OPIE USER'S MANUAL

Introduction

This manual is meant to show someone the steps involved with using the OPIE system. The user should have a fairly good general understanding of computers and cameras. A user unfamiliar with MATLAB should be careful because some of the ideas expressed in this manual and used in the analysis system are unique to that operating platform. Much of its detail involves customizing a test to provide better final results. While this system was meant to be user friendly it was also designed to be very universal and flexible. This flexibility is what makes it such a useful system, however it also accounts for most of the complexities.

Setting Up a Test

Before any testing can be done the limitations of the test must be known. For instance the OPIE software will have little chance of obtaining relevant information if the video it is analyzing is blurred by motion or has to low a resolution. Before recording video one must first make sure that the images are going to tell the whole story as well as possible. Eliminating blur and providing a high resolution are contradictory goals. The same goes for wanting to have a high sampling rate and use little of the computer's memory. These limitations demand compromise.

By running the file **param.m** through MATLAB some of these compromises may be solved. If you desire to know for instance what shutter rate you should use to eliminate blur, you can select the proper menu item, answer the questions and follow the instructions at the end. All of the menu items work basically the same and two or three questions will have to be answered to find the information.

To find:	Need to know:
Speed to avoid blur	Shutter speed, Resolution
Shutter speed to avoid blur	Max speed, Resolution
Resolution to avoid blur	Shutter speed, Max speed
Memory required	Frames rate, Test Length, Size of each frame
Length of test	Frame rate, Size of each frame, Memory
Frame rate	Distance between markers, Scaling factor, Max speed
Velocity to stay in search	Distance between markers, Scaling factor, Frame rate
Distance points should be separated	Frame rate, Max speed, Scaling factor

It is important to note that **param.m** considers blur the movement of one pixel width during a shutter period. This is a very strict definition of blur. Movement of four or more pixels might not even be very significant and therefore the blur avoidance characteristics can be followed fairly loosely.

The resolution required in **param.m** can be found easily by recording a calibration frame and then running the program **res.m** in MATLAB. This program works basically the same as the OPIE version but is in a command prompt style. The end of the program will give two numbers for resolution. The larger number is the worst case scenario for blur and should be used in **param.m**, however, the smaller number may be not meet the necessary accuracy requirements for the test so be careful when positioning and zooming the camera.

Setting up the Camera

Assuming the computer is set up correctly, the camera can be setup in the following manner. Connect the camera power supply to the camera and then connect the camera to the frame grabber via the red cable. For best results set the camera to **NON** (non-interlaced mode) and **NRM** (normal triggering mode). **INT** is for setting the camera to interlaced mode (the display method TV's use) and **ASY** is for setting the camera on asynchronous reset mode for use with external triggering. The dial between the two switches is used to change the shutter speed. The picture will get significantly darker as the shutter speed is increased and more light will be required to get a usable picture. The shutter settings are as follows:

Setting	NRM	ASY
0	no shutter	no shutter
1	1/125	1/16000
2	1/250	1/8000
3	1/500	1/4000
4	1/1000	1/2000
5	1/2000	1/1000
6	1/4000	1/500
7	1/8000	1/250
8	1/16000	1/125
9	N/C	shutter determined by double pulse

Information obtained from TM-9701 Operations and Maintenance Manual

The camera should be placed with some degree of care. It should first of all be set facing perpendicular to the expected motion. The system was designed to measure planar motion and anything else can reek havoc with the data. One should be careful of shadows, glare and other objects that the model could move behind or in front of during the test. It is very important to be careful when using a wide-angle lens to line up the camera with the center of the movement. This will help prevent distortion from the fish eye effect of the lens. If the camera must be set off center, the calibration should be done where the motion is expected to occur.

Often objects on the edge of the frame will not appear to be lined up correctly because of the problems encountered with perspective. These can be somewhat solved by moving the camera away from the object of interest and then to maintain the proper resolution the camera must be zoomed to get the same area.

Once the camera is in place and nothing else will be changed as far as positioning is concerned, the calibration and equilibrium pictures should be taken. The calibration picture should especially be taken with care. The calibration dot should be held the same distance away as the markers and should be flat as viewed from the camera. The equilibrium frame should have the relevant equilibrium positions. It is very important that during the shooting of these pictures and afterwards NOTHING SHOULD MOVE! If the camera gets bumped after this point the equilibrium picture will be off and if the zoom is changed the calibration will be off. Both circumstances could ruin an entire test.

