Rapid Electro-Optical (EO) TPS Development in a Military Environment James McKechnie^a, Alan Irwin^a, Thomas Gauntner^b

^aSanta Barbara Infrared, 30 S Calle Cesar Chavez, Suite C, Santa Barbara, CA 93103 ^bUS Army - ARDEC, Picatinny Arsenal, NJ 07806

ABSTRACT

Santa Barbara Infrared, Inc. has deployed IRWindows as an Electro-Optical test development and execution environment for military Test Program Sets (TPS). Advantages of TPS development for EO systems in the IRWindows environment are seen compared to the TPS development in ATLAS. The advantages of the IRWindows environment are:

- 1. Faster learning curve (graphical user interface is easier than test line interface)
- 2. Faster TPS development time (real time changes and asset control interface allows for faster development)
- 3. Asset control panel allows user to control assets real time and monitor all asset functions during development
- 4. Unit Under Test (UUT) image viewer allows user to set test parameters like Region of Interest more easily and more precisely
- 5. Continuous mode tests (like MTF allows user to real time adjustments)
- 6. Open architecture for test modifications

This paper will outline the details of how these advantages are utilized and how not only development time is decreased but also how test execution time can be minimized making traditionally long TPS run times on EO systems more efficient.

Keywords: IRWindows4, TPS Development, E-O Testing

1. INTRODUCTION

Test Program Sets (TPS) encapsulate the hardware, processes, and instructional material necessary for testing specific devices and assets deployed by the military. For this discussion, the hardware consists of the interface devices necessary to connect the unit under test (UUT) to the designated automatic test equipment (ATE) and the processes are captured in the software executed on the ATE. TPS development has traditionally required long development times and in the case of electro-optical (EO) UUT's, long execution times.

In a previous paper¹, the various models of test development were discussed along with their relative advantages and disadvantages. In this paper, a practical example compares the development of a TPS using an older, ad-hoc model to the development of a replacement TPS using a modern system with an alternative development model (UUT Centric). Our goals are the reduction in TPS development time and faster TPS execution time.

IRWindowsTM 4 is the latest generation of SBIR's test control application program. The software provides a framework for individual sensor test development, test sequence development, and automated factory testing. It is used to develop a TPS that will replace an older TPS currently in use. The development time and the time for executing the TPS are compared to the original development and execution times of the existing TPS.

Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXIII, edited by Gerald C. Holst, Keith A. Krapels, Proc. of SPIE Vol. 8355, 83550C · © 2012 SPIE · CCC code: 0277-786X/12/\$18 · doi: 10.1117/12.924992

2. IRWINDOWS DEVELOPMENT ENVIRONMENT

To utilize the EO TPS environment in IRWindows, a set of sequences (meaning a list of EO tests) were created in IRWindows and saved in the IRWindows application. All sequences were created and validated in IRWindows. Once this integration was completed, execution of these sequences was executed from National Instruments' TestStand using function calls. This technique allowed a user to use the EO friendly graphical interface in IRWindows.

An example of the easy to use graphical interface is below (Figure 1). This screenshot shows the list of canned tests (called sequences in IRWindows) the TPS Developer can choose from. The TPS Developer selects the applicable test and adds parameters associated with the UUT.

