**Proceedings of the 2001 IEEE/RSJ**
**International Conference on Intelligent Robots and Systems**
**Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001**

# Analysis on Time-Delay of Commercial Off-The-Shelf Vision System considering Motion-Blur

Sung Ho Kim[1] and Byung Kook Kim[2]
[1]*Micro-Robot Design Education Center, KAIST, Taejon, Korea*
[2]*Department of EECS, KAIST, Taejon, Korea*
*Email: shkim@mrdec.kaist.ac.kr, bkkim@ee.kaist.ac.kr*

## Abstract

*In this paper, we analyze inherent time-delay of Commercial Off-The-Shelf (COTS) vision system in measuring postures of moving objects and reveal minimum sensing time-delay obtainable. Slow shutter speed in COTS vision systems causes motion-blur in moving objects, which invites vagueness in time-delay as well as in measured position. We can overcome the vagueness by introducing exposure time-delay. Considering the exposure time-delay and image transfer time of overlay boards, we can figure out minimum sensing time-delay (MSTD) in COTS vision systems. Also, utilizing field-interleaved scanning and slow readout speed of COTS cameras, we propose a posture determination algorithm that guarantees MSTD, which uses both field-based processing and processing during overlaying methods. To verify validity of the proposed MSTD and the algorithm, overall time-delay of our COTS vision system is measured directly via pendulum experiments.*

## 1. Introduction

Nowadays, vision systems become affordable due to rapid improvements on commercial off-the-shelf (COTS) components such as video cameras, overlay (capture) boards, and personal computers. Recently, most robot vision systems and vision-based control systems take advantage of COTS vision systems instead of specialized vision systems due to their good performance in low cost. However drawbacks such as large and vague time-delay of COTS vision systems restrict their application to visual control.

In general, larger time-delay makes slimmer phase margin in visual control system, which in turn invites unstable behavior of visual control system. Hence it is important to minimize it as much as possible in dynamic vision applications. There were several researches on time-delay for vision systems. Most of them analyzed inherent time-delays of their own vision systems and compensated the delay empirically using various methods such as PID, Smith predictor, and etc. Calculating overall time-delay by using given specifications of their vision systems without direct measurement and focusing on compensation of time-delay, they did not consider effects of physical characteristics of COTS vision systems such as motion-blur. Hence their compensators were not exact and the performances were poor in real experiments.

Although motion-blur in images always occurs due to slow shutter speed in COTS cameras, human is ignorant of the blur. However, in point of machine vision, the motion-blur causes significant vagueness in time as well as measured postures. There were a lot of researches about motion-blur, which mainly described on recovering original images [4], extracting motion information from blurred images [7], obtaining images without motion-blur [5], and adjusting exposure time to reduce motion-blur [3]. However, there are few researches considering precisely both time and position together under effects of motion-blur, which have equal importance in dynamic visual sensing and control such as visual servoing.

In this paper, we analyze inherent time-delay (ITD) in measuring postures considering motion-blur and image grabbing time. Based on this analysis, we propose minimum sensing time-delay (MSTD) posture determination algorithm to keep the overall time-delay minimum obtainable for COTS vision systems. Taking advantage of field-interleaved scanning and readout speed of COTS vision system, we use both field-based processing and process-during-overlay methods in the algorithm.

## 2. Analysis on Time-Delay in Measuring Posture

Color CCD modules used in commercial video cameras feature field integration with electronic shuttering [2]. Electronic shuttering, which transfers collected electrons in photosites to vertical CCD columns, limits exposure time below one field time $T$ (1/60 sec for NTSC or 1/50 sec for PAL). In COTS vision systems, slow shutter speed causes motion-blurs of moving objects. Although using strong strobe lighting can reduce motion-blur, it is hard to use strobes in natural environments. In general, motion-blurred images are vague in time to be taken as well as their measured postures. However accurate time is also important as well as accurate posture in the aspect of visual control. In this section, we will show that the temporal and positional vagueness due to motion-blur at the same time can be eliminated by introducing the exposure time-delay for a specific shape.

