ATRテクニカルレポート表紙

〔非公開〕

TR-C-0094

Calibration of a Stereoscopic Display System without special equipment needs, and delay reduction in the eye tracking system

ヘルベ　コルタダ　Herve CORTADA

大村　克之　Katsuyuki OMURA

鉄谷　信二　Nobuji TETSUTANI

1994. 1. 11

ATR通信システム研究所

Calibration of a Stereoscopic Display System without special equipment needs, and delay reduction in the eye tracking system.

Herve Cortada, Katsuyuki Omura and Nobuji Tetsutani

ATR Communication Systems Research Laboratories.

ABSTRACT

The following report describes some research that has been conducted in ATR to test and augment the performances of a 3D screen and head tracking system. Given a particular screen, we tried to evaluate and correct its possible bias and reduce the delay of the head tracking system, so that the lenticular can be put to a practical use without bringing discomfort to the viewer.

INTRODUCTION

ATR is conducting research on virtual space teleconferencing system that achieves a sense of reality. One solution for this is to use a lenticular screen with an eye position tracking technique that doesn't require the viewer to wear special glasses, for the sake of comfort and naturalness. The display so created should be able in the future to provide multi-viewers with a virtual environment that varies according to their actions without any impediment such as wearing 3D glasses or having a captor on their face. But for now....

The research dealt with in this report has taken place from August to December 1993. The lenticular screen involved is, by its size, only designed for one viewer, but still the technique is basically the same for a multi-viewers screen which is the ultimate goal of this system. The eye-tracking method is based on the idea that the pupil provides good light reflection and that we can detect their position using two infrared-cameras without disturbing the scenery viewed, as infrared light is outside the visible spectrum. Still there is much reflection noise that can affect the tracking of the eyes. So, until this problem is solved, we place a marker that has a very high reflection coefficient on the viewer's chin. There is no noise problem in the tracking of this captor. The stereo images are sent left and right of the tracked position of the marker, using a "standard" distance between human eyes. The images sent are Computer Generated images that vary according to the same tracking information to provide the sensation of change of position in the perceived virtual space.

All positions referred to in the following are taken on a same XYZ orthonormal coordinates system, referenced by the lenticular screen being studied. Z direction is the one orthogonal to the screen ("front-rear" direction), Y direction is the up and down direction, and X is the "left-right" direction of the screen. The 0 position on X axis is the center of the screen, the on Y and Z axis they differ according to the needs of the experiment. All distances are taken in millimeters.

PART I: screen calibration

The screen has to properly send the images to the eyes of the viewers. This type of lenticular works with one projector associated with each different viewer. It must follow the change of position of that person to send one image to each of his eyes. The interest with this system is that it extends the viewing area, as an important motion of the head is followed by a small one of the projector.

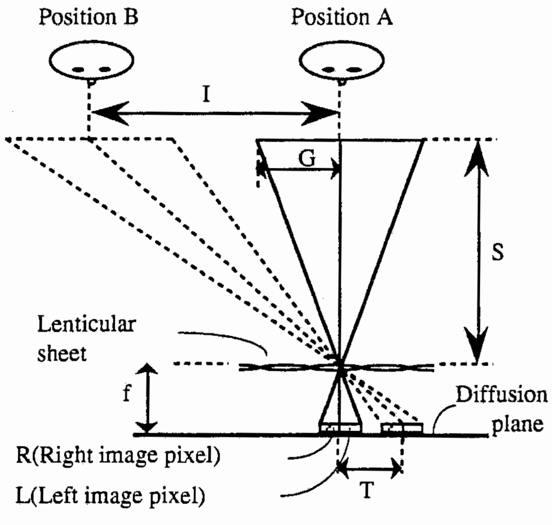


Fig. 1

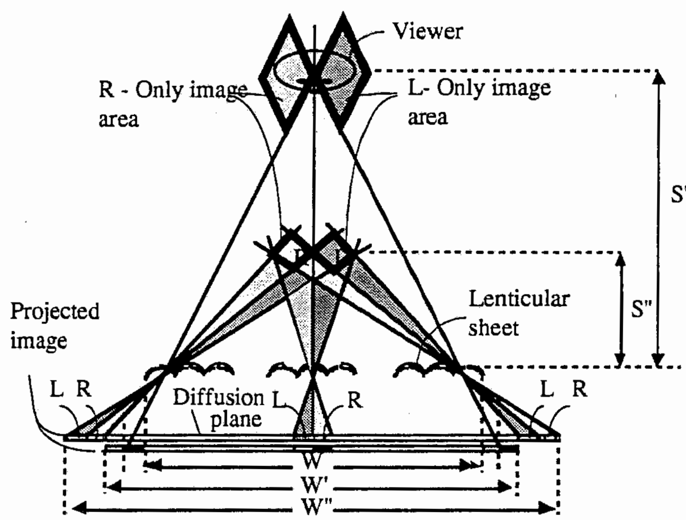


Fig. 2

A- Theoretical equations.

The moves of the projector are controlled by a program that sends a pulse number in absolute 2 dimensional coordinates according to where we want the images to be sent. Still, the knowledge we have of the screen is only theoretical. We know the type of equations it follows, and these are first used in the control program. But this is not precise enough to offer perfect 3D viewing. We have to adjust our equations to the characteristics of this particular lenticular.

The theoretical equations followed by this type of screen for X and Z directions are:

Lateral movement (X direction)

$$px = p0.X.f/(B.Z) + p0.m0$$

Where px is the pulse to be sent for lateral movement of the projector, f is the focal length of the screen and B is the magnifying power of the lens. p0 is a fixed value specific to this screen, and p0.m0 is a compensation term.

Front/rear movement (Z direction)

$$pz = p0.T.f/(2.Z) + p0.m1$$

Where pz is the pulse to be sent to the projector for z movement, f is the focal length of the screen, T is the standard projection distance, p0 is the same as above and p0.m1 is a compensation term.

Given a X and a Z this screen displays the same image for any position on the Y axis. That's why a motion of the viewer in this direction hasn't to be followed by a change of position of the projector.

To take in account the bias of the screen, we made a partition of the area in which the projector is moving in the X and Z directions and noted where were found the related best viewing positions (front position and first side position). We are mostly interested in the center best viewing position, as it is the one that we have that will be used in the tracking. It's the one we have the most precise knowledge of (by the equations) but it is not unique. And we will also need to know the behavior of the side positions, mainly because the viewer will have to look for the center best position at the beginning of the tracking and we want him to look for it in an area where he has no chance of being mistaken by a best side viewing position (it doesn't evolve with the same laws as the central one). Another reason is that ultimately this type of screen is designed for multiple viewers. And in this case one viewer could find a side position of one other camera crossing is eyes. Noting the way the first side viewing position handles itself could provide information to avoid that kind of trouble.

The next page shows the evolution of the evolution of the best central position, each noted with the corresponding position of the projector behind the screen. The following one is the same for the first best side position. An R before the point indicates that it is the right side position, as an L means left position. (Due to the position of the screen, the right point could not always be measured, and that is also true for left points. That's why in some cases we noted one, in other cases the other.) Once we had this results we pointed out to significant lines in Z direction with X fixed and others in X directions with Z fixed. And with that data we wanted to find what changes we should bring to the theoretical equations to control the projector more precisely. We looked for equations of the following type:

