Sepand X-ray Inspector (SXI)

Inspection Imaging Visualization Software

User Guide

Alireza Alipour

Maintained by Roholla Azimi Rad

Corresponding to SXI version v1.0.0, 2016/11/25

SGD
The Sepand Government Department

سپند

Contents

List of Figures

Preface

We introduce a non-intrusive inspection software, Sepand X-ray Inspector (SXI) that is a C/C++ graphic user interface (GUI) of the control and image visualization software and designed using OpenCV `http://www.opencv.org/` and GTK+ `http://www.gtk.org/` libraries. In this software we used a combinational approach to the image processing filters of raw x-ray image based carry-on personal luggage, long/short occupied vehicles scan images for improving the images, detection of threat objects on them and ensuring safety in high security areas.

This tutorial is an introduction to using Sepand X-ray Inspector (SXI). Here you will learn how to get started, how to use the interface, and how to modify images with image processing filters.

Alireza Alipour
Gilan, Iran
December 2016

About Sepand X-ray Inspector (SXI)

X-ray scanning systems for detection of threat objects in passenger luggage, cars and cargo containers in high security areas has a worldwide need and too much attention due to the increasing public concern for the purposes of national security, parrying terrorism activities and smuggling in present days. Based on some experimental results already utilized by the authors in [1] and re-considered as a benchmark, we present the application of a new non-intrusive vehicles and luggage inspection software based on image processing filters called Sepand X-ray Inspector (SXI) for the analysis of X-ray images generated from security equipment in order to detection of contraband and potential threat objects such as explosives material, weapons, drugs, or other illicit things to inspection, surveillance and ensuring national security at crowded and hazardous locations such as airport, railway station and custom entrances [2]. The manual inspection of large/small containers is not practical because of the time constrains and the high labor requirements for unpacking and repacking them content. Therefore, there is a need for non-intrusive scanning systems together with the use of dedicated software. Our motivation for designing SXI increases the rates of detecting concealed and low-density threat items in inspection systems as compared to raw images.

SXI is a software system encompassing three goals: 1. Its basic feature is acquire images rapidly and to use a range of sophisticated image processing tools to assess threats. 2. Increase operators' alertness, without decreasing inspection speed or customer satisfaction. 3. Minimize the role of human operators in monitoring and detection by automating or semi-automating the process of inspection at stations.

The SXI is designed using C/C++ code implemented in Microsoft© Visual studio ultimate 2013 compiler under Windows© 7. The entire software package and its associated library are written in C/C++. Optimized algorithms of image processing are obtained from OpenCV [1] [3] library. The graphical user interface is based on GTK+ [2] [4] library.

[1]Open Source Computer Vision
[2]GIMP Toolkit

Chapter 1 Compiling Main C/C++ Code

In this chapter you will learn what steps you need to perform in order to compile Main C/C++ code by using the OpenCV and GTK+ libraries inside Microsoft Visual Studio. Therefore, you should learn to install OpenCV and GTK+ libraries in your new project.

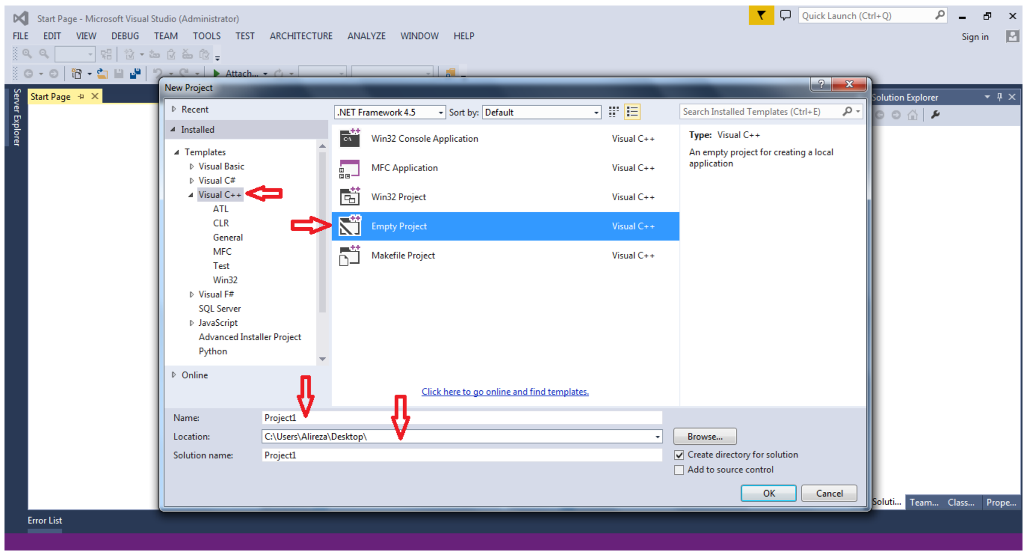
Installing the OpenCV Library

OpenCV [OpenCV] is an open source (see `http://opensource.org`) computer vision library available from `http://SourceForge.net/projects/opencvlibrary`. Th e library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Python, Ruby, Matlab, and other languages. OpenCV was designed for computational efficiency and with a strong focus on realtime applications. OpenCV is written in optimized C and can take advantage of multicore processors. If you desire further automatic optimization on Intel architectures [Intel], you can buy Intel's Integrated Performance Primitives (IPP) libraries [IPP], which consist of low-level optimized routines in many different algorithmic areas. OpenCV automatically uses the appropriate IPP library at runtime if that library is installed [5].

One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. The OpenCV library contains over 500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics. Because computer vision and machine learning often go hand-inhand, OpenCV also contains a full, general-purpose Machine Learning Library (MLL). This sublibrary is focused on statistical pattern recognition and clustering. The MLL is highly useful for the vision tasks that are at the core of OpenCV's mission, but it is general enough to be used for any machine learning problem. The main OpenCV site is on SourceForge at `http://SourceForge.net/projects/opencvlibrary` and the OpenCV Wiki [OpenCV Wiki] page is at `http://opencvlibrary.SourceForge.net` [5].

You will learn what steps you need to perform in order to use the OpenCV library inside a new Microsoft Visual Studio project. **Step 1:** download latest version OpenCV from `http://opencv.org/downloads.html` for windows and go create a new solution inside Visual studio by going through the File $\mapsto$ New $\mapsto$ Project menu selection. Choose *Empty Project* as type. Enter its name and select the path where to create it. Then in the upcoming dialog make sure you create an empty project (Fig. 1).