Motion-blurred images usually show intermediate colors

0-7803-6612-3/01/$10.00 © 2001 IEEE

between object and background.

***Definition 1*** An *blend color* $C$ of color vectors $C_o$ and $C_f$ in color space $\mathbb{C} \subset \mathbb{R}^3$ is defined as a color vector that satisfies

$$C = C_o + n(C_f - C_o) \qquad (1)$$

where $n \in [0,1]$ is its normalized intensity.

In general, intermediate colors between object and background colors of a blurred image change gradually from object color to background color and can be expressed as a blend color.

***Definition 2*** A *duo-tone image* $\mathcal{D}$ *of colors* $C_o$ *and* $C_f$ is defined as a color image of which all colors are blend colors of colors $C_o$ and $C_f$ such as

$$\mathcal{D} = \left\{ C(x,y) \middle| C(x,y) = C_o + n(x,y)\left(C_f - C_o\right), \forall x,y \in \mathbb{R} \right\} \qquad (2)$$

where $n(x,y)$ is normalized intensity at $(x,y)$.

Set of normalized intensity $n(x,y)$ of Definition 2 can be thought as a gray image of $\mathcal{D}$.

***Definition 3*** A *normalized gray image* $\mathcal{G}$ of duo-tone image $\mathcal{D}$ is defined such that

$$\mathcal{G} = \left\{ n(x,y) \middle| C(x,y) = C_o + n(x,y)\left(C_f - C_o\right), \forall x,y \in \mathbb{R} \right\} \qquad (3)$$

where $C(x,y) \in \mathcal{D}$.

By Definitions 1, 2, and 3, we can represent a duo-tone image $\mathcal{D}$ uniquely as a triplet $(\mathcal{G}, C_o, C_f)$. Also, thresholding a duo-tone image $\mathcal{D}$ by an intermediate color $C_{th}$, we can use its scalar $n_{th}$ instead of $C_{th}$.

***Property 1*** For any duo-tone image $\mathcal{D}$, there exists a normalized scalar threshold $n_{th}$ that gives the same as threshold binary image by a blend color $C_{th}$.

***Proof:*** Since $C_{th}$ is a blend color satisfying monotonic Eq. (1), we can determine $n_{th}$ corresponding to any blend color $C_{th}$ in $\mathcal{D}$ uniquely as follows

$$n_{th} = \frac{c^i_{th} - c^i_o}{c^i_f - c^i_o} \qquad (4)$$

where $c^i_{th}$, $c^i_o$, and $c^i_f$ are *i*-th component of color vectors $C_{th}$, $C_o$, and $C_f$ respectively. Since unique $n_{th}$ can be determined monotonically according to $C_{th}$, the same binary image is given by thresholds $C_{th}$ and $n_{th}$. ■

A motion-blurred image can be considered as a kind of duo-tone image if monotone object and background are used. In this case, a triplet ($\mathcal{G}$, background color, object color) represents the motion-blurred image. Also pixel values in $\mathcal{G}$ are proportional to the number of accumulated photoelectrons during exposure time $t_e$ as follows [**2**]

$$\mathcal{G} = \left\{ n(x,y) \middle| n(x,y) = K\int_0^{t_e} b(x,y,t)dt,\ \forall x,y \in \mathbb{R} \right\} \qquad (5)$$

where $K$ is a normalization constant and $b(x,y,t)$ is a accumulation function, which returns 1 if object includes point $(x,y)$ at time $t$; otherwise 0. By Property 1, a binary image $\mathcal{B}$ of $\mathcal{D}$ can be given by a threshold function $\mathfrak{B}(\cdot)$ with $\mathcal{G}$ and $n_{th}$ as

$$\mathcal{B} = \mathfrak{B}(\mathcal{G}, n_{th}) \qquad (6)$$

Also a centroid $(x_e, y_e)$ of $\mathcal{B}$ can be given by the centroid function $\mathfrak{C}(\cdot)$ as

$$(x_e, y_e) = \mathfrak{C}\left(\mathfrak{B}(\mathcal{G}, n_{th})\right) \qquad (7)$$