Recording Video

After the hardware has been setup properly, it can be tested and used by going into the Sequence Recording software. After beginning the program, go to **Options**, **Select Camera**, **Non-Standard** then find the directory called **CAmCfg**. In this directory find the file labeled **Tm9700a.ini** and double click it. After clicking **OK**, the program will probably produce a warning message. The warning message refers to a problem which might occur if the selected setup needs more RAM than is available. This should not be to much of a problem for reasonable tests but the amount of RAM on the computer should probably checked before proceeding.

Once the camera has been selected go to the **Options**, **Set Record Options** selection. This menu allows the user to control the sampling rate, output type and length of the test. The frame rate is chosen by selecting a method of sampling in the **Recording Interval** section. Putting a **1**, **2**, **3**, … in the **On Frame** choice will grab every frame, every other frame, every third frame etc. By putting a time in the **On Timer** the frame grabber will grab a frame every "?" seconds. It is important to make sure the **File Format** is on the **BMP** selection because the OPIE software will not work with out it. The length of the test is controlled by the number of frames that will be grabbed and the rate at which they will be taken. Enter into the **Number of Frames** section a number which will not exhaust the amount of RAM available. It should be noted that if the **Display** option is selected, higher frequency frame rates will not record. This is easily fixed by disabling the **Display** in the **Record Options** menu. After all of this has been completed and **OK** has been chosen the same warning message as before will appear. If the numbers have been chosen correctly there should be no problem and you should proceed.

One way of saving on memory when recording video is to choose **Options**, **Set Frame Size**. In this menu you can change the number of pixels that are grabbed and thereby reduce the size of each frame without tampering with the resolution. However, be careful when using the **Scale** section of the menu set to **2:1**, as this will grab the same area but reduce the resolution by half.

Now that the software is ready to record the final preparations should be made to make sure the relevant data is being captured. By clicking on **View**, **Adjust Video** the user can observe the lighting, positioning, focus and zooming of the camera. The lighting can most easily be controlled using the F-stop on the camera lens however the lighting controls in the software may also aid in obtaining the relevant information. It should be noted that this view doesn't show the entire frame that the frame grabber and camera are grabbing. To see what is actually on the edges the user must save a frame and view it in MATLAB, MSPaint, or MSImaging. This will allow you to more properly see what is in the field of view. The other important thing to note is that the video display may not be showing the frames as fast as they are actually being captured due to limitations in the video card. This will not affect the recording in any way and the proper sampling rate should be achieved.

Once all of the previous steps have been completed the **View**, **Record Video** option should be selected. From here the recording may be begun at the user's leisure. The recording may be stopped at any time and then resumed or started over. However, once some usable video is obtained it should be saved as soon as possible to avoid accidentally deleting it. To save the video use the **File**, **Save as** option. The video should usually be saved in the **C:\MATLAB\BIN\VIDEO** directory for ease of use of the OPIE software later. The file name given to video sequences should have four characters or less and end with the ".seq" extension.

Any video that has been recorded can be watched by opening it if it is not already the current sequence. Once this has been done using the **File**, **Open** option the **View**, **Play Video** option will bring up some VCR like controls. It should be noted that many video sequences will not play and cause an error message to display when using the ">" button but will play using the "»" button. This has to do with the speed at which the video card can display the images. When the "»" is used the original timing between the frames is not preserved.

Altering Images

It may be necessary to alter one or more frames because of problems encountered during testing. Hopefully the number of alterations required are few and do not necessitate a program specifically equipped to alter video. It may be possible to do alterations to frames (even entire video sequences) within MATLAB, however its capabilities are somewhat limited. Another method of altering frames is to use MSPaint. This program provides enough tools to let you do most jobs. The only problem with this approach comes in saving the altered pictures. MSPaint doesn't save pictures with a color map. A color map is a sort of look up table that tells the computer what colors to use to reconstruct the picture. MATLAB cannot reconstruct the picture without this information. It is necessary when taking this approach to altering pictures that you save the frame upon exiting MSPaint and then go to MSImaging (also under the Accessories group). In this program you will simply want to re-**open** and then re-**save** the picture.

This will attach a color map to the image and will allow MATLAB to open the file successfully .

Running the Analysis

The video has been saved and the analysis is ready to be done. After starting **MATLAB**, type **opie** at the prompt to start the program.