Step 2: Every project is built separately from the others. Due to this every project has its own rule package. Inside this rule packages are stored all the information the IDE

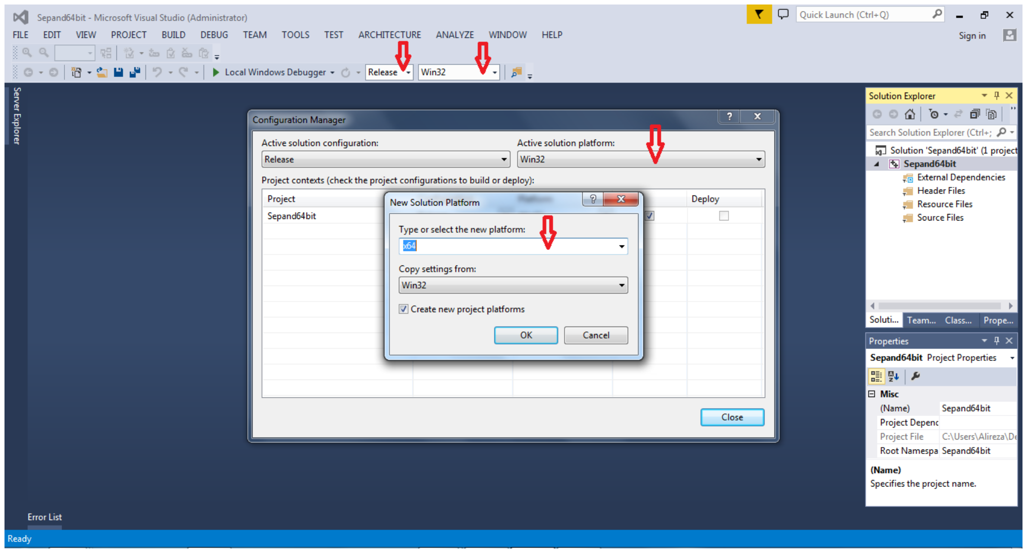

Start Page - Microsoft Visual Studio (Administrator)
FILE EDIT VIEW DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Quick Launch (Ctrl+Q)
Sign in
New Project
Recent
Installed
Templates
Visual Basic
Visual C#
Visual C++
ATL
CLR
General
MFC
Test
Win32
Visual F#
SQL Server
JavaScript
Advanced Installer Project
Python
Online
.NET Framework 4.5
Sort by: Default
Search Installed Templates (Ctrl+E)
Win32 Console Application Visual C++
MFC Application Visual C++
Win32 Project Visual C++
Empty Project Visual C++
Makefile Project Visual C++
Click here to go online and find templates.
Browse...
OK
Cancel
Solution Explorer
Error List

Type: Visual C++
An empty project for creating a local application

Name: Project1
Location: C:\Users\Alireza\Desktop\
Solution name: Project1
Create directory for solution ✓
Add to source control ☐

Figure 1: Create a new project inside Visual studio.

needs to know to build your project. For any application there are at least two build modes: a Release and a Debug one. The Debug has many features that exist so you can find and resolve easier bugs inside your application. In contrast the Release is an optimized version, where the goal is to make the application run as fast as possible or to be as small as possible. So we select *Release* mode and change platform to *x64* (Fig. 2).

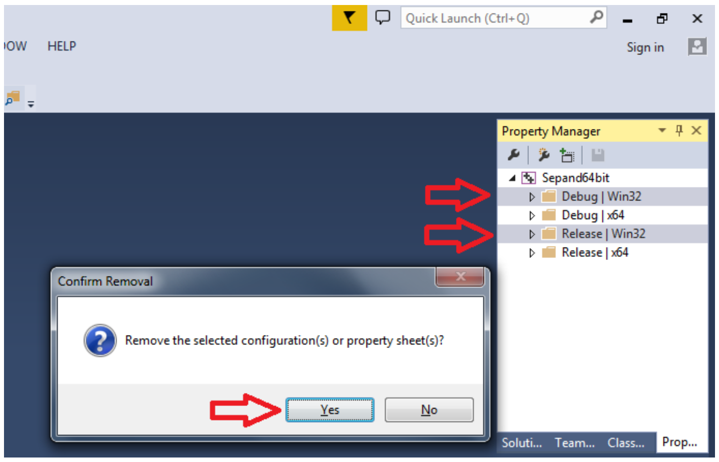

Sepand64bit - Microsoft Visual Studio (Administrator)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Quick Launch (Ctrl+Q)
Sign in
Local Windows Debugger
Release
Win32
Configuration Manager
Project contexts (check the project configurations to build or deploy):
Project
Deploy
Sepand64bit
New Solution Platform
OK
Cancel
Close
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Sepand64bit' (1 project)
Sepand64bit
External Dependencies
Header Files
Resource Files
Source Files
Properties
Sepand64bit Project Properties
Misc
Error List
Ready

Active solution configuration: Release
Active solution platform: Win32
Type or select the new platform: x64
Copy settings from: Win32
Create new project platforms ✓

(Name) Sepand64bit
Project Depenc
Project File C:\Users\Alireza\D
Root Namespa Sepand64bit
(Name)
Specifies the project name.

Figure 2: Select Release mode and x64 platform.

Step 3: Delete modes related to *Win32* platform in the Property Manger (Fig. 3).



Remove the selected configuration(s) or property sheet(s)?

Figure 3: Delete modes related to *Win32* platform.

Step 4: You may figure that these modes also require different rules to use during build. Therefore, there exist different rule packages for each of your build modes. These rule packages are called inside the IDE as project properties and you can view and modify them by using the Property Manger. You can bring up this with View ↦ Property Pages. Expand it and you can see the existing rule packages (called Proporty Sheets). The really useful stuff of these is that you may create a rule package once and you can later just add it to your new projects. Create it once and reuse it later. We want to create a new Property Sheet that will contain all the rules that the compiler and linker needs to know. Of course we will need a one just for the Release Build (Fig. 4).