Note that centroid of motion-blurred image is a function of threshold $n_{th}$ for a given image $\mathcal{G}$. In the remainder of this chapter, we use term '*object*' instead of '*object image on image plane*' for simplicity. Linear motion of an object with constant velocity $\vec{v}$ is given as follows

$$\vec{v} = \frac{l}{t_e}\vec{d} \qquad (8)$$

where $l$ is the traveled length of a moving object during exposure time $t_e$ and $\vec{d}$ is a normalized direction vector of motion. Trajectory of the centroid is given as follows

$$(x(t), y(t)) = \vec{v}t + (x_0, y_0) \qquad (9)$$

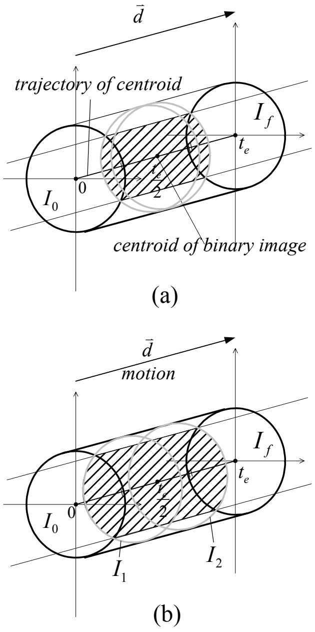
where $(x_0, y_0)$ is an initial centroid at $t = 0$. Also we know all points on the object image move along the line that penetrates the points with direction $\vec{d}$. For bisymmetric shapes, which are symmetric with respect to two perpendicular lines, centroid of the motion-blurred object can be given easily.

***Property 2*** If a bisymmetric object has constant linear motion with velocity of Eq. (8), its centroid $(x_e, y_e)$ becomes a constant that is independent on $n_{th}$ as follows

$$(x_e, y_e) = (x(\frac{t_e}{2}), y(\frac{t_e}{2})) \qquad (10)$$

where $(x(t), y(t))$ is centroid of moving object at $t$ and $t_e$ is the exposure time.

Proof is skipped due to shortage of space [**6**]. In most real movements, rotation is included as well as translation. Since rotation with respect to an arbitrary points on an object can be decomposed into translation and rotation w.r.t. centroid. Hence, we can consider only translation and rotation w.r.t. centroid without loss of generality. In case of movement including rotation, Property 2 does not hold generally. However, circular shape satisfies Property 2 in spite of rotation [Figure 1] since circular shape is invariant to rotation w.r.t. centroid. In the figure, $I_0$ and $I_f$ are images of the circular shape at start time and end time of an exposure respectively. Also slashed regions of (a) and (b) are binary images with two different thresholds respectively.



(a)

(b)

**Figure 1. Motion-blurred image of circular shape**

In the figure, we can confirm Property 2, which tells that centroids of binary images are all the same in spite of different thresholds. By Property 2, motion-blurred effect in measuring postures of circular objects can be compensated by the exposure time-delay $t_m$ from the end time of exposure as follows.

$$t_m = \frac{t_e}{2} \tag{11}$$

Assumption Eq (8). of linear motion reflects real movement very well when object movement is much slower than the shutter speed as follows

$$\tau >> t_e \tag{12}$$

where $\tau$ is effective time-constant of robot system in image coordinates. In aspect of visual servoing, assumption of Eq. (12) is reasonable since movement of robot must be observable in update rate of vision system.