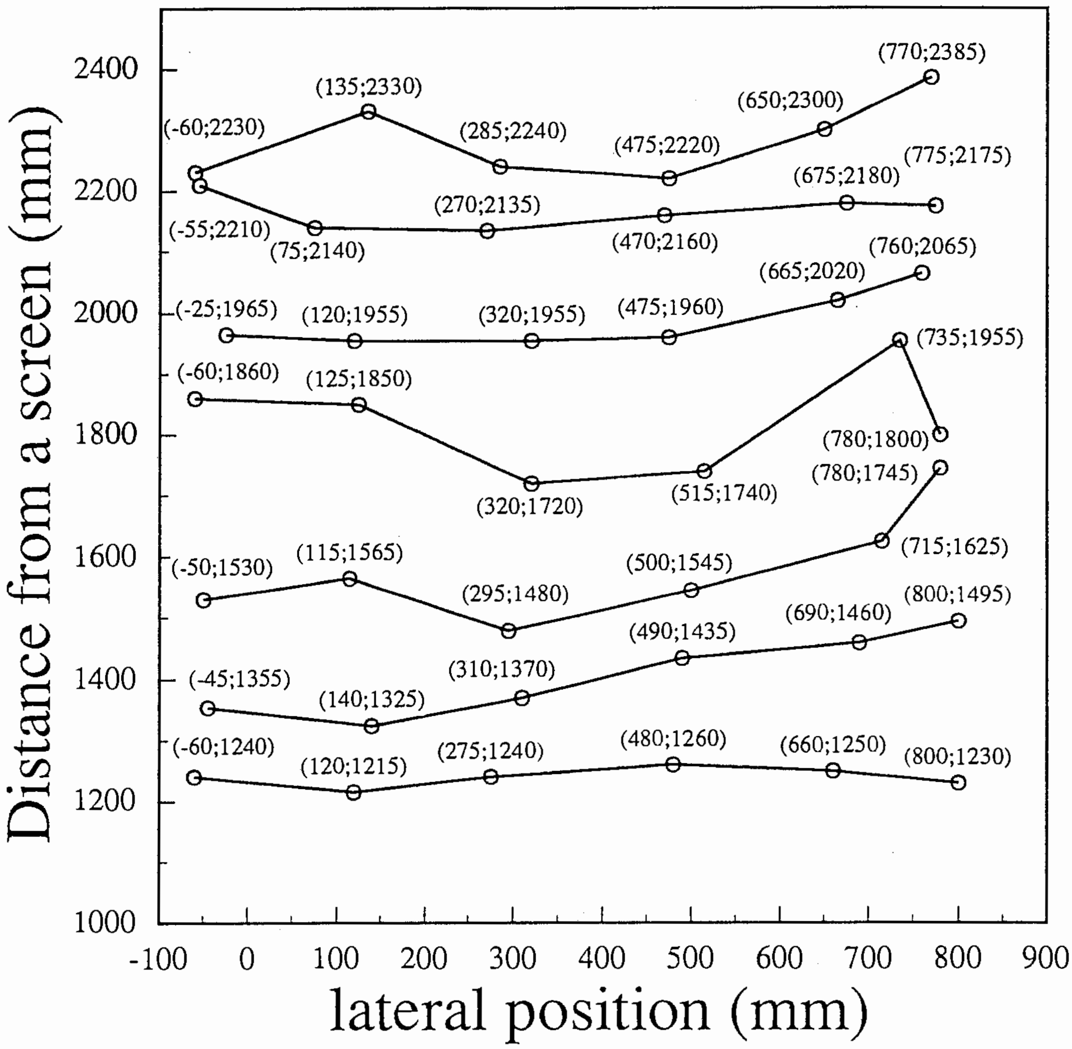
If px and pz are the pulse defined by the theoretical equations of the screen, we want the new pulse to be

$$p'x = Apx + B$$
$$p'z = Cpz + D$$

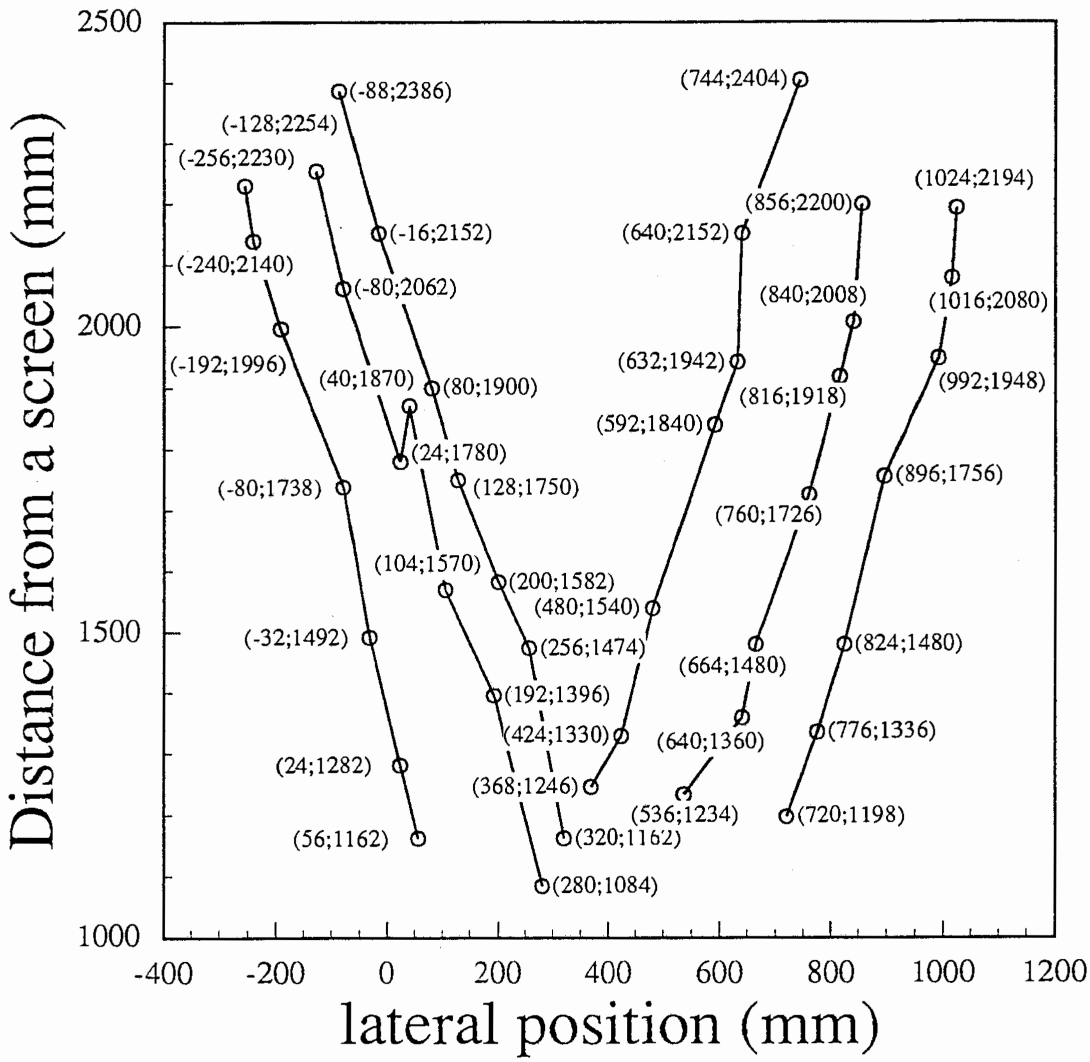
The important coefficient to adjust are A and C. The coefficient B and D are only used to see how close p'x and p'z are to the experimental results. They won't be implemented in the control program, as the viewer will have to determine his best viewing position at the beginning of the tracking sequence, which corresponds to finding his own B and D coefficients.

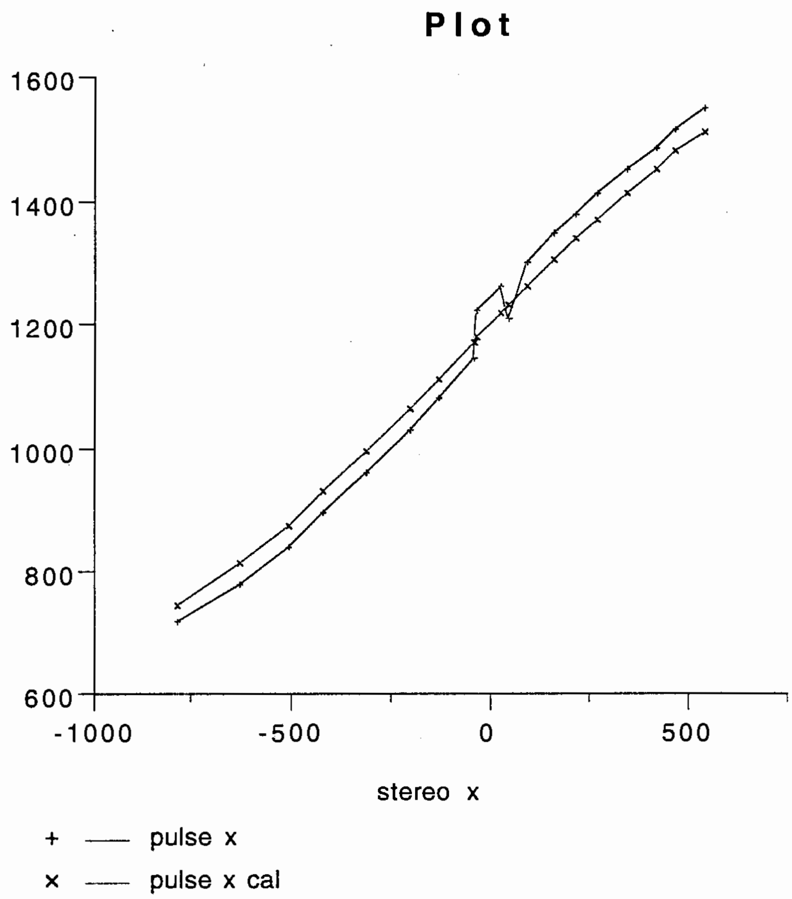
The curves we obtained show that the experimental results are well approximated by a first order function and that if the A coefficient can be set to 1(no modification needed for X direction), there is a serious bias for Z direction, as C as to be set to 0.62.