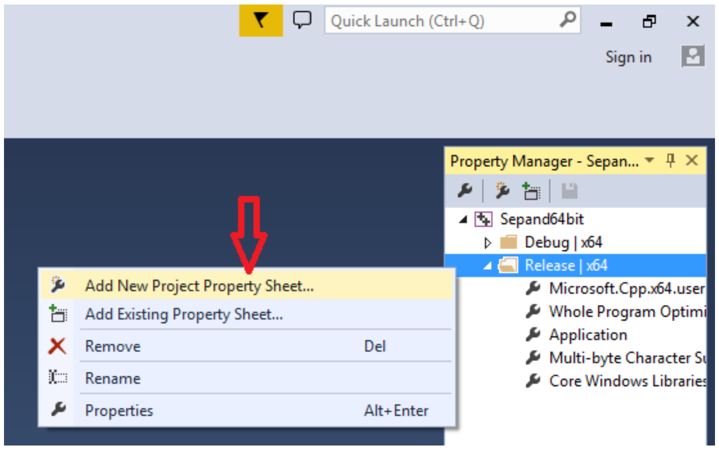
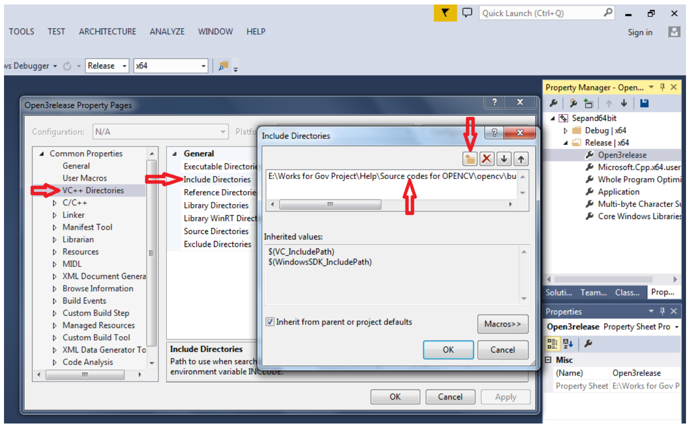


Figure 4: Create a new Property Sheet for the Release Build using OpenCV library.

Use for example the *Open3release* name. Then by selecting the sheet Right Click ↦ Properties. In the following I will show to set the OpenCV rules locally, as I find unnecessary to pollute projects with custom rules that I do not use it. Go the *Command Properties* groups *VC++ Directories* entry and under the "Include Directories" add the path to your OpenCV include. For example, `E:\....\your opencv file\build\include` (Fig. 5).



Include Directories: E:\Works for Gov Project\Help\Source codes for OPENCV\opencv\bu

Inherited values:
$(VC_IncludePath)
$(WindowsSDK_IncludePath)

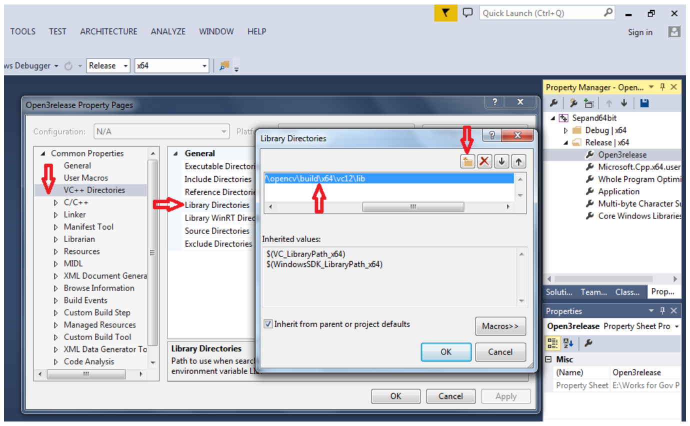
☑ Inherit from parent or project defaults

Include Directories
Path to use when search
environment variable INCLUDE.

(Name) Open3release
Property Sheet E:\Works for Gov P

Figure 5: Add the path of include file in OpenCV folder to Include Directories.

Under the "Library Directories" add the path to your OpenCV include. For example, `E:\....\your opencv file\build\x64\vc12\lib` (Fig. 6).



Library Directories: \opencv\build\x64\vc12\lib

Inherited values:
$(VC_LibraryPath_x64)
$(WindowsSDK_LibraryPath_x64)

☑ Inherit from parent or project defaults

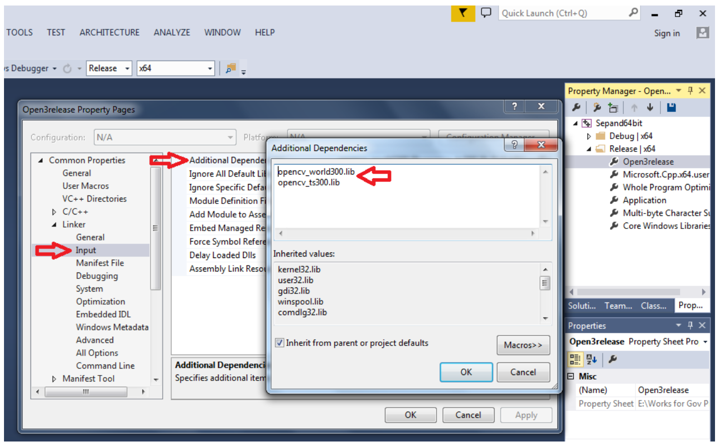
Library Directories
Path to use when searc
environment variable LI

(Name) Open3release
Property Sheet E:\Works for Gov P

Figure 6: Add the path of lib folder in OpenCV folder to Library Directories.

Then you need to specify the libraries in which the linker should look into. To do this go to the Linker ↦ Input and under the "Additional Dependencies" entry add the name of all modules which you want to use. For example, go to `E:\....\your opencv file\build\x64\vc12\bin`. For the latest version would contain:

opencv_world300.lib and *opencv_ts300.lib* (Fig. 7).



```
opencv_world300.lib
opencv_ts300.lib
```

Inherited values:

```
kernel32.lib
user32.lib
gdi32.lib
winspool.lib
comdlg32.lib
```

Figure 7: Copy name *.lib files in OpenCV folder to Additional Dependencies.

You can find your property sheets inside your projects directory. At this point it is a wise decision to back them up into some special directory, to always have them at hand in the future, whenever you create an OpenCV project. Note that for Visual Studio 2010 and upper versions of it the file extension is *.props, while for 2008 and lower versions of it this is *.vsprops. Next time when you make a new OpenCV project just use the ”Add Existing Property Sheet...” menu entry inside the Property Manager to easily add the OpenCV build rules (Fig. 8).