As shown in Figure 2, we can observe that electronic shuttering ends exposure. Also, we can observe that it takes one field time $T$ to complete transfer of each field. Hence inherent time-delay of COTS vision systems in measuring postures becomes

$$\text{ITD} = T + t_m \tag{13}$$

which is the sum of the exposure time-delay and the field transfer time. When $t_e = T/2$ (usual case),

$$\text{ITD} = 1.5\,T \tag{14}$$

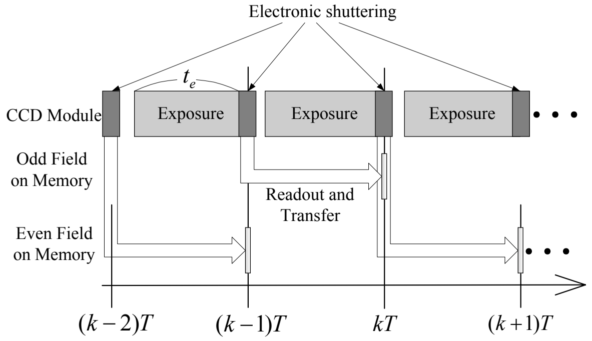


**Figure 2. Timing of exposure and field transfer**

As a result, we recommend circular marks again to overcome vagueness in time and position due to motion-blur although circular marks are already used widely in visual control applications because of its simple and rotation-invariant shape. We call overall time-delay of COTS vision system as *sensing time-delay*, which includes ITD and processing time of image processing and control algorithm. Hence, the ITD becomes MSTD if the processing time approaches zero as follows

$$\text{MSTD} = \text{ITD} = T + \frac{t_e}{2} \tag{15}$$

## 3. MSTD Algorithm for Posture Generation

Most conventional vision systems store an image on frame memory and then process it sequentially. Hence separate clocks are needed to store and process images respectively, which causes temporal redundancy. Also, when objects or cameras are moving, COTS vision systems suffer "tearing" phenomenon as shown in, which is due to separate exposures of both fields in a frame [**2**].The MSTD posture determination algorithm eliminates the temporal redundancy in storing and processing and the tearing phenomenon.

Color CCD modules used in commercial video cameras feature "field integration" using complementary CFA (Color Filter Array). In field integration mode, even though all photosites are read out during each video field, mixed colors of adjacent photosites through a complementary CFA are decoded into pixels of its own field, which make alternatively interleaved fields [2]. Hence, COTS vision systems have separate exposures for both fields respectively. Tearing phenomenon [Figure 3] can be avoided by processing only currently updated field and calculates based on frame coordinates. We propose field-based processing method against tearing phenomenon as follows.

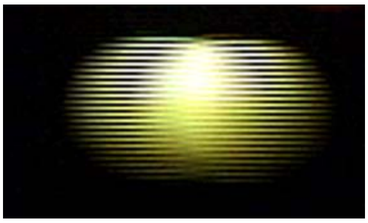


**Figure 3. Captured frame image of moving ball**

1. Get a field image from frame buffer at field rate (60 Hz or 50 Hz).
2. Perform threshold and Connected Component Algorithm(CCA) with the field image to get the binary image.
3. Calculate the posture of binary image in frame coordinates.

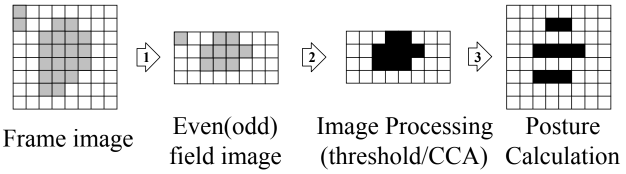


**Figure 4. Posture generation based on field image**

Step numbers of above list corresponds to numbers of arrows in Figure 4 respectively. All steps are iterated sequentially at field rate. On the other hand, field-based processing may cause *flickering* phenomenon, where measured postures have different means for odd/even fields respectively. In general, large and round shape alleviates flickering. We propose process-during-overlay method to eliminate temporal redundancy in storing and processing. It minimizes sensing time-delay to ITD. Usually, transfer time to frame memory is almost negligible in comparison with readout time of CCD modules, since transfer speed of them is much faster than readout speed (14.318 MHz or 17.7 MHz [1]). Hence it is possible to process stored part of currently updating field by a personal computer. The process-during-overlay method needs some sophisticated calculations about access time to objects.