Evaluation of the viewable position

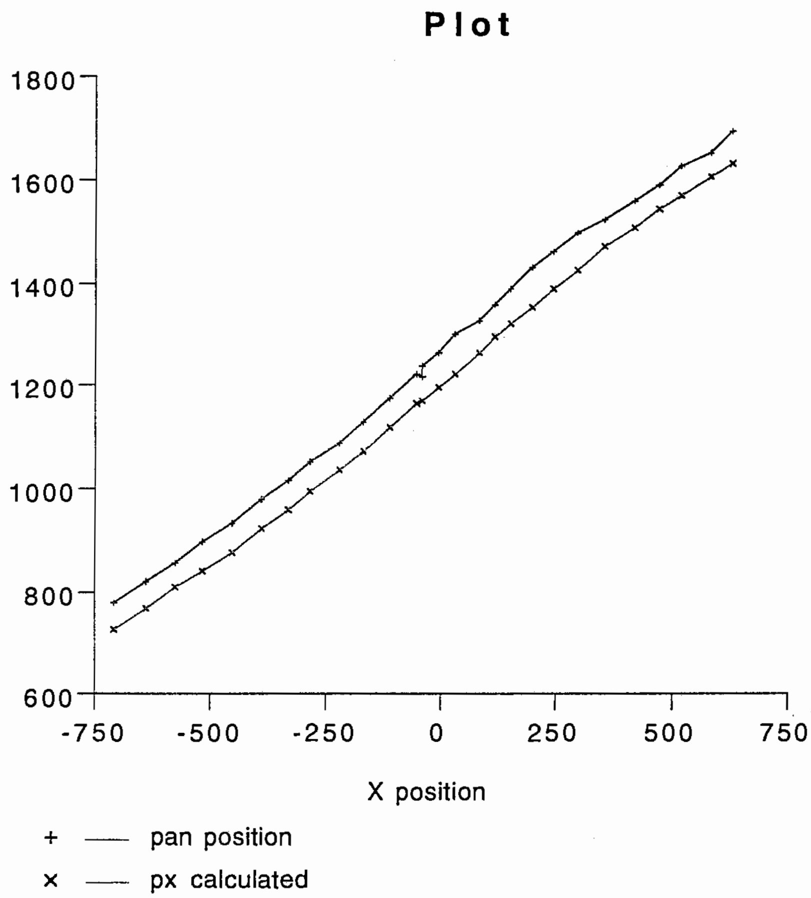


Evaluation of the viewable position

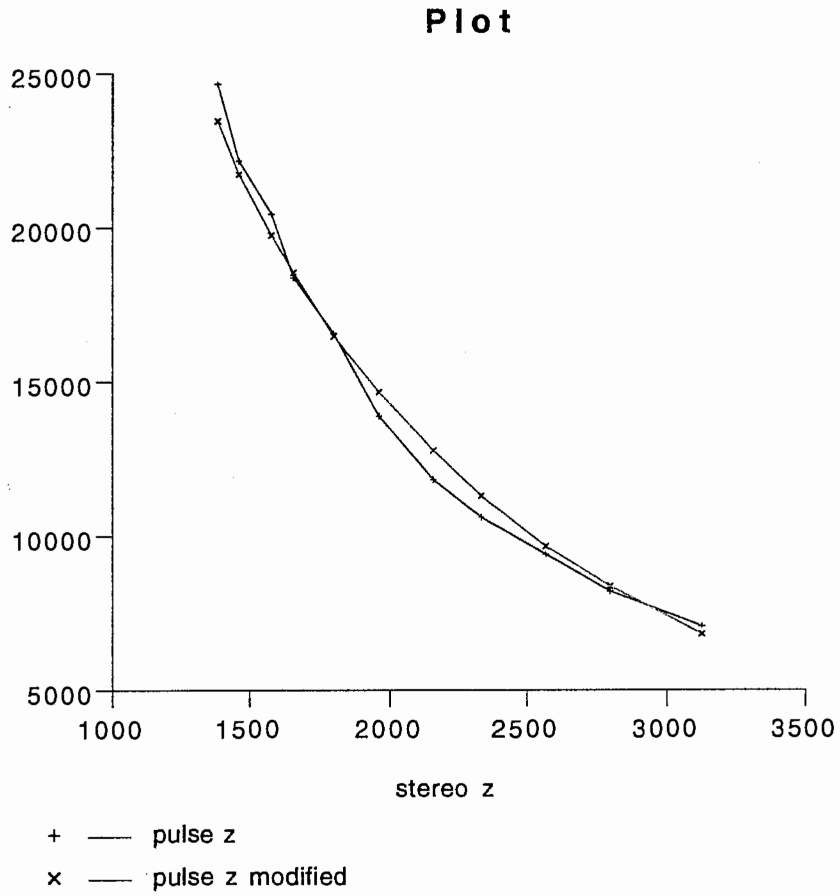




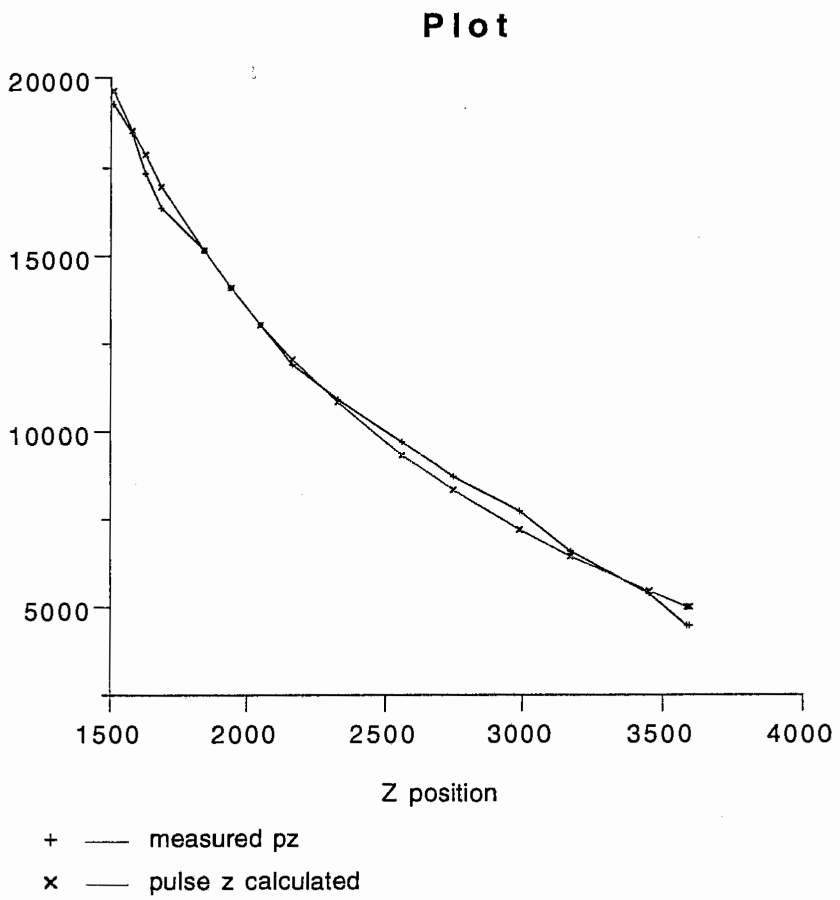
$$\frac{\frac{10000}{13}\cdot(47.1\cdot stereo\ x)}{(18\cdot stereo\ z)}+1200$$