R Sequence Control Script: N	IGATS DVT INFRARED		
File Edit View Debug			
Sequence Steps Errors			Sequence Description
H 🖀 🚸 🗙 🙇			
Name	Test Name		
Auto-Alignment	Infrared Auto-Alignment		
Cont MTF	Infrared/MTF		Pause after test configuration evenutes
FOV Square	Infrared/FOV Square		Start videos automatically
MTF	Infrared (MTF		Test Dores
P Boresight	Infrared\Boresight		Eveniece SITF
Boresight 2	Infrared\Boresight		Eyeplece Uniformity
MultiBoresight	MultiSpectral/MultiBoresight		Solution Solution
Disable Infinity	Tools\Disable Asset		🛞 🌃 Image Alignment
Manual MRTD	Infrared/Manual MRTD		Sinterpolated MTF Magnification NFOV
SITF	Infrared/SiTF		🛞 😹 Magnification WFOV
NETD Temporal	Infrared/NETD Temporal		G Manual MRTD
P Enable Infinity	Tools\Disable Asset		MTF Phase Corrected
Install IR Hood Source			
P Flood Source On	Tools Vasset Controls		+ Spatial
INETU Spatial	Infrared UVE TO Spatial		E SNETD Temporal
a 3D Noise	Intrared SD Norse		* S Radiometric
Prood Source Uff	Tools Vasset Controls		Resolution IR
Remove IR Hood Source	**		Stapped Channels
Install LST			+ 🚽 Uniformity
P LSI ON	Toois Lasset Controls	•	Soon
Step time: 1 second	ła –		🔹 🔓 Laser
 Configuration 			+ / Mirage + / MultSpectral
Frames to Average		16	
Temperature Mode		Radiometric Long Wave	Version: 1.68
Temperature Setpoint (°Q	-5.0	Uses an edge measurement
Image Center (pixel)		0,0	Configuration
MTF Correction (cyc/m	ad(MTF)	0 points	Frames to
Continuous			Average
Field of View (mrad)		148.8.123.1	Mode Differential
Nyguist Limit (cyc/mrac	9	11 6	Temperature 50
Tarnet		Tarnet (\$10 must course w)	Setpoint (*C)
iarge.		in Ani Corto unan Admut	Image Center 159.5, 119.5
* Reporting			MTF Correction
andoning			(cyc/mrad(MTF)
Pedestal			Continuous
ROI (pixel)		UL 302, 226 LR 331, 250 V .	Field of View 100, 100
Orientation		Horizontal v	Nyauist Limit
			Line: 1 Column: 1

Figure 1: IR Windows Sequence Editor

The ATLAS interface currently utilized in the IFTE V5 is a text interface. Parameters sent to the hardware to control target projector functions and video parameters to fine tune the region of interest to perform the EO tests are provided in a text format. Feedback to the TPS developer on where the target appears and detailed portions of the video the tests are being performed are not available. To "fine tune" the particular sequence the operator is required to change parameters in the test, rerun the test, and evaluate the results. There is no intermediate feedback during the testing to help the developer.

IRWindows tests provide a more intuitive graphical interface to the TPS developer providing a real time video capture of the UUT scene allowing the developer to fine tune important test parameters. This feature significantly reduces

development time. This also allows the TPS developer to change parameters on video previously captured also reducing development time.



Figure 2: IRWindows Asset Control Screen



3. EXAMPLE

Modulation Transfer Function (MTF) is used to find the position of best focus of an infrared imaging system. IRWindows allows the developer to view the captured image, fine tune the region of interest the MTF is being calculated as well as other parameters (like frames to average and image filtering) to best fit the MTF value. Additionally, a continuous mode is offered allowing the operator to adjust UUT parameters (like focus) to maximize the MTF value. This significantly decreases test development time compared to simple text inputs. Also, the TPS developer has the option of including the continuous mode in the TPS, allowing the operator to best focus the system during TPS operation and then executing a MTF for captured results.

Examples of a screen shot showing the image viewer with the target for MTF is below (Figure 4).



Figure 4: Image Viewer Screen Shot with MTF Target

The resultant MTF is shown below in a graph (Figure 5). The TPS developer can view the results in either a graph or table format and make changes on the captured image to fine tune the results.



Figure 5: IRWindows Screenshot of MTF Results

Simply put, a TPS developer is able to set up a target projector set of parameters, capture UUT video, perform a test, and then fine tune important parameters with this information. Once this fine tuning process is complete, the test can be validated. This approach is significantly more efficient than the ATLAS approach where a test is performed, executed, results returned with no feedback of how the values were calculated.