Figure 5 depicts only a currently scanning field. First, we sort objects according to top-left-to-bottom-right order. Then we calculate complete time $t_k$ to store $k$-th window in frame memory as follows.

$$t_k = \delta b_k \qquad (16)$$

where $\delta$ is a constant of scan time per pixel and $b_k$ is the number of bytes from start of field to end of $k$-th window. $\delta$ can be determined by scanning standard and image size. In case of NTSC and 640×480 size, $\delta$ is given as

$$\delta = 0.09143 \text{ usec/pixel} \qquad (17)$$

After processing of current window is completed, complete time $t_{k+1}$ of next window is compared with current time and determine whether process or not. To be conservative, we check again completion of the next window by using dirty bytes (end bytes of windows). After processing all windows, we wait for rewriting end byte of current field as field synchronization.

Following Algorithm 1 gives MSTD posture determination algorithm, which uses field-based processing and process-during overlay methods.

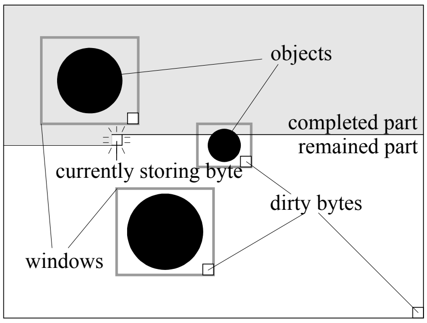


**Figure 5. Overlaying field image**

**Algorithm 1. MSTD Posture Determination Algorithm**

```
/* 1. Initial search */
1.1 Determine N initial target locations using full-frame search
/* Sort targets according to scan order*/
1.2 Sort N objects according to left-top-to-right-bottom order
/* Prediction for next update */
1.3 For target I = 1:N
    1.3.1. Determine window size to fit 2-byte pixel
      alignment (YUYV format image)
    1.3.2. For each scan path of odd/even field
        1.3.2.1. Determine locations of dirty bytes
            1.3.2.2. Determine max. scan time from start of
              field to the dirty bytes.
1.4. Wait for a next start of field. (field sync.)
/* 2. Update target location at each field*/
2.1 For target I = 1: N
    2.1.1. Wait for rewriting the dirty byte of current
      window until maximum scan time
    2.1.2. Color detection and CCA
    2.1.3. Calculate postures.
/* Sort targets according to scan order*/
2.2. Sort N targets according to left-top-to-right-bottom order
```

```
/* Prediction for next update */
2.3. For target = 1:N
    2.3.1. Determine window size to fit 2-byte pixel alignment
    2.3.2. For each scan path of odd/even field
        2.3.2.1. Determine locations of dirty bytes
        2.3.2.2. Determine maximum scan time from the start
          of field to the dirty bytes.
/* Synchronization */
2.4. Wait for rewriting the dirty byte of each field (field sync.)
3. Repeat Step 2
```