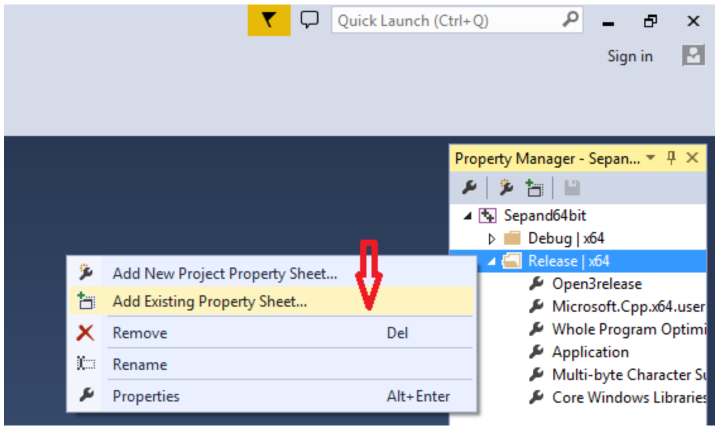


Figure 8: Use the Property Sheet for new OpenCV projects.

Step 5: Finally, you should learn to insert OpenCV library in your PC. First Right Click ↦ My Computer and push Properties. Click ↦ Advanced system setting and next Click ↦ Environment Variable (Fig. 9).

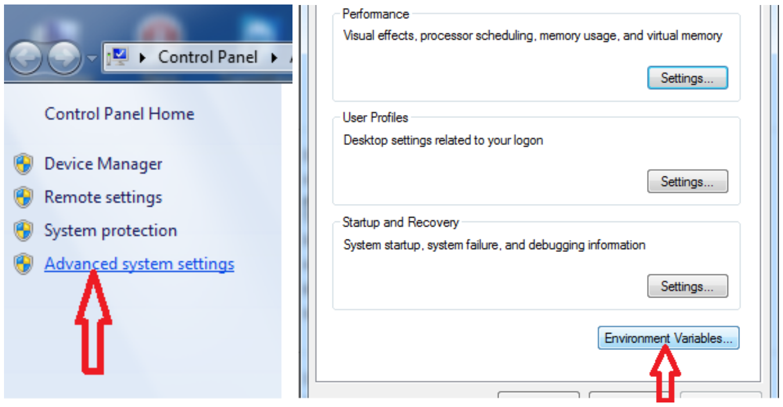
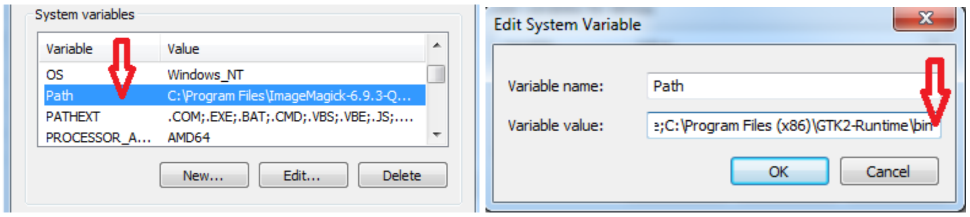


Figure 9: Going to Environment Variable.

In system variable double Click ↦ path ↦ Edit and next to final line and insert a semi-colon [;]. Finally, copy for example this path: E:\....\your opencv file\build\x64\vc12\bin and push OK (Fig. 10).


System variables

Variable	Value
OS	Windows_NT
Path	C:\Program Files\ImageMagick-6.9.3-Q...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64

Edit System Variable

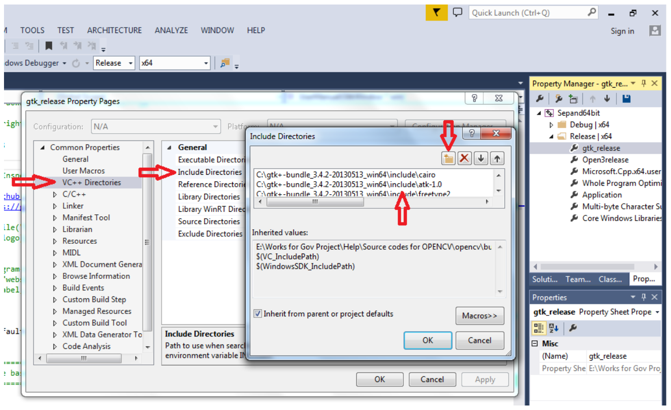
Variable name: Path

Variable value: ;C:\Program Files (x86)\GTK2-Runtime\bin

Figure 10: Add bin folder in OpenCV folder to system variable.

Installing the GTK Library

GTK+, or the GIMP Toolkit, is a multi-platform toolkit for creating graphical user interfaces. Offering a complete set of widgets, GTK+ is suitable for projects ranging from small one-off tools to complete application suites. GTK+ is cross-platform and boasts an easy to use API, speeding up your development time. Take a look at the screenshots to see a number of platforms GTK+ will run. GTK+ is written in C but has been designed from the ground up to support a wide range of languages, not only C/C++ [4]. GTK+ is free software and part of the GNU Project. However, the licensing terms for GTK+, the GNU LGPL, allow it to be used by all developers, including those developing proprietary software, without any license fees or royalties. The main GTK+ site is on The GTK+ Project at `https://www.gtk.org/` [4]. You will learn what steps you need to perform in order to use the GTK+ library inside a new Microsoft Visual Studio project. **Step 1:** download latest version GTK from `http://www.tarnyko.net/dl/gtk.htm` for windows. **Step 2:** We want to create a new Property Sheet that will contain all the rules that the compiler and linker needs to know. Of course we will need a one just for the Release Build (similar to Fig. 4). Use for example the *gtk_release* name. Then by selecting the sheet Right Click ↦ Properties. In the following I will show to set the GTK+ rules locally, as I find unnecessary to pollute projects with custom rules that I do not use it. Go the *Command Properties* groups *VC++ Directories* entry and under the "Include Directories" add the path to your GTK+ include. For example, `E:\ ...\ your gtk+ file\lib\glib-2.0\include`, `E:\ ...\ your gtk+ file\include`, `E:\ ...\ your gtk+ file\include\`**`cairo`**, `E:\ ...\ your gtk+ file\include\`**`atk-1.0`**, ... and include below files similar to **cairo** and **atk-1.0** files: **glib-2.0,freetype2, gail-3.0, fontconfig, gdk-pixbuf-2.0, gio-win32-2.0, jasper, librsvg-2.0, gtk-3.0, libcroco-0.6, libxml2, libpng15, pango-1.0, pixman-2, lzma** (Fig. 11).