On the information page of the program type the four or less character prefixes for the file designations. The number of frames must be a number less than or equal to the number of frames that were taken in the video sequence. The program will always start with the first frame in the sequence and finish with whatever number you type here. If you put a number in which is too large the program will kick you out after it reaches the end of the video sequence. This wastes a lot of your time because the little information you get will not be complete. The sampling rate should be in frames per second and can be any number. However if this number is wrong then the velocities and accelerations, not to mention the time scale will be wrong also. The units of the diameter of the calibration dot are important only on the graphs. The axes on the graph will be labeled with whatever you enter into this slot. The diameter of the calibration circle should be measured carefully. This number will help determine number of pixels per distance. If this number is off then the calibration will also be off.

When calibrating one should try and get the computer to recognize the entire calibration circle. The system will find the row with the most amount of recognized pixels as well as the column. It perceives these as the widest part of the calibration circle and therefore equal to the diameter. It is important to get the calibration right because while the computer may be able to accurately determine distance in pixels, it cannot translate this to a useful measurement system without a good calibration.

The equilibrium of the markers is relatively arbitrary. The equilibrium can be chosen as the place where the markers are at rest or some other fixed point. The displacement of each marker is measured from these equilibrium points so it must be taken into consideration when displacement is crucial. The reference angle used to compute angular displacement is also calculated from these points. If displacement from a fixed point needs to be known and the angular displacement is also crucial then the analysis should be run twice with the equilibrium point first in the best place for displacement and then with the equilibrium point on the static position of the markers.

To start the actual analysis the markers must be located for the computer. Once this is done the program will automatically locate the rest of the positions of the markers from subsequent frames.

Sometimes the computer will not find the marker but a shadow or some other inconsequential spot. When this occurs it is often a result of the wrong size search area. On each frame the program searches two circles, one for each marker, each centered on the marker's last known position. The diameter of the circles is equal to the distance between the markers and so the radius is equal to this diameter divided by two. The number by which the diameter is divided can be changed by going to **File**, **Open M-file** and opening **shan.m** in the **C:\matlab\bin\opie** directory. In the top of this file there is a line that determines the size of **ratio**. You can change the value of **ratio** and then save the file under the same name. If you increase the value of **ratio** the size of the search area will decrease.

While it is often important to keep the search area small, if it is made to small the markers may not be found for another reason. If the search area is small and the velocities involved are large then the markers may move out of the search area in the time between two frames. To counter act this the value of **ratio** should be decreased. For the integrity of the data **ratio** should not be smaller than two because this would allow for the possibility of having both markers in the same circle.

You may find out that a shadow appeared or some other problem occurred confusing the search algorithm. Because of this possibility it is often beneficial not to dismantle the experiment until you have at least run a trial analysis. This means you should not touch anything with the setup: lights, focus, zoom, position of the camera or model. This may help prevent you from wishing you had another chance to do it exactly the same because of one small error.

Output

After the analysis is done, OPIE provides graphs with some useful information. The first graph represents the path the markers followed. This graph uses the variable B(f,4) (The variable name is B and it is of size f by 4. All arrays are labeled likewise in this manual.) and plots out the position of each marker at all the time steps.

The next graph displays the displacement from equilibrium for each of the markers. The x-direction and y-direction displacements for each marker are shown. These displacements are stored in the array D(f,4) and are found by subtracting the equilibrium positions (xref(4)) of the markers from their position for each of the time steps B(f,4).

The next graph created contains velocity information. This graph shows the x-direction and y-direction components of velocity (vel(f,4)) for each marker. These velocities are found by dividing a backward difference of the positions (B(f,4)) by the time between frames (1/fps). The next graph is much the same except it displays components of acceleration (accel(f,4)) which are found by a backward difference of velocities divided by the time between frames.

The total speed and total acceleration for each marker are displayed on the next graph. The total speed for each marker (veltot1(f),veltot2(f)) is found by adding the squares of the component velocities and then taking the square root. The total acceleration (acceltot1(f),acceltot2(f)) is done in nearly the same way with the components of acceleration.

The angular displacement of the line between the markers from the line between the markers at equilibrium (theta(f)) shown on the next graph is found by using the *atan2* function in MATLAB. This function computes angles from 0 to π and from 0 to $-\pi$ in radians. After this was computed using the difference in position (from B(f,4)) the reference angle found in the same manner (from xref(4)) was subtracted out.

The last graph provided by OPIE shows the angular velocity (omega(f)) and angular acceleration (alpha(f)). These were found using backward differences on theta(f) and omega(f) respectively and are in radians/sec or radians/sec^2.