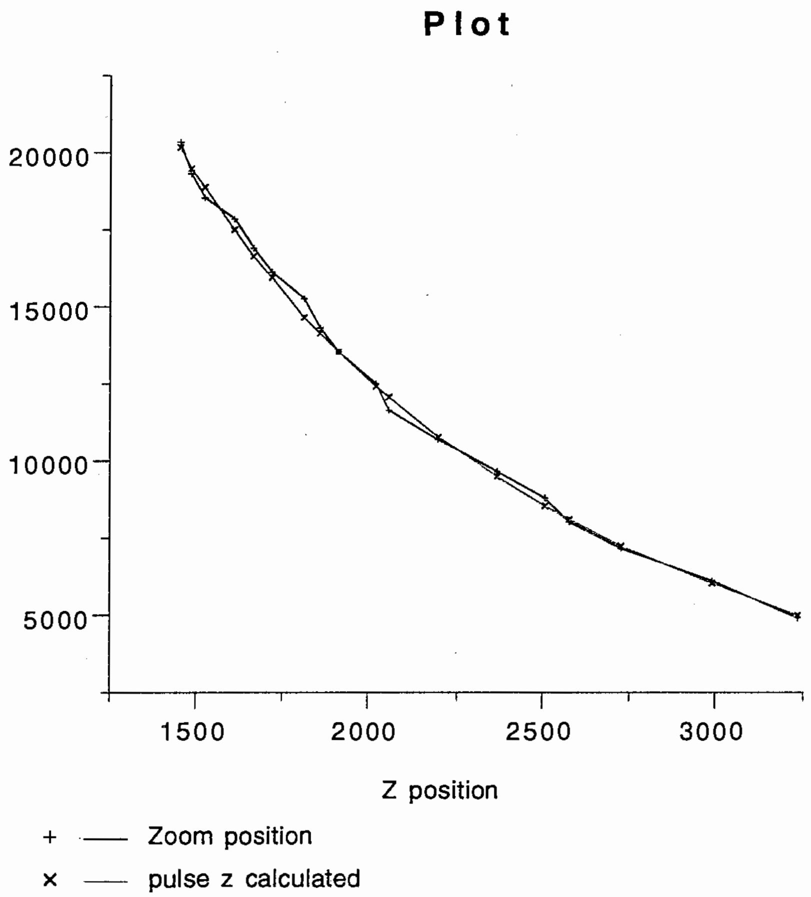
$$\left(\frac{\frac{10000}{13} \bullet (47.1 \bullet X\ position)}{(18 \bullet Z\ position)} + 1200 \right)$$



$$\left(\frac{10000\bullet 3500\bullet 47.1}{(2\bullet stereo\ z\bullet 13)}-26225\right)\bullet 0.65+10700$$



$$\left(\frac{10000\bullet 3500\bullet 47.1}{(2\bullet Z\ position\bullet 13)}-26225\right)\bullet 0.6+10200$$



$$\left(\frac{10000 \cdot 3500 \cdot 47.1}{(2 \cdot Z\ position \cdot 13)} - 26225\right) \cdot 0.63 + 9200$$

B- The cameras.

To determine the position of the viewer's face we use two infrared cameras that send their position information to the control program. And these cameras put in their own bias that is cumulative with the one of the screen. So we have to check the tracking system to approximate the default and rectify it in the control program.

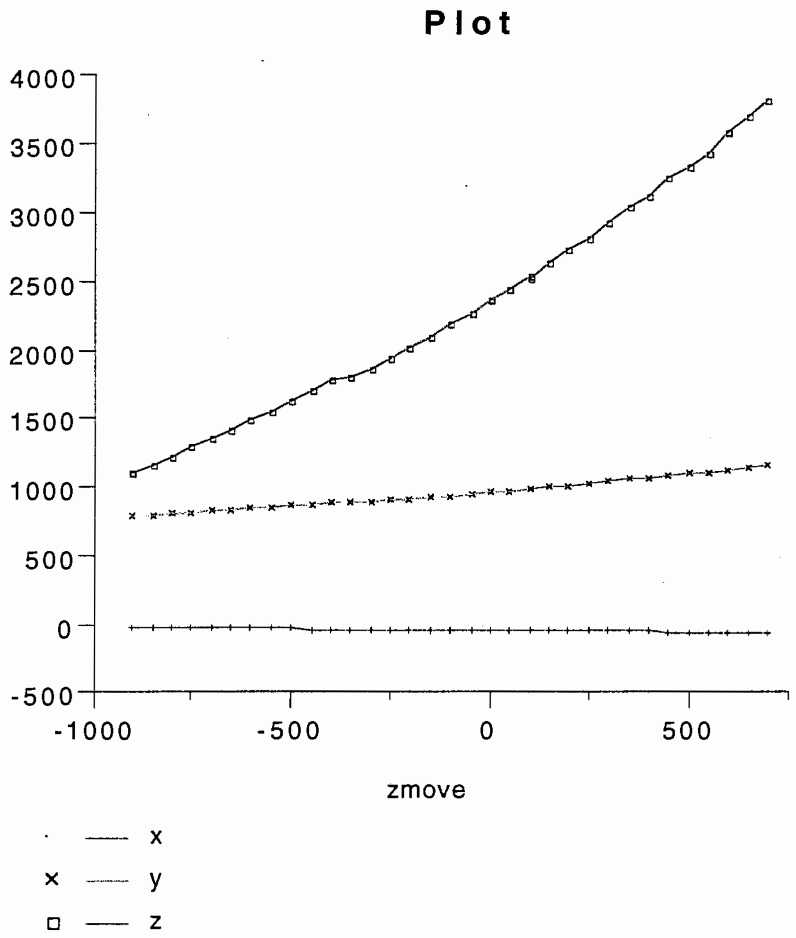
As the cameras measure Y position and send it back to the control program, we have to see that this information too is correct. The information on Y position is not needed for properly sending the images to the eyes, but they are important to determine what scene will be sent: the CG images received by the eyes change according to the viewer's position, modifying the perceived perspective according to all position coordinates. Error on Y coordinate could diminish quality of the motion feeling in the virtual environment. So for each of the 3 directions, we have measured the position of the reflective surface, and recorded what the cameras were sending to the control system as its tracked position.

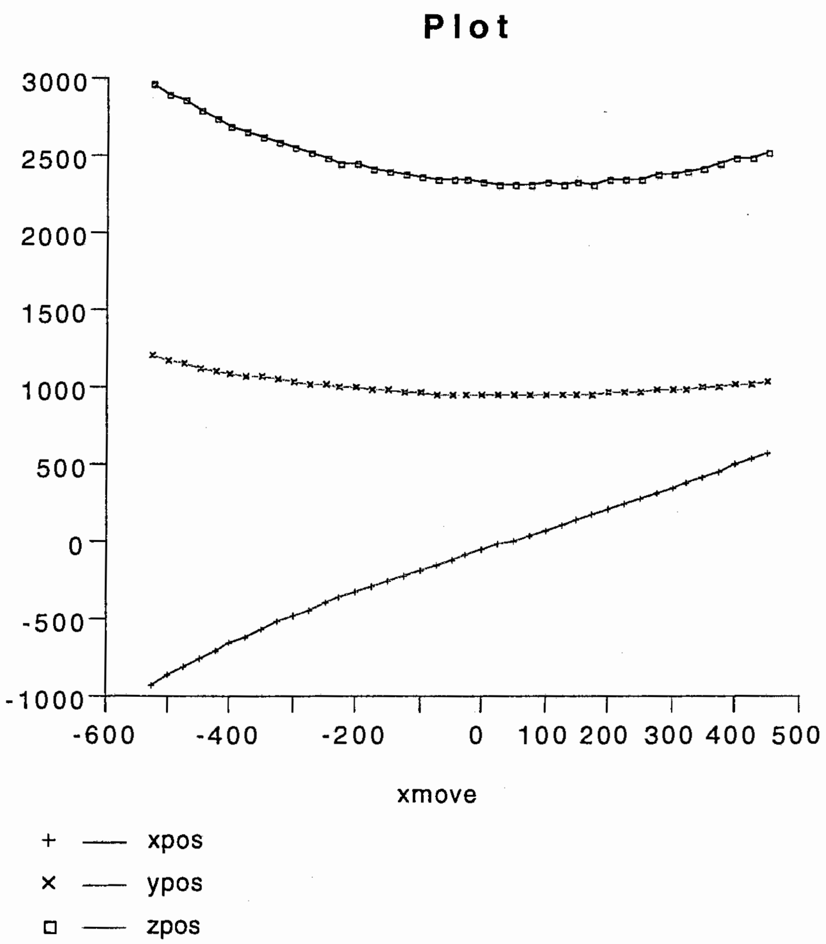
Examples of theses measurements follow, one for each axis. They show that a linear approximation is enough to correct the bias of the tracking system. We implemented all these corrections in the control program, with the ones made earlier.

We also made tests to see the variations of the tracked Z and Y according to X position, and of the tracked X and Y according to Z position. Maybe it's partly caused by the difficulty to have very precise measures "by hand", but where with should have obtained flat lines, what appeared where lines with a slight angle in all series of measures.