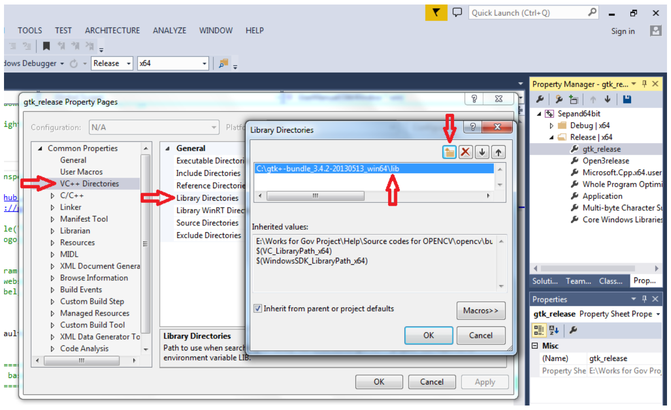
```
Include Directories
C:\gtk+-bundle_3.4.2-20130513_win64\include\cairo
C:\gtk+-bundle_3.4.2-20130513_win64\include\atk-1.0
C:\gtk+-bundle_3.4.2-20130513_win64\include\freetype2

Inherited values:
E:\Works for Gov Project\Help\Source codes for OPENCV\opencv\bu
$(VC_IncludePath)
$(WindowsSDK_IncludePath)

[✓] Inherit from parent or project defaults
```

Figure 11: Add the path of include files in GTK+ folder to Include Directories.

Under the "Library Directories" add the path to your GTK+ include. For example, `E:\ ...\ your gtk+ file\lib` (Fig. 12).



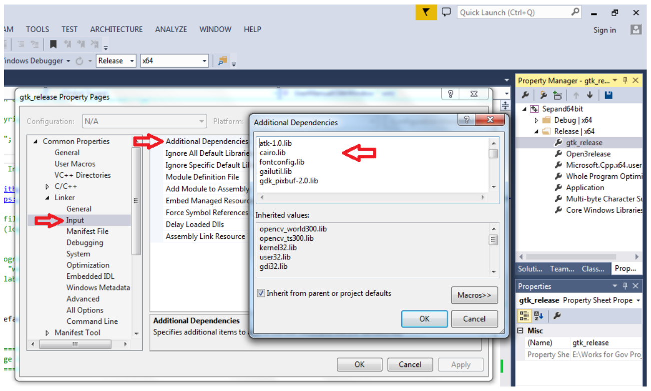
```
Library Directories
C:\gtk+-bundle_3.4.2-20130513_win64\lib

Inherited values:
E:\Works for Gov Project\Help\Source codes for OPENCV\opencv\bu
$(VC_LibraryPath_x64)
$(WindowsSDK_LibraryPath_x64)

[✓] Inherit from parent or project defaults
```

Figure 12: Add the path of lib folder in GTK+ folder to Library Directories.

Then you need to specify the libraries in which the linker should look into. To do this go to the Linker ↦ Input and under the "Additional Dependencies" entry add the name of all modules which you want to use. For example, go to `E:\ ...\ your gtk+ file\lib`. For the latest version would contain: *atk-1.0.lib*, *cairo.lib*, *fontconfig.lib*, ... (Fig. 13).



Additional Dependencies

Specifies additional items to

Additional Dependencies

atk-1.0.lib
cairo.lib
fontconfig.lib
gailutil.lib
gdk_pixbuf-2.0.lib

Inherited values:

opencv_world300.lib
opencv_ts300.lib
kernel32.lib
user32.lib
gdi32.lib

☑ Inherit from parent or project defaults

(Name)	gtk_release
Property She	E:\Works for Gov Pro

Figure 13: Copy name *.lib files in GTK+ folder to Additional Dependencies.

You can find your property sheets inside your projects directory. At this point it is a wise decision to back them up into some special directory, to always have them at hand in the future, whenever you create an GTK+ project. Note that for Visual Studio 2010 and upper versions of it the file extension is *.props, while for 2008 and lower versions of it this is *.vsprops. Next time when you make a new GTK+ project just use the "Add Existing Property Sheet..." menu entry inside the Property Manager to easily add the OpenCV build rules (Fig. 14).

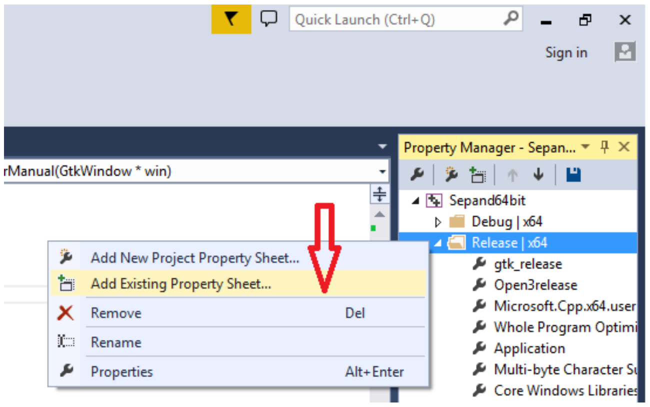
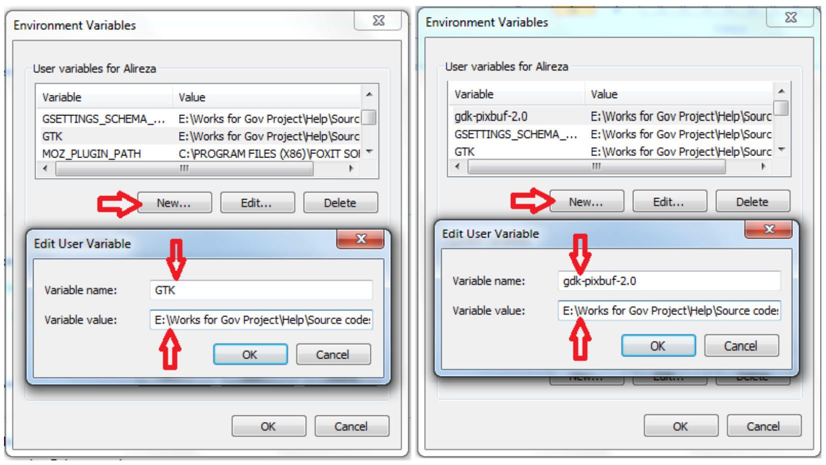


Figure 14: Use the Property Sheet for new GTK+ projects.

