows 3D translational pose parameters obtained by our algorithm and by the classical algorithm (respectively represented by square and *-symbols). Results show that the two algorithms give appreciably the same results with static object views (first and last measurements). When the velocity increases, a drift appears in the classical algorithm results while our algorithm remains accurate (the 3D straight line is accurately reconstructed by pose samples) as it is illustrated on Table I where are represented distances between computed poses with each algorithm and the reference trajectory. Table II presents computed rotational pose parameters. Results show the deviation of computed rotational pose parameters from the reference orientation. Since the motion was a pure translation, orientation is expected to remain constant. As one can see, a drift appears on classical algorithm results while our algorithm results show a very small deviation due only to noise on data.

Another result analysis concerns the velocity parameters. Figure 9 shows that the translational velocity vector is clearly parallel to translational axis (up to noise influence). Table III represents magnitude of computed velocity vectors in comparison with measured reference values. Results show that

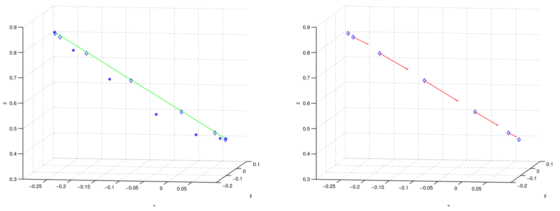

Fig. 9. Pose and velocity results: reconstructed trajectory (top image), translational velocity vectors (bottom image).

TABLE I
DISTANCES FROM COMPUTED POSES TO REFERENCE TRAJECTORY (CM).

Image number	1	2	3	4	5	6	7
Classical algorithm	0.00	0.19	0.15	1.38	3.00	4.54	0.00
Our algorithm	0.18	0.24	0.26	0.22	0.32	0.11	0.07

the algorithm recovers correctly acceleration, deceleration and static phases. Note that computed rotational velocities are small and only due to noise.

The algorithm was also tested with coupled rotational and translational motions (Figure 10). Equivalent results were observed. The manifold of instantaneous rotation axis vectors was correctly oriented. The mean value of the angles between the computed rotation axis and $\vec{N}$ was 0.50 degrees. Results in table IV shows a comparison of the computed rotational velocity magnitudes and the measured values.

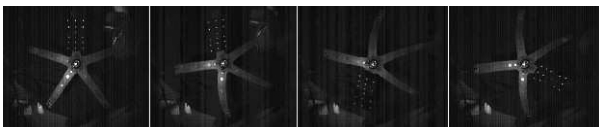

Fig. 10. Image samples of coupled rotational and translational motions.

V. CONCLUSION AND PERSPECTIVES

An original method for computing pose and instantaneous velocity of rigid objects using a single view from a rolling shutter camera was presented. The approach was evaluated on real data showing its feasibility. The method is not less accurate than similar classical algorithms in case of static objects, and improves pose accuracy in case of fast moving objects. In addition, the method gives the instantaneous velocity parameters using a single view. For instance, the approach can enable us to avoid numerical derivation for velocity estimation from several poses. In addition, its implementation offers more flexibility to measure the full 3D pose and velocity parameters. These properties make it an original tool for many fields of research in robotics. We discussed improvements expected from using this method as a cartesian exteroceptive pose and velocity sensor for parallel mechanisms dynamic identification and control.

Our future work, will concern the implementation of the presented method on a real parallel robot.

TABLE II
ANGULAR DEVIATION OF COMPUTED POSES FROM REFERENCE ORIENTATION (DEG.).

Image number	1	2	3	4	5	6	7
Dementhon's algo.	0.00	2.05	4.52	6.93	6.69	3.39	0.30
our algorithm	0.07	0.13	0.15	0.24	0.90	0.91	0.40

TABLE III
COMPUTED TRANSLATIONAL VELOCITY MAGNITUDE IN COMPARISON WITH MEASURED VELOCITY VALUES (M/S)

Image number	1	2	3	4	5	6	7
Measured values	0.00	1.22	2.02	2.32	1.55	0.49	0.00
Computed values	0.06	1.10	1.92	2.23	1.54	0.50	0.02

TABLE IV
COMPUTED AND MEASURED ROTATIONAL VELOCITY MAGNITUDES (RAD/S)

Image number	1	2	3	4	5	6	7	8	9	10
Measured values	0.00	1.50	9.00	11.20	10.50	10.20	10.10	10.00	10.00	7.50
Computed values	0.06	1.20	8.55	10.38	10.32	10.30	9.80	9.90	9.73	8.01

REFERENCES

[1] O. Ait-Aider, N. Andreff, J. M. Lavest, and P. Martinet. Exploiting rolling shutter distortions for simultaneous object pose and velocity computation using a single view. In *Proc. European Conference on Computer Vision*, Graz, Austria, May 2006.