4. ADDITIONAL FUNCTIONS

There are two additional functions that IRWindows provides. One is continuous mode. IRWindows has multiple tests that can be utilized in continuous mode. The 2 most important of these tests is a continuous boresight and continuous MTF. These tests allow a TPS user to fine tune UUT parameters (such as focus in the MTF case) in real time while providing test result feedback. This feature allows flexibility in both TPS development time as well as TPS run time.

Another feature in IRWindows is the open architecture of the test algorithms. IRWindows allowed a skilled developer to modify the test algorithms within IRWindows. Modifications of IRWindows algorithms does not affect the base line test, it simply creates a duplicate test with the parameters changed by the user.

A screen shot of the open architecture is below (Figure 6).

Image:	Test Scri	fest Script: MTF										
<pre> li i i i i i i i i i i i i i i i i i i</pre>	jie Edit View Debug											
In Substrate • Meeter * Austral • Meeter * Austral • Meeter * Austral In Substrate In Substrate • Meeter * Austral • Meeter * Austral In Substrate In Substrate * Meeter * Austral • Meeter * Austral In Substrate In Substrate * Meeter * Austral * Meeter	HA	(1 × 2 = ×	(PR PR A.	课 课 IIA IIA	3 25 1 3	Version: 1 . 68						
<pre>in features is features i</pre>	Class	t TestScript							Member SunTest		Description	
i i Petersläge Petersläge Petersläge i Petersläge Petersläge Petersläge Petersläge i Petersläge Petersläge<		int Smoothing:							- Minutes - Minutes		Description	
in the stand of the stand		int Pedestal;								^	Modulation Transfe	r huction
<pre>print the function of the</pre>		string Transform	Filter;								calculate the spatial	resolution of a system.
Note:		string display_pr	rint_results;									
is the frage formats										1	Parameter List Previ	ew
Tracklasser tracs; Tracklasser tracs; Tracklasser tracs; Tracklasser tracs; (file:searchest constit); (file:searchest constit); Tracklasser tracs; Tracklasser tracs; (file:searchest constit); (file:searchest constit); Tracklasser tracs; Tracklasser tracs; (file:searchest constit); (file:searchest constit); Tracklasser tracs; Tracklasser tracs; Tracklasser tracs; (file:searchest constit); (file:searchest constit); (file:searchest constit); Tracklasser tracs; Trackl		PointF roiOffset	;								4 Configuration	
<pre>split using investigations containing</pre>		TraceListener tra	ace;								Frames to	1
<pre> this trace : control : Target trace : trace:: this trace : control : Target trace:: this trace : trace:: this trace : trace:: this trace : trace:: trace::</pre>	6	public void RunTe	est(TestScriptConte	xt context)							Average Temperature	[Differential and
<pre>res.bitstate(=</pre>		this.trace =	context.Trace;								Mode	Dimerential
Trees within ((chines, ******)) (fit (chines, *******)) (fit (chines, *******)) (fit (chines, *******)) (fit (chines, *******)) (fit (chines, ********)) (fit (chines, ********)) (fit (chines, ********)) (fit (chines, ************************************		trace.WriteL:	ine("	");							Temperature	5.0
<pre>// if is contagenetic to transformed in the section is a section</pre>		trace.WriteL	ine(context.TestNam	e + " v" + conte	ext.TestScriptV	ersion);					Setpoint ("C)	
<pre>setSort/pertisourst.Configureting); detExpertise(Sortet.Confi</pre>		// Get Confi	guration Parameters	7,							Image Center	0,0