## 4. Experimental Results

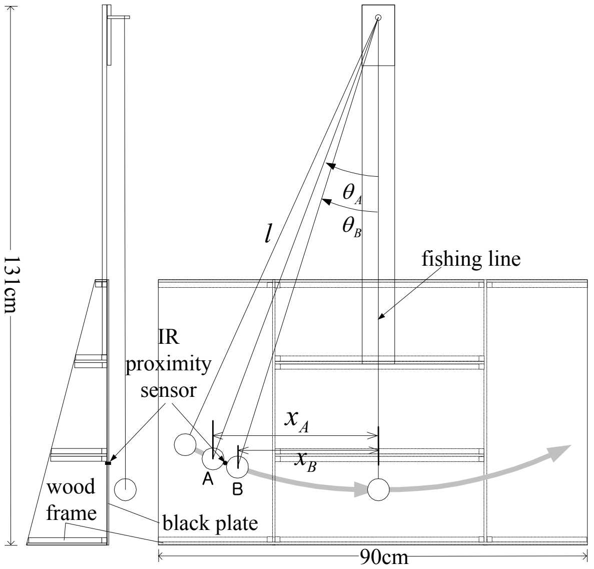


**Figure 6. Experimental pendulum system**

In this section, we verify the result of Section 2 and MSTD posture generation algorithm proposed in Section 3 by measuring directly overall time-delay of our COTS vision system. Since motion of pendulum can be described completely by several parameters, we can find the accurate position of pendulum at an arbitrary time if an initial state is given. Comparing its model with measured motion in vision systems, we can measure the time-delay in off-the-shelf vision system. Figure 6 shows an pendulum system used in our vision system. An official tennis ball is used as a pendulum. Points A and B shown in Figure 6 are sensing points where IR sensor detects the pendulum. The sensor can detect the ball up to 10 cm. The black plate is adjusted to be included in field of view and be perpendicular to optical axis of the camera. Our COTS vision system consists of a PC using Celeron 433 CPU and Windows 98, a 8mm camcorder SV-H66, and an overlay board of local brand using Bt848 video-capture processor. In the experiments, position of pendulum is defined by its central angle $\theta$. Its origin is 6 o'clock direction and sign convention follows the right-hand law. State representation of pendulum motion is arranged as follows

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 - kx_2 \end{bmatrix} \tag{18}$$

where $x_1 = \theta$ and $x_2 = \dot{\theta}$. And $l$ is the length of string and $g$ is gravity acceleration, and $k$ is a coefficient of fluid resistance of air. Let origin of time set to arrival time to point A. Then we can get its numerical solution using Runge-Kutta method with a following initial state as

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} \theta_A \\ \frac{\theta_B - \theta_A}{\tau} \end{bmatrix} \tag{19}$$

where $\tau$ is first time interval in which the pendulum travels point A to point B. All parameters used in the experiment equipment except coefficient of fluid resistance $k$ are measured and arranged in Table 1.

**Table 1. Parameters of experimental equipment**

| | Description | Value |
|---|---|---|
| $g$ | Gravity | $980\ cm/\sec^2$ |
| $l$ | Length of pendulum | $116.7\ cm$ |
| $x_A$ | Position of left sensing limit | $-36.2\ cm$ |
| $x_B$ | Position of right sensing limit | $-31\ cm$ |
| $\hat{k}$ | Estimated coefficient of fluid resistance of air | $0.028\ \sec^{-1}$ |

Since it is hard to measure the coefficient $k$ of tennis ball directly, an estimate of $k$ can be determined in order that degree of decay in model Eq. (18) should agree with measured trajectories. As a result, we can find an estimate $\hat{k}$ .

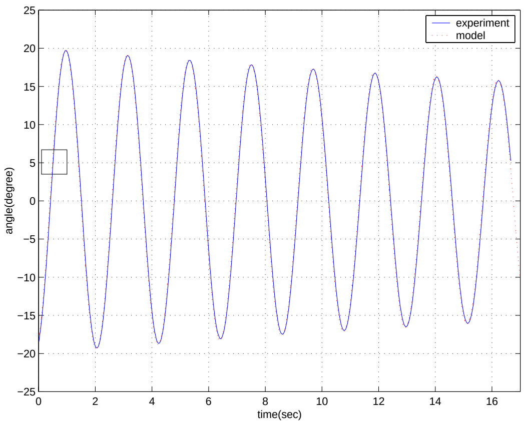


**Figure 7. Experimental trajectories of pendulum**

In Figure 7, we overlay a numerical solution of Eq. (18) using measured parameters on an experimental trajectory. In the figure, we can confirm that the model Eq. (18) reflects real movement of pendulum very well. However, a close look shows difference between two curves: There exists a time-delay between them, which is observed easily in magnification of a box in Figure 7 [Figure 8]. Using interpolation of numerical solution of Eq. (18), we can find the real passing time at a measured position. Then we can get overall time-delay as difference of the measured time and given real time at the position.

Since it is erroneous to measure time-delay near turning points due to sluggish movement of pendulum, time-delays within ±10 degree of first two periods are taken in Figure 9. Upper data in the figure is for full exposure (1/60 sec) and bottom is for exposure time of 1/100 sec ("flickerless" option). As shown in the figure, measured time-delays overlap expected ITDs .