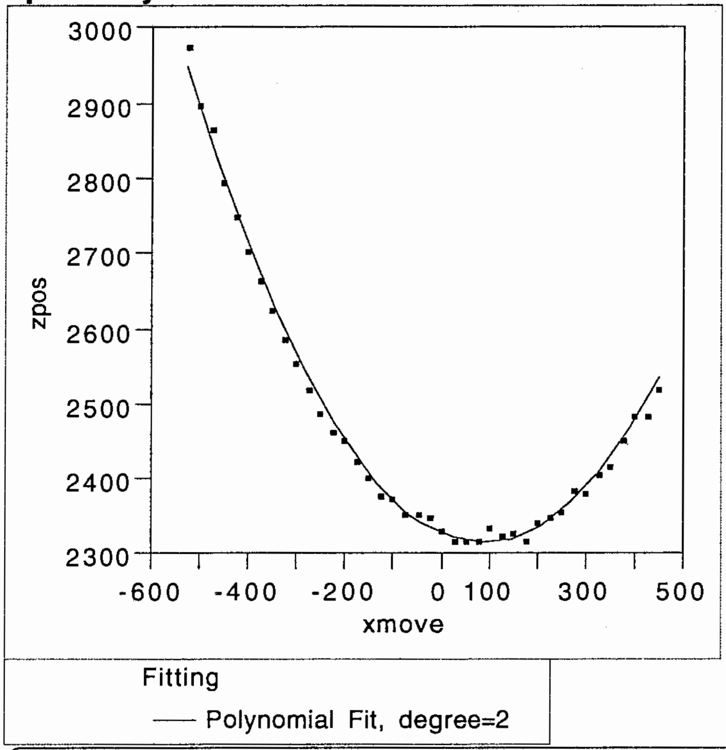
This hasn't been yet corrected in the control program, and it's small enough to be ignored. But in one case, the variations of Z position according to X pulse, the results could not be approximated by a first order function. The second order was necessary, and in fact fitted rather well. It's a strange enough deformation so that the measures were made many times to be sure. We suppose it to be caused by the lens of the infra-red cameras, that are of a low quality. For now we haven't corrected this in the control program. The reason is this would augment the calculation time and this deformation doesn't really diminish the comfort of the viewer. For what we can see, the best viewing position is stable enough along the Z axis, and we can admit some lack of precision in this direction.

This is not the case in the X direction which it is very unstable and demand that the projector precisely responds to the viewer's motions.




Plot
3000
2500
2000
1500
1000
500
0
-500
-1000
-600
-400
-200
0
100
200
300
400
500
xmove
xpos
ypos
zpos

zpos By xmove



Polynomial Fit, degree=2

Summary of Fit

Rsquare	0.99592
Root Mean Square Error	11.59866
Mean of Response	2480.499
Observations (or Sum Wgts)	40

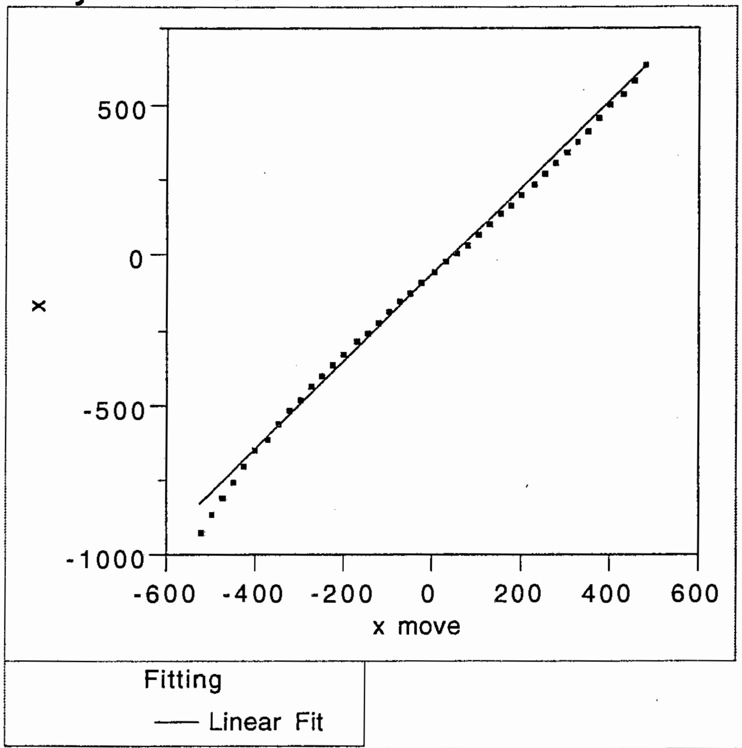
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	2	1214983.0	607491	4515.698
Error	37	4977.6	135	Prob>F
C Total	39	1219960.5		0.0000

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	2327.4425	0.00002	850.38	0.0000
xmove	-0.296035		-44.73	0.0000
xmove^2	0.0016762		68.02	0.0000

x By x move



Linear Fit

Summary of Fit

Rsquare	0.996107
Root Mean Square Error	27.4379
Mean of Response	-101.792
Observations (or Sum Wgts)	41

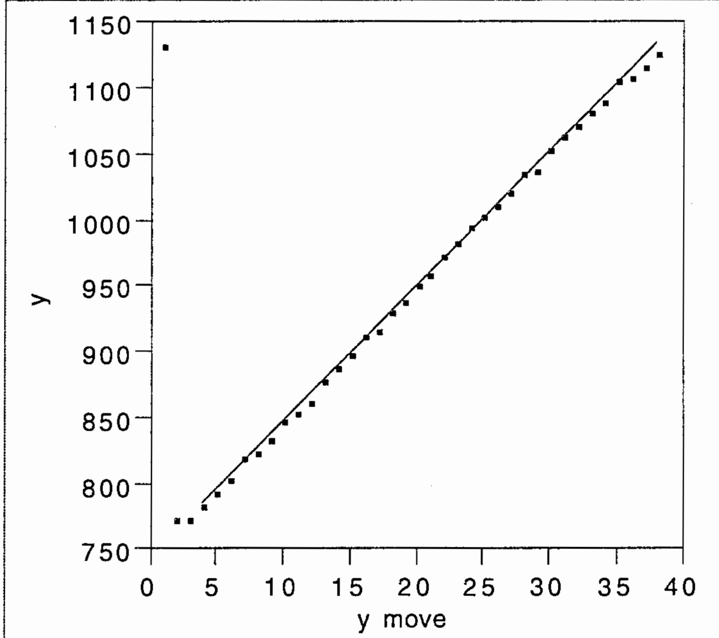
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	7512170.2	7512170	9978.462
Error	39	29360.7	753	**Prob>F**
C Total	40	7541530.9		0.0000

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	-65.61582	4.30036	-15.26	0.0000
x move	1.4470607	0.01449	99.89	0.0000

y By y move



Fitting

—— Linear Fit

Linear Fit

Summary of Fit

Rsquare	0.999155
Root Mean Square Error	3.100182
Mean of Response	959.5497
Observations (or Sum Wgts)	35

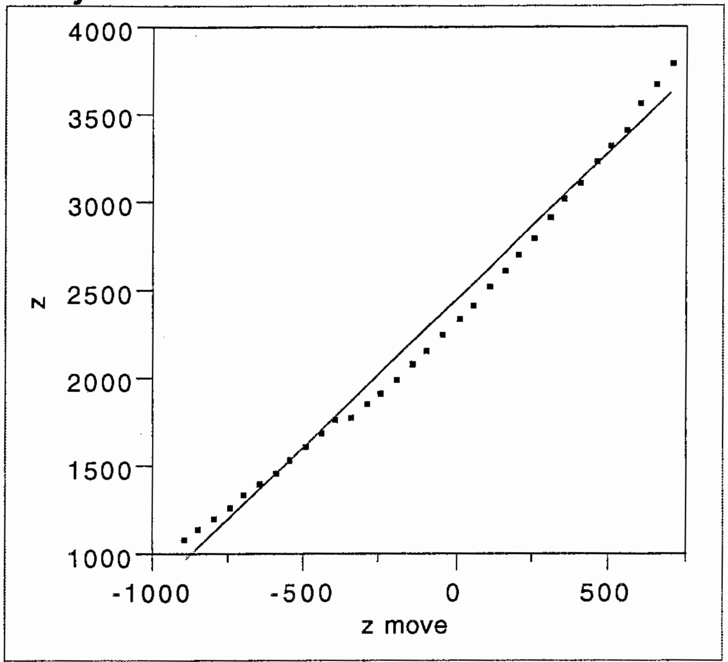
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	375156.36	375156	39033.53
Error	33	317.17	10	Prob>F
C Total	34	375473.53		0.0000

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	744.27583	1.20907	615.57	0.0000
y move	10.251137	0.05189	197.57	0.0000

z By z move



Fitting

—— Linear Fit

Linear Fit

Summary of Fit

Rsquare	0.98794
Root Mean Square Error	90.41064
Mean of Response	2287.677
Observations (or Sum Wgts)	33

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	20757546	20757546	2539.434
Error	31	253397	8174.084	Prob>F
C Total	32	21010943		0.0000