<pre>ittleprtPress(context.hserife); titlep</pre>		GetConfigurat	tion(context.Config	uration);							(pixel)	
<pre>min interf = Transit Literation interf = Transit Lite</pre>		GetReportPari	ams(context.Reportion	ng);	TT						MTF Correction	0 points 🔛
<pre>// Intervaliate Frame Scheer // Intervaliate Frame Scheer // Streng S</pre>		this.roi = to	empRoi.GetROIOffset	(ROIUtil.GetOff:	setPoint(tempRo:	i.IsOffset, "RDI Off	Fset", contex	t.Measurem	ents, true)).ToRectangle();		Continuous	
if remain name: if remain nam		// Testantia	te Frame Grahher								Field of View	
<pre>// verify valid cording // verify valid cordinations // verify valid cordination // verify valid verify verif</pre>		<pre>// Instantist rime onabor Iframedrabor "remedrabor"(AssetManager.det().HindssetMytype(typen(UFramedrabor));</pre>									(mrad)	0,0
<pre> f((fillsDifters 1 + eff) (fillsDifters 1 + eff) f(fillsDifters 1 + eff) f(fills</pre>		// verify vai	lid config								(cyc/mrad)	0.0
for the second diget of the product the operation will be operatine will be operation will be operation will be operatine will be		if ((FieldOfView.X == 0#) (FieldOfView.Y == 0#))										
instant instruction in Testion (attorned initial organization) Imperind initial organization in the initial organization initi i i i i i i i i i i i i i i i i		(mile Bernarder Services Site of View and the service and the service)									Target	larget V
return; // Strip Canada Objects // Strip Canada Objects // Strip Canada Objects rrvce:vitite(rStrip) rvce:vitite(rStrip) // Strip Canada Objects // Strip Canada Objects rvce:vitite(rStrip) Strip Canada Objects // Strip Canada Objects // Strip Canada Objects rvce:vitite(rStrip) Strip Canada Objects // Strip Canada Objects // Strip Canada Objects rvce:vitite(rStrip) Strip Canada Objects // Strip Canada Objects // Strip Canada Objects rvce:vitite(rStrip) Strip Canada Objects // Strip Canada Objects // Strip Canada Objects rvce:vitite(rStrip) Strip Canada Objects // Strip Canada Objects // Strip Canada Objects rvce:vitite(rStrip) Strip Canada Objects Objects Description // Strip Canada Objects 1 France to Average Temperature Mode Temperature Mode Temperature Mode Temperature Mode Temperature Mode Temperature Mode Configuration oper perminante Other Strip Objects Other Strip Objects // Strip Objects 2 Temperature Mode Temperat		context.	HistoryOption = Tes	tScriptContext.	istoryOptions.	NoHistory;	tering cente //				 Reporting 	
j Strate Contract Objects Pectral 0 0 // Strate Contract Objects Strate Strate Contract Objects Pectral 0 0 // Strate Contract Objects Strate Strate Contract Objects Pectral 0 0 0 // Strate Contract Objects Strate Strate Contract Objects Pectral 0 0 0 0 // Strate Contract Objects Strate St		return;									Smoothing	0
// Strip Command objects		1								Pedestal	0	
Image: Center Frage: Configuration Configuratin Configuration Configu		// Setup Command Objects								ROL(sizel)		
Contraction Contr		trace.WriteLine("Setup command objects");								nos (pixel)		
V(Consumbly Strategitable*): Consumptions - Yster("NBMDe*): Consumpti	AsseringList tags = Assertinger.Oet().Asserings; CommandOb(b)Deep = null;								Orientation	Horizontal		
Construction Construction<		//commonly broker tas;"Bilode";										
Parameter Since Parameter Since Param		CommandObj targetiheel = tags["WheelPosition"];										