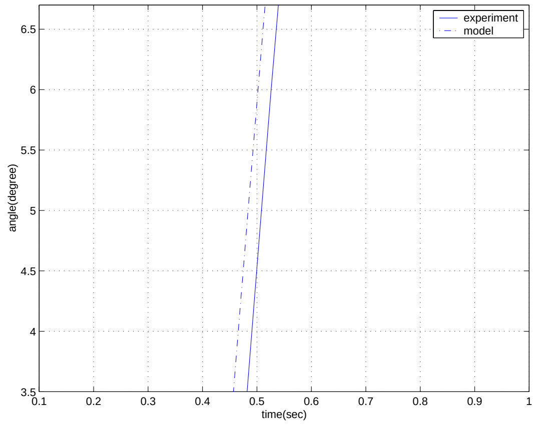


**Figure 8. Magnified trajectories of Figure 8**

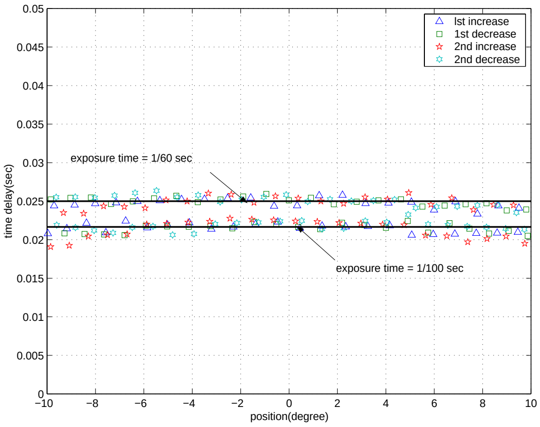


**Figure 9. Measured time-delay**

Means of measured time-delays are 25.2msec (expected as 25msec) and 21.9msec (expected as 21.7msec). The results confirm both as follows: Measured overall time-delays agree with Eq. (15) and the proposed MSTD algorithm works as expected. Also, since field transfer time of COTS vision system is fixed according to its scanning standard (NTSC: 1/60 sec per a field), we can verify Eq. (11) of exposure time-delay from Figure 9.

## 5. Conclusion

In the paper, taking advantage of characteristics of COTS vision system such as slow shutter speed, field interleaving scan, and slow readout speed, we analyzed MSTD including exposure time-delay and field transfer time, and also proposed the MSTD posture determination algorithm. It was shown that motion-blur due to its slow shutter speed effects in ITD and the effect of motion-blur can be taken care of by introducing "exposure time-delay", which equals to a half of exposure time. For MSTD posture determination, we proposed an algorithm featured by field-based processing and process-during-overlay methods. Using pendulum experiments, we verified validity of proposed MSTD and the algorithm.

## References

[1] Charles A. Poynton, *A Technical Introduction of Digital Video*, John Wiley & Sons, 1996

[2] Gerald C. Holst, *CCD Arrays, Cameras and Displays*, SPIE Optical Engineering Press, 1996

[3] Ken Owens and Larry Matthies, "*Passive Night Vision Sensor Comparison for Unmanned Ground Vehicle Stereo Vision Navigation*," Proceedings of IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications, pp. 59-68, 1999

[4] Sang Kyu Kang, Jung Hoon Jung, Joon Ki Paik, and Young Chan Kim, "*Analysis of multiple moving objects in video for removing motion-blur*," Proceedings of TENCON'93, vol. 2, pp.1267-1270, December 1993

[5] Shi Ding-ji, Wang shou-yan, Huang han-wen, and Wu bo-xin, "*Reduced Image Blur Due to Motion By Sight-line Following*," Proceedings of the 1994 IEEE International Conference on Industrial Technology, pp.659-664, 1994

[6] Sung Ho Kim, *A Study on Direct Visual Servoing with Field-Rate COTS Vision System*, Ph.D. Dissertation, KAIST, 2000

[7] Wei-Ge Chen, N. Nandhakumar, and W. N. Martin, "*Image Motion Estimation From Motion Smear – A New Computational Model*," IEEE Transactions on PAMI, Vol. 18, No.4, April 1996