Step 3: Finally, you should learn to insert GTK+ library and other files in your PC. First Right Click ↦ My Computer and push Properties. Click ↦ Advanced system setting and next Click ↦ Environment Variable (Fig. 9). In User variable Click ↦ New and copy *GTK* on *variable name*. Next copy path of your GTK+ file on *variable value*. For example: `E:\......\your gtk folder`. Similary, do it for *gdk-pixbuf-2.0* file and copy path of it. For example: `E:\......\your gtk folder\include\gdk-pixbuf-2.0` (Fig. 15).


Environment Variables

User variables for Alireza

Variable	Value
GSETTINGS_SCHEMA_...	E:\Works for Gov Project\Help\Sourc
GTK	E:\Works for Gov Project\Help\Sourc
MOZ_PLUGIN_PATH	C:\PROGRAM FILES (X86)\FOXIT SOI

Edit User Variable

Variable name: GTK

Variable value: E:\Works for Gov Project\Help\Source code:

Environment Variables

User variables for Alireza

Variable	Value
gdk-pixbuf-2.0	E:\Works for Gov Project\Help\Sourc
GSETTINGS_SCHEMA_...	E:\Works for Gov Project\Help\Sourc
GTK	E:\Works for Gov Project\Help\Sourc

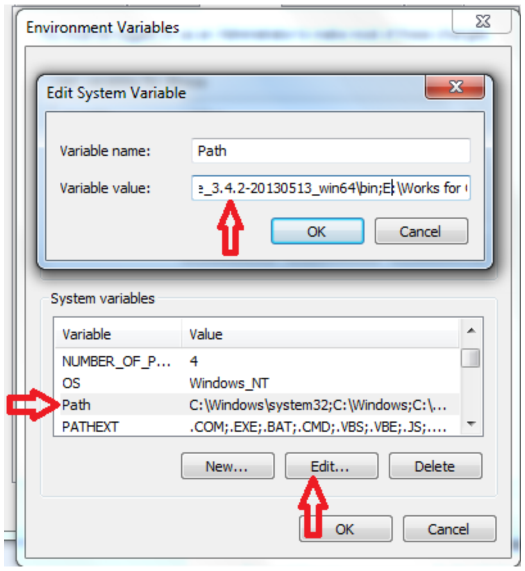
Edit User Variable

Variable name: gdk-pixbuf-2.0

Variable value: E:\Works for Gov Project\Help\Source code:

Figure 15: Insert your GTK+ and gdk-pixbuf-2.0 files to User variable.

In system variable double Click ↦ path ↦ Edit and insert a semicolon [;] in final line. Copy for example this path: `E:\....\your GTK+ file\bin` and push OK (Fig. 16).


Environment Variables

Edit System Variable

Variable name: Path

Variable value: ɛ_3.4.2-20130513_win64\bin;E:\Works for

System variables

Variable	Value
NUMBER_OF_P...	4
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

Figure 16: Add bin folder in GTK+ folder to system variable.

When OpenCV and GTK+ libraries install on your project, you can insert your main c/c++ code and compile it (Fig. 17).

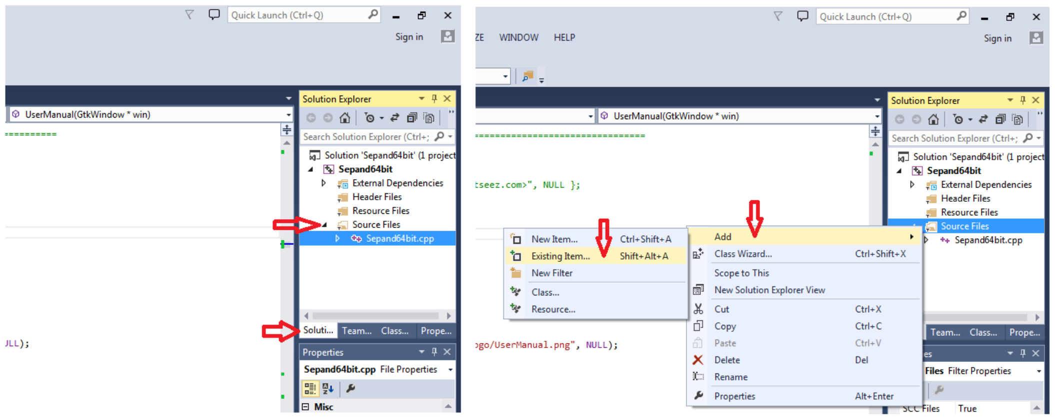

```
JLL);
```
```
tseez.com>", NULL };
go/UserManual.png", NULL);
```

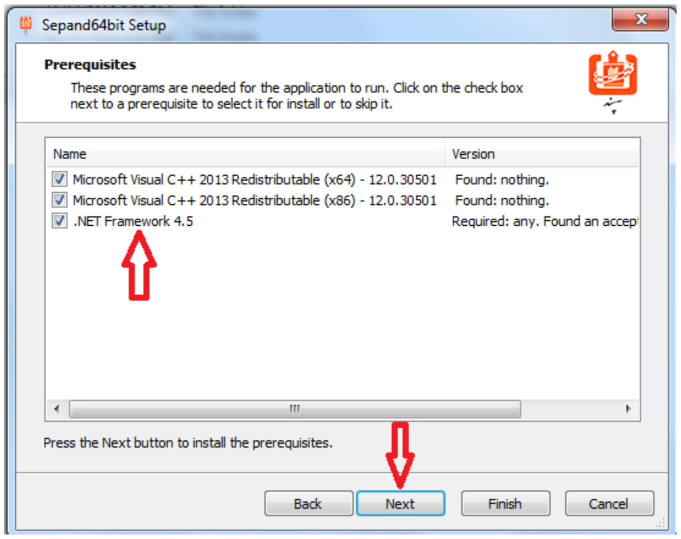
Figure 17: Compile main c/c++ code.

Chapter 2 Installing SXI software on PC

In this chapter you will learn what steps you need to perform in order to install Sepand X-ray Inspector (SXI) software on your PC. Therefore, you should learn to install requirements and dependencies in your PC.