[2] O. Company, F. Pierrot, and H.B. Choi. Design and control of a novel 4-dofs parallel robot h4. *ICRA'03: International Conference on Robotics and Automation*, pages 1185–1190, 2003.

[3] D. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1/2):123–141, June 1995.

[4] J. Gangloff and M. de Mathelin. High-speed visual servoing of a 6 DOF manipulator using multivariable predictive control. *Advanced Robotics. Special issue: advanced 3D vision and its application to robotics*, 17(10):993–1021, December 2003.

[5] M. Gautier and W. Khalil. Exciting trajectories for the identification of base inertial parameters of robots. *Int. J. Robotics Research*, 11(4):362–375, 1992.

[6] M. Gautier and Ph. Poignet. Extended kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Practice*, 9(12):1361–1372, 2001.

[7] G. Gogu. Fully-isotropic t3r1-type parallel manipulators. *On Advances In Robot Kinematics*, pages 265–272, 2004.

[8] S. Guegan. *Contribution à la modélisation et l'identification dynamique des robots parallèles*. Thèse, Ecole Centrale de Nantes et Université de Nantes, 2003.

[9] M. Husty. An algorithm for solving the direct kinematics of general Gough-Stewart platforms. *Mech. Mach. Theory*, 31(4):365–380, 1996.

[10] W. Khalil. General solution for the dynamic modeling of parallel robots. *ICRA 04 : IEEE International Conference on Robotics and Automation*, pages 3665–3670, 2004.

[11] W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*. Taylor and Francis, 2002.

[12] JM. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration. In *Proc. European Conference on Computer Vision*, pages 158–174, Freiburg, Germany, June 1998.

[13] D. G. Lowe. Fitting parameterized three-dimensional models to image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[14] J. T. Lapreste M. Dhome, M. Richetin and G. Rives. Determination of the attitude of 3-d objects from a single perspective view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, December 1989.

[15] F. Marquet. *Contribution à l'étude de l'apport de la redondance en robotique parallèle*. Thèse, Université Montpellier II, 2002.

[16] M. Meingast, C. Geyer, and S. Sastry. Geometric models of rolling-shutter cameras. In *Proc. of the 6th Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, Beijing, China, October 2005.

[17] J-P. Merlet. An algorithm for the forward kinematics of general 6 d.o.f. parallel manipulators. Technical Report 1331, INRIA, November 1990.

[18] M.M. Olsen and H.G. Petersen. A new method for estimating parameters of a dynamic robot model. *Trans. on Robotics and Automation*, 17(1):95–100, 2001.

[19] P. Renaud, N. Andreff, J.-M. Lavest, and M. Dhome. Simplifying the kinematic calibration of parallel mechanisms using vision-based metrology. *IEEE Transactions on Robotics*, 2005.

[20] P. Renaud, A. Vivas, N. Andreff, P. Poignet, P. Martinet, F. Pierrot, and O. Company. Kinematic and dynamic identification of parallel mechanisms (in press). *Control Engineering Practice*, 2005.

[21] J. Swevers, C. Ganseman, B.D. Tükel, J. De Schutter, and H. Van Brussel. Optimal robot excitation and identification. *Trans. on Robotics and Automation (In press)*, 13(5):730–740, 1997.

[22] R. Horaud T. Q. Phong and P. D. Tao. Object pose from 2-d to 3-d point and line correspondences. *International Journal of Computer Vision*, pages 225–243, 1995.

[23] A. J. P. Theuwissen. *Solid-state imaging with chargecoupled devices*. Kluwer Academic Publishers, 1995.

[24] Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, Miami Beach, 1986.

[25] P. Wenger and D. Chablat. Kinematic analysis of a new parallel machine tool : the Orthoglide. *Advances in Robot Kinematic*, pages 305–314, 2000.

[26] B. Wilburn and al. High-speed videography using a dense camera array. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington DC, USA, June - July 2004.