Presente: Binori Presente: Second Description												
Image: Note of the Arrows Display Name Display Name </th <th>Paramete</th> <th>s Errors</th> <th></th>	Paramete	s Errors										
Note Opping Name Data Name <thdata name<="" th=""> <thdata name<="" th=""> <thdata< th=""><th></th><th>X 10 CH 11-</th><th>44</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></thdata<></thdata></thdata>		X 10 CH 11-	44									
1 Faneta b-Arrow	Order	Scripting Name	Display Name	Param Type	Section	Units	Default	Options	Description			
2 Temperature Mode Temperature Mode <t< th=""><th>1</th><th>Frames to Average</th><th>Frames to Average</th><th>Integer</th><th>Configuration</th><th>NONE 0</th><th>1</th><th>8</th><th></th><th></th><th></th><th></th></t<>	1	Frames to Average	Frames to Average	Integer	Configuration	NONE 0	1	8				
Independent Antion Production Straine Product	2	Temperature Mode	Temperature Mode	Disk! ist	Configuration	_	Differential					
3 Temperature Steps: Temperatu	2	Temperature wode	remperature wode	PICKLIST	Configuration	-	Differencial	×.				
4 Image Center Image Center Image Center Configuration One (piele) 0.0.00	3	Temperature Setpoir	Temperature Setpoint	Double	Configuration	degree Celsius (diffe	5.0	-				
5 MT Correction MT Correction Multicator Configuration - Factor Configuration - Configuration 6 Configuration Bolgean Configuration - Factor Factor Configuration 7 Factor Factor View Paint Configuration olice per militandar - - 8 Ngapat Limit Ngapat Limit Double Configuration olice per militandar - - 9 Target Target Objectivitation configuration - - Paint 10 Smoothing Smoothing Integer Reporting NONE 0 0 Paint 11 Pedetal Integer Reporting NONE 0 0 Paint	4	Image Center	Image Center	Point	Configuration	pixel (pixel)	0.0, 0.0	-				
6 Continuous Continuous Continuous Continuous Continuous Continuous 7 Field of Vee 8 Nyquist Limit Nyquist Limit Double Configuration Golds of Vee - 7 Field of Vee Teget Teget Operation Operation - - 9 Teget Teget Operation NoRe 0 P P 10 Smoothing Smoothing Reporting NORE 0 P P 11 Pedetal Integer Reporting NORE 0 P P	5	MTF Correction	MTF Correction	PointList	Configuration	cycles per milliradiar	-	-	Correction Curve (typically set to 0, or values from			
7 Field of View Field of View Point Configuration Imiliadian (mail 00.00	6	Continuous	Continuous	Boolean	Configuration	-	False	-				
8 Nyext Link Nyext Link Configuration Opders primited are 0.0 - The Hypoint link of the UUT 9 Target Target Object/Relikitk Configuration - - De 10 Smoothing Smoothing Smoothing Reporting NNNE 0 0 De 0-20 11 Pedetal Integer Reporting NNNE 0 0 De 0-100	7	Field of View	Field of View	Point	Configuration	milliradian (mrad)	0.0, 0.0	-				
9 Target Target ObjectRubble Configuration 10 Smoothing Integer Reporting NONE 0 0 (P) 0 - 20 11 Pedetal Integer Reporting NONE 0 (P) 0 - 100	8	Nyquist Limit	Nyquist Limit	Double	Configuration	cycles per milliradiar	0.0	-	The Nyquist limit of the UUT			
10 Smoothing Integer Reporting NOHE () 0 () 0 -20 11 Pedestal Integer Reporting NOHE () 0 () 0 -100	9	Target	Target	ObjectPickList	Configuration	-	-	2				
11 Pedestal Pedestal Integer Reporting NONE 0 0 P 0-100	10	Smoothing	Smoothing	Integer	Reporting	NONE 0	0		0 - 20			
	11	Periertal	Padartal	latagar	Reporting	NONEO	0		0.100			
	**	- cochar	- Colorado	priede,	mponing	none g	•		v			