Installation requirements and dependencies

Sepand X-ray Inspector (SXI) software design for Windows 32 bit and 64 bit. For installing SXI, double click on *Sepand32bit.exe* or *Sepand64bit.exe* files. SXI require and depend on some files: (1) *Microsoft .NET Framework 4.5* and (2) *Microsoft Visual C++ 2013 redistributable (x64)/(x86) 12.0.30501*. When your PC connect to internet, push Next to requirements and dependencies files install on your PC (Fig. 18).



Prerequisites

These programs are needed for the application to run. Click on the check box next to a prerequisite to select it for install or to skip it.

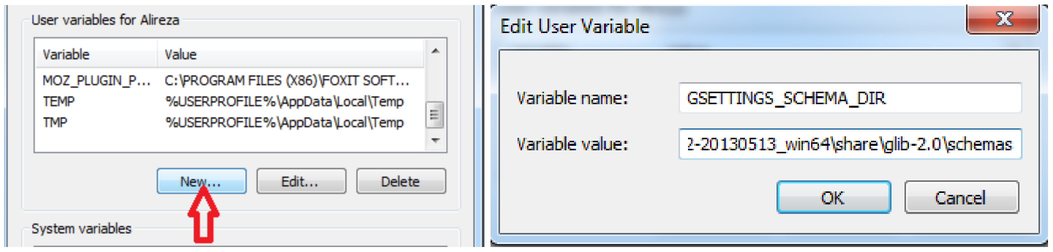
Name	Version
☑ Microsoft Visual C++ 2013 Redistributable (x64) - 12.0.30501	Found: nothing.
☑ Microsoft Visual C++ 2013 Redistributable (x86) - 12.0.30501	Found: nothing.
☑ .NET Framework 4.5	Required: any. Found an accep

Press the Next button to install the prerequisites.

Figure 18: Installation requirements and dependencies on PC.

When requirements and dependencies files installed, push Next buttons to complete Installation. Note, you don't change path of Installation. Finally, Right Click $\mapsto$ My Computer and push Properties. Click $\mapsto$ Advanced system setting and next Click $\mapsto$ Environment Variable (Fig. 9).

In User variable Click ↦ New and copy *GSETTINGS_SCHEMA_DIR* on *variable name*. Next copy below path: C:\Program Files\Sepand32bit\share\glib-2.1\schemas for Win 32bit and C:\Program Files (x86)\Sepand32bit\share\glib-2.1\schemas for Win64bit on *variable value* and push OK (Fig. 10).


User variables for Alireza

Variable	Value
MOZ_PLUGIN_P...	C:\PROGRAM FILES (X86)\FOXIT SOFT...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

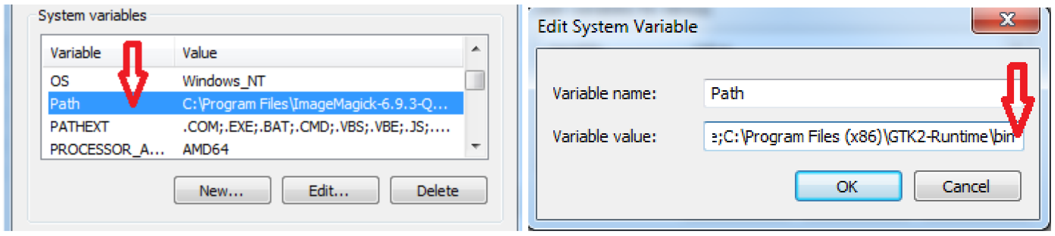
Edit User Variable

Variable name: GSETTINGS_SCHEMA_DIR

Variable value: 2-20130513_win64\share\glib-2.0\schemas

Figure 19: Insert schemas file to User variable.

In system variable double Click ↦ path and next to final line and insert a semicolon [;]. Copy path: C:\Program Files\Sepand32bit\share\glib-2.1\schemas for Win 32bit and C:\Program Files (x86)\Sepand32bit\share\glib-2.1\schemas for Win64bit on *variable value* (Fig. 20).


System variables

Variable	Value
OS	Windows_NT
Path	C:\Program Files\ImageMagick-6.9.3-Q...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64

Edit System Variable

Variable name: Path

Variable value: ≥;C:\Program Files (x86)\GTK2-Runtime\bin

Figure 20: Insert schemas file to system variable.

Chapter 3 Getting Started with SXI

In this chapter you will learn how to get started, how to use the interface, and how to modify images with image processing filters.

Opening and Log In

On a PC, click on Sepand icon on the desktop. Next, Log In page open (Fig. 21). In *Readme.txt* file, Username and Password defined and you can use of them to Log In page.

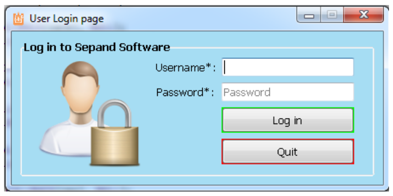


Figure 21: Insert Username and Password in Log In page.

MAIN WINDOW

When Log In page, layout of SXI interface open (Fig. 22).

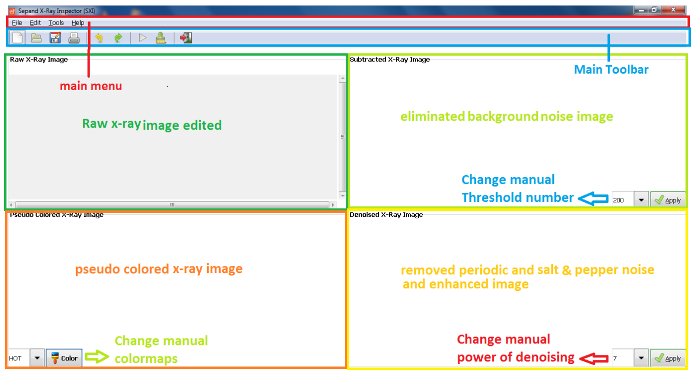


Figure 22: Layout of SXI interface.

SXI's main window consists of three parts. The main menu, main toolbar and four packet image panes that display: Raw x-ray image edited (up-left), eliminated background noise image (up-right), removed periodic and salt&pepper noise and enhanced image (down-right) and the pseudo colored x-ray image (down-left).

MAIN MENU

SXI's main menu contains four modules: File, Edit, Tools and Help (Fig. 23).