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	2454.2622	16.0819	152.61	0.0000
z move	1.665855	0.03306	50.39	0.0000

Part II: Delay compensation

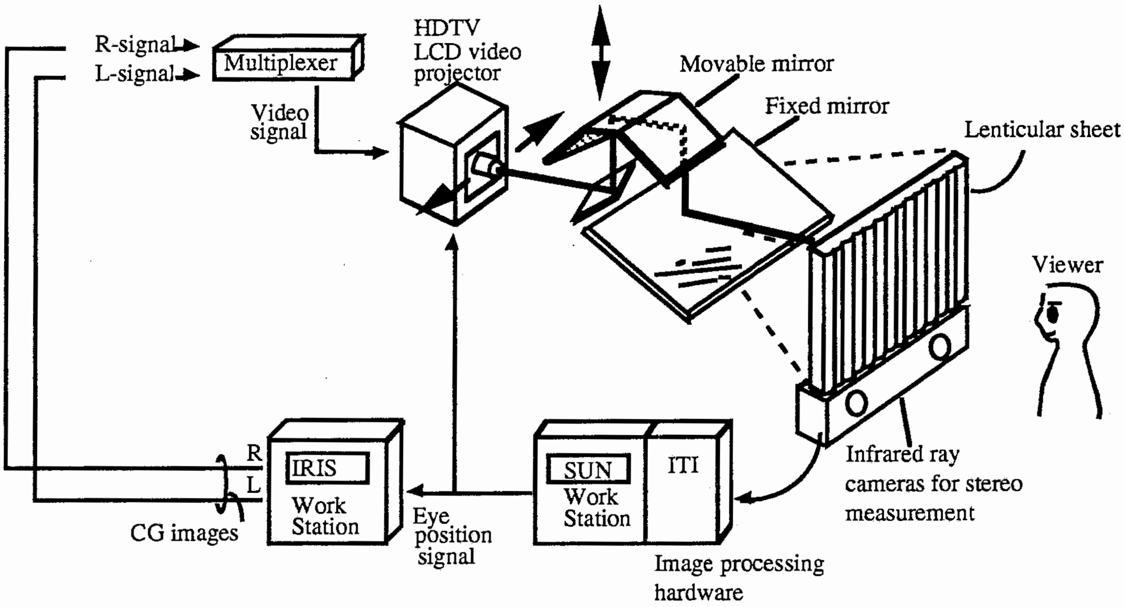
The problem of delay compensation is really important to have any comfort in viewing. This is particularly problematic in the X direction where a small lack of precision on the head and the projector respective positions causes the viewer to have the scene he is witnessing filled with vertical black stripes.

The delay is caused by computing time and by the inertia of mechanical response in the projector movement. The total delay is about 200ms, 75ms of which are computing time.

A problem in estimating it at any time during the compensation is that we cannot measure it by way of the computer clock. The control program contains a very important part which is the tracking loop. Some instructions in this loop are present to find the current position of the reflective surface in space and to send an information to the projector given in absolute two dimensional coordinates to where it must go. But in this loop there is no feedback of information from the camera. As a result it is impossible to know when its motion will end and hence there is no way to know the length of time its change of position took. Especially when the occurrence n of the loop cancel the information sent in occurrence n-1 to give a different one, even if the change commanded at n-1 did not have enough time to end. So what we had to do was judge subjectively by trying different versions of delay compensation programs one right after the other, which one offered the best results in terms of speed and comfort for the viewer.

The first thing we notice using this system is that the delay is not the same for X and Z directions. The response is slower for Z direction. This is caused by the difference of mechanism for the movement in Z and X direction.

Moves of the viewer in the X direction are compensated by motions of the projector lens on the X axis. Moves along the Z direction are compensated by changes of position of the movable mirror along the Y axis. The mechanism for Z changes is slower than the other one, and has a harder time following sharp variations of speed or direction.



(Generating CG images) (Calculating eye-position)

Fig. 3

Another important difference is that delay is more unpleasant for the viewer for X direction. Even if there is less delay than in Z direction it brings more discomfort to see vertical black stripes passing along the screen than to view the well illuminated area on the screen decreasing in surface. So maybe the method of delay correction would have to be different for the two directions.

We cannot directly diminish the inertia of the projector. All we can do is add lines of program in the tracking loop to compensate for this. But we cannot put very complicated function in it either. Not only because we don't want to augment the computing time, but also because the loop works in real time and is very sensitive. Adding too many things in it can create tracking problems. An example of this is given later.

The basic idea to suppress the delay is to make the projector anticipate the movements of the viewer, according to his last ones. Then again we don't want to store too many information and changing them at each occurrence of the loop because of the increased computing time. To begin with, we looked for a function of the following type:

If p0 is the pulse to be sent to the projector without any compensation and p1, p2 and p3 are the same at the last three occurrences of the loop, the compensated pulse cp is looked for in the following form:

$$cp = p0 + A(p0-p1) + B(p1-p2) + C(p2-p3)$$

The first intention was to fix the A coefficient to a high enough positive value and then to look for B and C negative coefficients that seemed to make a good balance for the coefficient A.

In that way, when the movement of the head had a constant speed, the projector would be sent slightly in advance of the person's head ($A>|B|+|C|$), and the faster it goes, the more advance it will take. And in case of a sudden head acceleration, the projector would accelerate too, because the factor multiplying A would take more weight compared to the one of B and C. It would be true too if B and C were positive, but the A coefficient determined at constant speed would not be so great and would not take so much preponderance compared to the two others in the case of an increase of speed of the marker being tracked.

So we tried fixing the A coefficient to 8 or more and search for the other two. What appeared were important vibrations in the system. In fact the greater the coefficients, the greater the vibrations. The following curves intend to give an impression of these vibrations. In the first case, the marker is moved along the Z axis at more or less constant speed. The compensation function for X direction is:

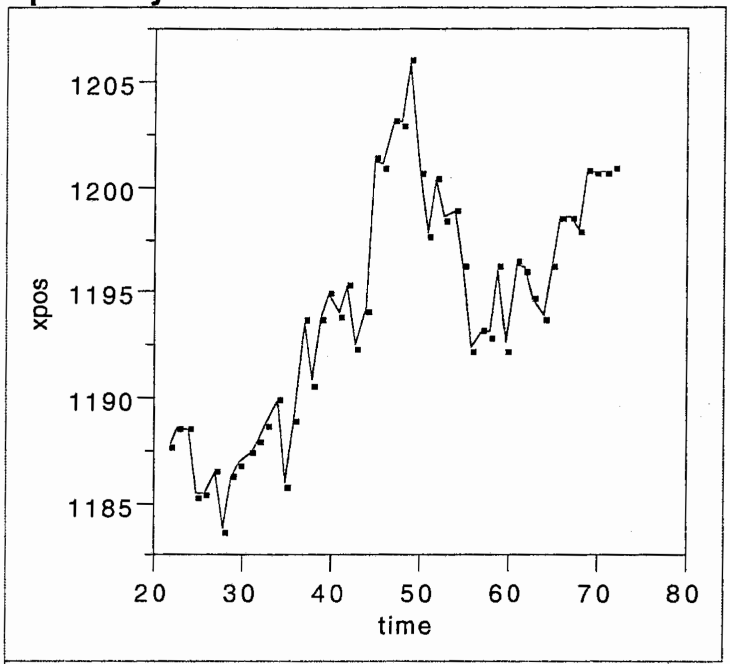
$$cpx = pox + 8(pox-p1x) - 4(p1x-p2x) -1(p2x-p3x)$$

In the second case, the marker is moved along X axis at constant speed, and the compensation function for Z is:

$$cpz = poz + 9(poz-p1z) - 3(p1z-p2z) - 1(p2z-p3z)$$

We can see that the scale of vibration is greater for Z (second case) than for X (first case), even if both greatly impede good viewing.