Figure 6: IRWindows Screenshot of Open Architecture

For Autotestcon on October of 2011, 5 tests were created using the NGATS test system under IRWindows control and testing the CROWS (Common Remotely Operated Weapon Station) TIM (Thermal Imaging Module) UUT. The EO TPS was developed in IRWindows and executed from TestStand with the results saved to a common SQL database.

Below is a screen shot of an EO TPS being executed remotely (Figure 7).

					And the local distance of the local distance		
e 			Test	Patan.			
Run Test		Due Text	Test F	listory			
lest ivame		Hun Test		Test ID	Config Name	Test Name	Time
Run Sequence				30	NEInp Temporal	Visible \NEInp Tempo	3/24/2012 1:47:09 PM
Sequence Name	NGATS DVT INFRARED	Run Sequence		31	NEInp Spatial	Visible\NEInp Spatial	3/24/2012 1:47:55 PM
ave Image				32	Uniformity	Visible\Uniformity	3/24/2012 1:49:54 PM
mage Number	Format BMP IDF			33	Auto-Alignment	Infrared\Auto-Alignm	3/24/2012 5:46:55 PM
ave Location		Save		34	FOV Square	Infrared\FOV Square	3/24/2012 5:49:15 PM
			•	35	MTF	Infrared\MTF	3/24/2012 5:49:40 PM
Server Status	0.0%			36	Boresight	Infrared\Boresight	3/24/2012 5:49:59 PM
Status	U: UK	Get Status		37	Boresight 2	Infrared\Boresight	3/24/2012 5:50:19 PM
onsole				38	MultiBoresight	MultiSpectral\MultiBo	3/24/2012 5:50:23 PM
tstatus		*		39	Manual MRTD	Infrared\Manual MR	3/24/2012 5:58:36 PM
tstatus				40	SITF	Infrared\SiTF	3/24/2012 6:03:33 PM
1-1-1 ···				41	NETD Tomporol	Informed/NETD Tomo	2/24/2012 C-05-0C PM
etstatus			Query	Database			
etstatus							
etstatus							
etstatus							
etstatus							
etstatus							
etstatus							
etstatus							
etstatus							
etstatus							
	TS DVT INFRARED	E					
nsequence NGA ommand Receive							

Figure 7: Remote Access Screenshot

5. DEVELOPMENT TIME COMPARISON

This section compares a TPS created on the IFTE V5 station under ATLAS control to a similar TPS created using IRWindows and executed via TestStand. There are 5 tests included in the TPS:

Boresight Boresight – Continuous Manual MRTD MTF MTF – Continuous

Below is a comparison of execution times on the NGATS station under IRWindows control compared to the IFTE V5 station under ATLAS control.

Test	Execution time on a NGATS station	Execution time on a IFTE V5
	under IRWindows control	station under ATLAS control
Boresight	Approx. 45-60 sec	Approx. 2-3 min
Manual MRTD	Approx. 2 min	Approx. 5-6 min
(with the caveat that it can take as		
long as you want it to)		
Modulation Transfer Function	instant	Approx. 2-3 min

Table 1: Test execution time comparisonbetween IRWindows4 and IFTE V5.

Note: Continuous Boresight and MTF aren't comparable since there is no correlating functionality on V5.

TPS Development Time comparison:

CROWS TIM TPS on IFTE V5 --> 9-12 months CROWS TIM TPS on NGATS/IRWindows --> 5 weeks

TPS Runtime comparison:

CROWS TIM TPS on V5 --> 45 min CROWS TIM TPS on NGATS/IRWindows --> 15 min (including additional testing)

6. SUMMARY/CONCLUSIONS

By taking advantage of updated test development models and an environment designed and developed to operate EO tests, the TPS developer working with IRWindows has many advantages:

- 1. Faster learning curve (graphical user interface is easier than test line interface)
- 2. Faster TPS development time (real time changes and asset control interface allows for faster development)
- 3. Asset control panel allows user to control assets real time and monitor all asset functions during development
- 4. Unit Under Test (UUT) image viewer allows user to set test parameters like Region of Interest more easily and more precisely
- 5. Continuous mode tests (like MTF allows user to real time adjustments)
- 6. Open architecture for test modifications

We were able to demonstrate these advantages in the case of the CROWS TIM TPS development. IRWindows 4 running on the NGATS EO interim solution test station shows significant savings in TPS development and execution time. Additionally, this application was able to take advantage of IRWindows remote interfacing so that NI TestStand could serve as the executive controlling the overall operation.

7. REFERENCES

[1] Irwin, A., LaVeigne, J. and Nehring, B., "A common architecture for TPS development", Proc. SPIE 8014, (2011)