It may often be beneficial to save certain or all of these variables for use and manipulation later. If the variables will be imported to Excel or some other spread sheet use the MATLAB command **save filename variable variable … -ascii -tabs**. This will save the variables listed in the file "filename" in an ascii, tab delimited format. If the variables will be used later in MATLAB then the **save filename variable variable …** format can be used. When saving in the tab delimited format, be careful to note which variables were saved, how big they are and what order it was done in because no headings will be given and they will be listed one after another.

Glossary and List of Variables

A(m,n)	An array which contains the grayscale information for the current frame. Each element in the array is the grayscale value at a corresponding pixel.
accel(k,4)	An array containing acceleration components for each marker in an X1Y1 X2 Y2 format.
acctot1(k)	An array containing the sum of the x and y components of acceleration for the first marker.
acctot2(k)	An array containing the sum of the x and y components of acceleration for the second marker.
alpha(k)	The angular acceleration of the two dots with respect to each other.
B(k,4)	The position of the markers in each frame arranged in rows of X1 Y1 X2 Y2.
b(4)	The predicted value of the markers in the next frame. In the LPP search this is the last known position of the marker.
C(?,?)	This is used more than once. It is usually some sort of copy of A(m,n) or a subset of it.
cal(4)	A vector containing the x and y calibration factors in the format xcal ycal xcal ycal.
calx	The calibration factor for the x-direction. In inches/pixel or meters/pixel etc.
caly	The calibration factor for the y-direction. In inches/pixel or meters/pixel etc.
cfile	The file name used to identify the file containing the calibration information. The default for this file is “C:\matlab\bin\video\ “cname” 0001.bmp” To change this look for the line “cfile = [‘C:\matlab\…’]; in the file “calb.m” and change it to the directory of choice.
cname	The four character prefix used to identify the calibration file.
D(k,4)	The x and y displacements from equilibrium arranged in rows of X1 Y1 X2 Y2.
d	The distance between the markers on the first frame in pixels.

dact	The distance between the markers on the first frame in inches or meters etc.
diam	The diameter of the calibration dot.
efile	The file name used to identify the file containing the equilibrium information. The default for this file is "C:\matlab\bin\video\ "ename" 0001.bmp" To change this look for the line "efile = ['C:\matlab\…']; in the file "equi.m" and change it to the directory of choice.
file	The file name used to identify the current frame in the video sequence. The default for these files is "C:\matlab\bin\video\ "fname" "k"… .bmp". To change this look for the lines "file = ['C:\matlab…']; in the file "shan.m" and change it to the directory of choice.
fname	The four character prefix used to identify the video sequence.
fps	The number of frames per second and the inverse of the sampling period.
i	A counter for tracking the rows of A or C.
j	A counter for tracking the columns of A or C.
k	Counter for tracking which frame the process is on.
m	The number of pixels vertically in the frame.
n	The number of pixels horizontally in the frame. Usually 11 will be subtracted from this number to give the number of actual usable columns. It appears that some information is lost in the conversion to a BMP and shows up as 11 black lines at the right edge of the frame.
numpnt	A number defining how many markers are being used. It should probably not be changed from two unless the user has an excellent understanding of the program.
omega(k)	The angular velocity of the two dots with respect to each other.
ratio	A number which helps determine the size of the search area. . The size of the search area is determined by the distance between the markers. The radius of the search area is d/ratio. To prevent the two markers from entering the same search area, "ratio" must be greater than two. However if "ratio" is to large the markers will move out of the search area in the time between frames. "ratio" can be changed by finding the line in the file "shan.m" which looks like "ratio=…;".

shutter speed	The length of time the CCD chip is allowed to gather light. If the shutter is set for 1/1000 then the chip will gather light for 1/1000 of a second. The faster the shutter the less light is absorbed and the less blur occurs.
t(f)	A vector of time starting at 0 and determined by multiples of 1/fps.
theta(k)	An array containing the angular displacement of the two dots from the equilibrium angle.
theta_ref	The angle made by the two markers in their equilibrium position.
trigger	A signal that tells the camera to open the shutter and begin taking a picture. If the triggering is set for every 1/5 of a second then the camera will take 5 frames per second.
units	A string containing the type of units which the calibration dot diameter was measured.
vel(k,4)	An array containing velocity components in a X1 Y1 X2 Y2 format.
veltot1(k)	An array containing the total speed using the sum of the x and y components of velocity for the first marker.
veltot2(k)	An array containing the total speed using the sum of the x and y components of velocity for the second marker.
xref(4)	The equilibrium position of the markers.