xpos By time

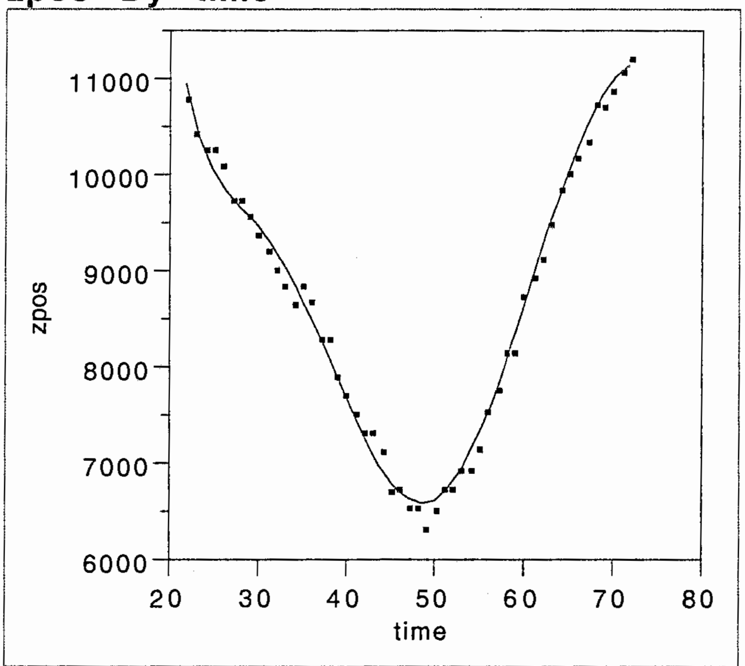


Fitting

—— Smoothing Spline Fit, lambda=0.01

Smoothing Spline Fit, lambda=0.01

zpos By time

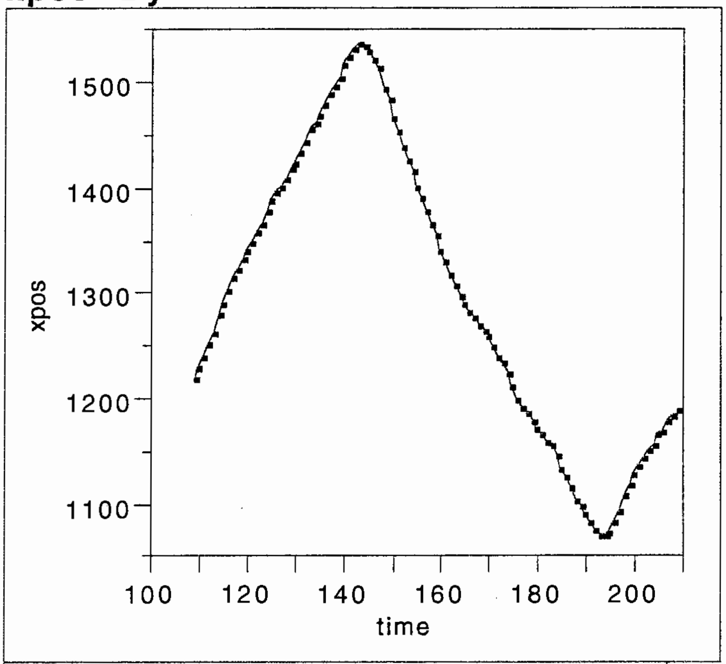


Fitting

—— Polynomial Fit, degree=6

Polynomial Fit, degree=6

xpos By time

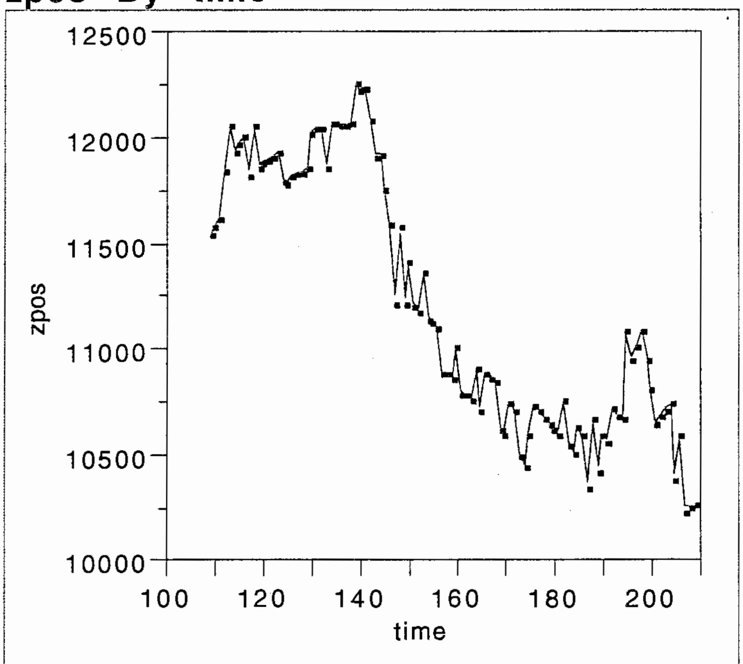


Fitting

—— Smoothing Spline Fit, lambda=0.01

Smoothing Spline Fit, lambda=0.01

zpos By time



Fitting

—— Smoothing Spline Fit, lambda=0.01

Smoothing Spline Fit, lambda=0.01

These vibrations made it impossible to have a viewable image, especially in the X direction where small changes have a great effect. So we had to find a way to get rid of them. Since the response of the projector was fast enough, we tried to decrease the A coefficient and the two others accordingly. We obtained a good enough tracking with nearly no vibration in the X direction for A=6.5, B=-2, C=-1. But the vibrations wouldn't disappear in the Z direction, and not speaking of the viewing conditions, we feared it would give the mechanism too much stress: we can see on the curves that the scale of the vibrations is far greater along the Z axis than long the X axis, and we could never decrease these to an acceptable degree.

It seemed to appear that negative coefficients put some instability in the system when the head decreased its speed, the system oscillating back and forth between next and previous position. This was especially obvious during very slow motions of the head. The image was shaking, the system unable to fix his position on the marker. So we tried to suppress the last term, to have only one negative coefficient. We looked for a function:

cp = p0 + A(p0-p1) + B(p1-p2)

With A positive and B negative. We still had a good tracking in the X direction with A=6 and B=-2. To find good parameters for Z direction, we tried to fix the difference A-|B| at constant speed, and then decrease the A coefficient until the disappearance of the vibration problem. Unfortunately, too much vibrations remained and the tracking speed decreased with A. So with A too small (2 or 3) we still had little vibrations and we didn't have a good tracking any more.

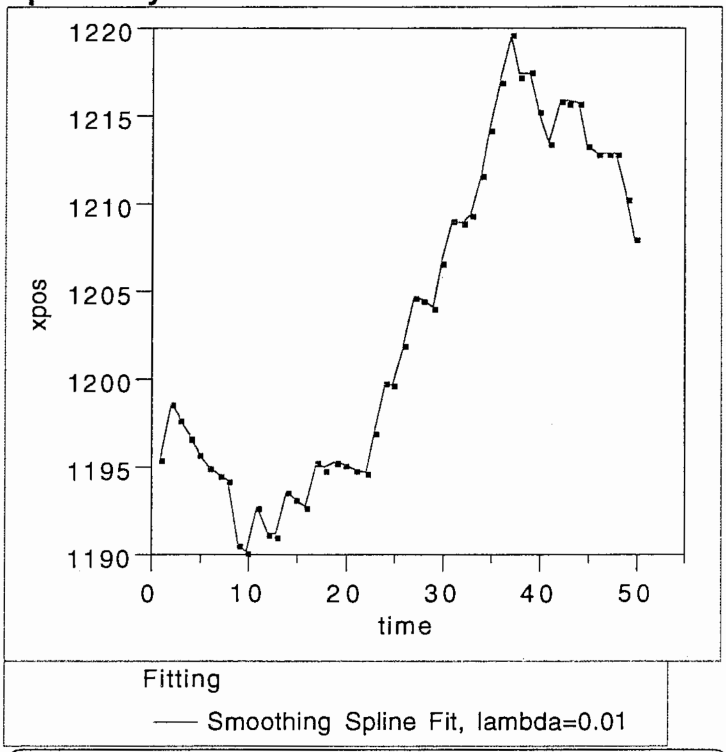
The next curves are examples of the scale of vibrations for X and Z axis. The first one presents the vibrations in X direction when movement occurs mostly along Z, with a corrected pulse:

cpx = pox + 8(pox-p1x) - 4(p1x-p2x)

The second curve presents vibrations in Z direction when the marker moves along X, with a corrected pulse:

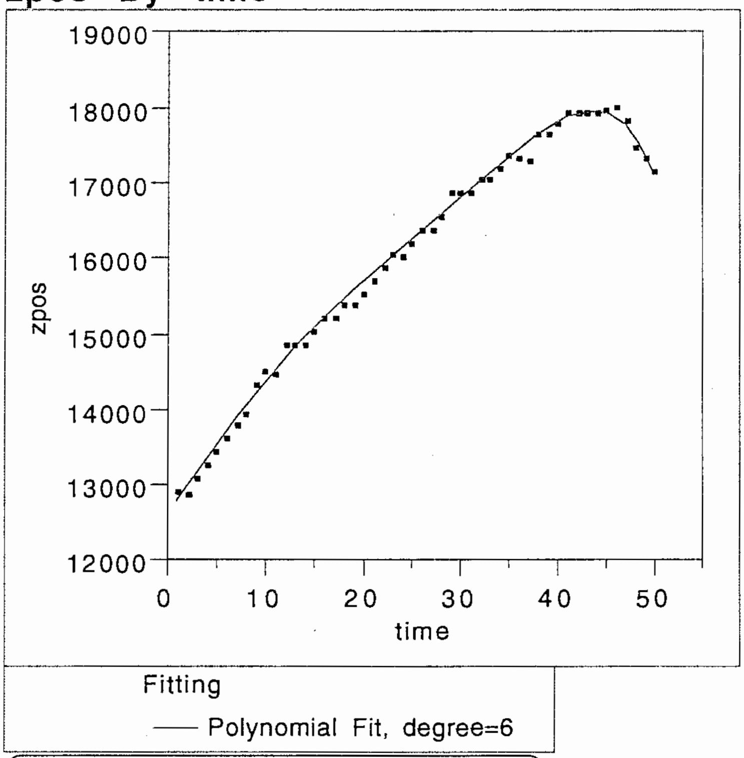
cpz = poz + 8(poz-p1z) - 3(p1z-p2z)

xpos By time



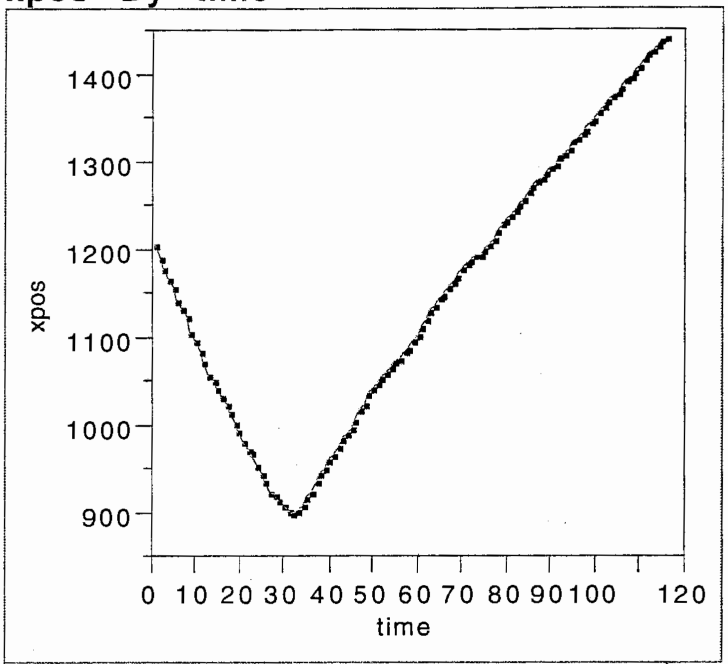
Smoothing Spline Fit, lambda=0.01

zpos By time



Polynomial Fit, degree=6

xpos By time

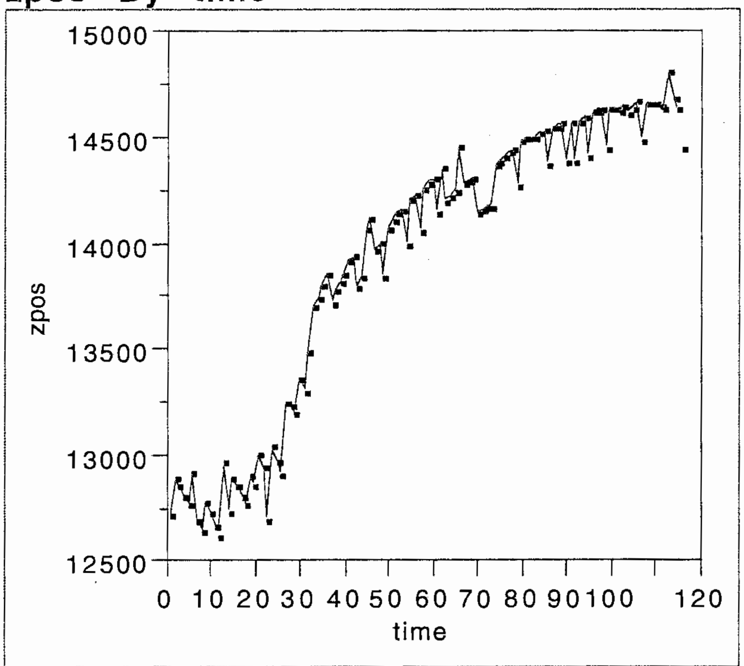


Fitting

—— Smoothing Spline Fit, lambda=0.01

Smoothing Spline Fit, lambda=0.01

zpos By time



Fitting

—— Smoothing Spline Fit, lambda=0.01

Smoothing Spline Fit, lambda=0.01

So what we wanted to do then was to suppress the vibrations brought by the negative coefficients, but we wanted to keep the speed of response they brought. So we tried to modify the program with another function, putting in fractions. The form is the following:

cp = A (p0-p1)/(p1-p2) + B (p1-p2)/(p2-p3)

With of course tests to check that (p1-p2) and (p2-p3) are different from zero, and A and B coefficient both positive.

For small values of A and B coefficients, if we don't have any visible vibration, the response is too slow so we have to increase them. What happens is that some vibrations reappear if we augment A and B coefficients, and even if they are not as important as with the first function we studied, it still prevents comfort in vision in the X direction. And if these small vibrations don't cause problems to the mechanism for Z direction, this method tends to too slow to provide good viewing even for great values of A, maybe in part because of the greater computation time in the loop.

Another problem that did appear (although a real minor one) was the borders of the screen. The diminution of delay shouldn't be diminishing the viewing area as a side effect. But if we send the camera before the head of a viewer as an attempt to anticipate its next move, we may well send it out of the borders of the screen, and get out of the tracking loop because of the mistake that occurs. To prevent that we used simple tests to see that the projector wasn't sent out the limits. The unexpected result was an lack of good tracking certainly caused by this small amount of computing time. So for now the tests are suppressed of the program, and the compensation experiments were further made without it.

As the system didn't deal very well with either fractions or negative coefficients, we finally settled for the simplest form of function we could find. That is:

cp = p0 + A(p0-p1) + B(p1-p2) + C(p2-p3)

With all coefficients A, B and C positive. And finally we found a suitable function for Z direction. What happens is that the tracking speed is nearly as good as with the negative B and C and that vibrations are nearly non existent for two reasons: the lack of negative coefficient first, but also the fact that the coefficients we need for good tracking speed are a lot smaller than the one we had in the first place (we made a difference of terms, now we add them), so the scale of the vibrations is reduced.

For good tracking in the Z direction, we have settled for A=2, B=2 and C=0. C coefficient doesn't bring any obvious improvement in tracking for Z direction.

For X direction the best coefficients we found are A=2.5 and B=1. But still, the first form of function we found seems to provide a better tracking. That is with A=6.5, B=-2 and C=-1.

So here are the final functions we used:

$$cpx = pox + 6.5(pox-p1x) - 2(p1x-p2x) -1(p2x-p3x)$$

$$cpz = poz + 2(poz-p1z) + 2(p1z-p2z)$$

The viewing is not perfect, but its the best that was achieved. Both directions provide stable viewing at constant speed. In the case of bursts of speed along the Z axis, we observe a small decrease of the viewable field. However this is really small and doesn't really affect the quality of the scenery perceived. For X direction however, sudden increase in head's velocity causes a black stripe to appear on one side of the screen. One problem is of course that it reduces the vision field of the 3D image for a small part. But mostly, having this stripe flickering in the corner of the eye is really annoying. So maybe a way should be found to get rid of this problem. Or at least the screen should be tested as it is now for longer periods of use, to see how fast the eyes tire, and to what extent viewers may feel annoyed by the lasting tracking imperfections.

CONCLUSION

We have corrected most screen and infrared cameras bias to put the lenticular to effective use. And we have tried to find a suitable way to reduce the delay of the projector's motions. But for that we had to make compromises between a good response to changes in motion and stability of viewing at constant speeds. The final result is that the observed image presents some black flickering zones, mostly on the sides, that we could not get rid of. So, even if we can obtain a fair enough viewing under "normal" circumstances (not many sharp changes in directions...), we still need to know the feelings of a viewer involved in the virtual space after a prolonged period of time.

References:

[1] N. Tetsutani and F. Kishino, "A Study of Stereoscopic Display System Employing Eye-position Tracking for Multi-viewers"

[2] N. Tetsutani and F. Kishino, "3-Dimensional Display Method without Special Glasses for Virtual Space Teleconferencing system"

[3] K. Omura, N. Tetsutani and F. Kishino, "Lenticular Stereoscopic Display System with Eye Position Tracking and without Special Equipment Needs"