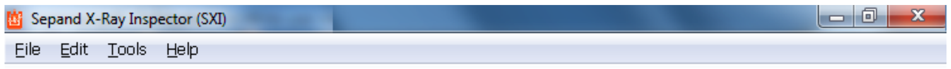


Figure 23: The Main menu of the SXI.

File menu contains items to open, save, print images and to quit the SXI application. Edit menu contains items to resize, crop, rotate and to undo/redo images. Tools menu contains items to play (compile code) and clear panes. Help menu contains items to help the user (Fig. 24).

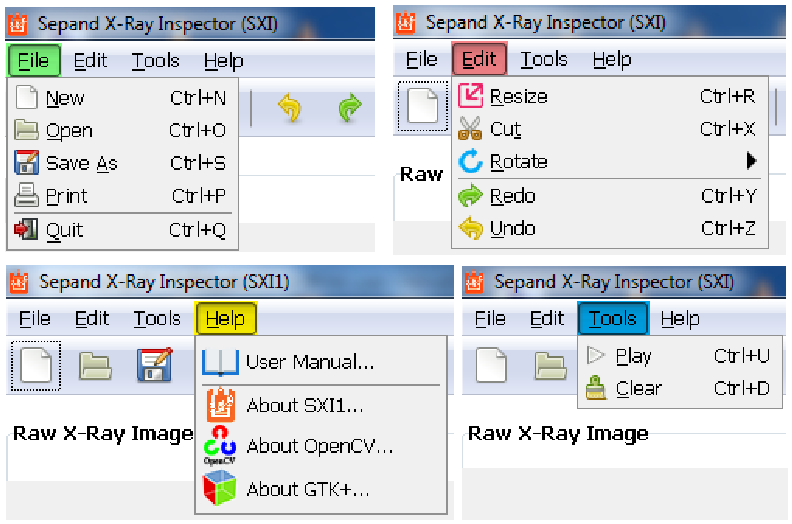


Figure 24: Items of main menu.

SXI allows users to read a raw x-ray image and edit it with different items. Push the Open button in File menu and read an image. This image rotate anti-clockwise (-90). Next, cut it with pressing mouse on image and stretch across it the place where you want and finally push the Cut button in Edit menu. If size image was small/big, you can increase/decrease size it with Resize button (Fig. 25).

When an image edited is ready user can play software and see result on three panes. If background noise doesn't eliminate goodness, a button insert on pane that user can change threshold number to eliminate background noise goodness. Also, if noises doesn't remove

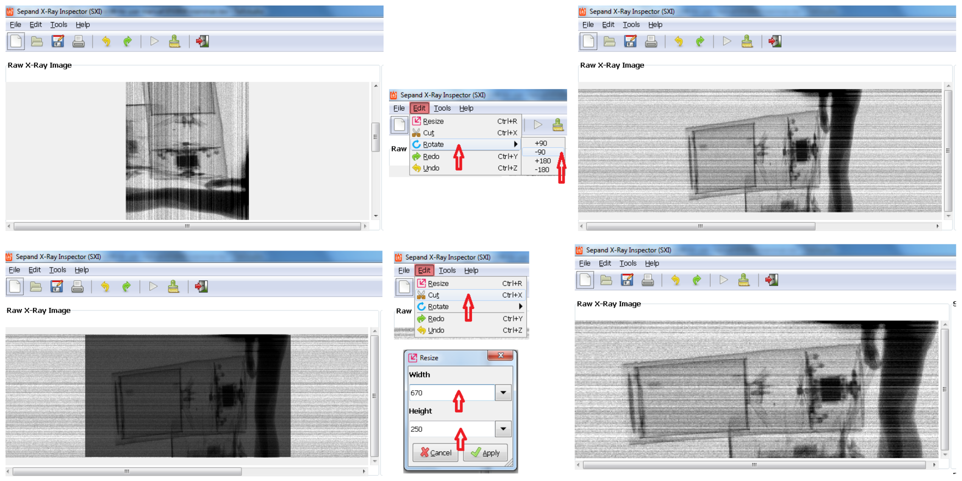


Figure 25: An example of editing raw x-ray image.

goodness, a button insert on pane that user can change power of removal noises. Finally, a button put in pane for change color scale. Figure 26 show an example of compiling SXI.

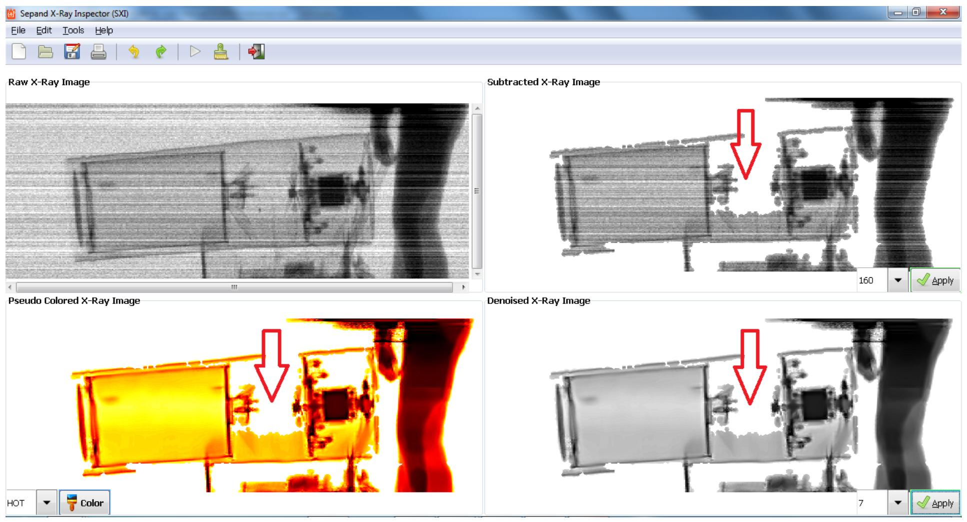


Figure 26: An example of compiling SXI.

Results of eliminating background noise is not goodness. So, by using a button you can change threshold number, for example for this image change it from 160 to 185 and push Apply. Next, push button for removal noise. Finally, you can change type of color coding for example, from Hot to Jet (Fig. 27).

You can push Save button in File menu and save as results in your PC. First, select place for saving, for example *Desktop*. Next, push create folder and define a name for folder and

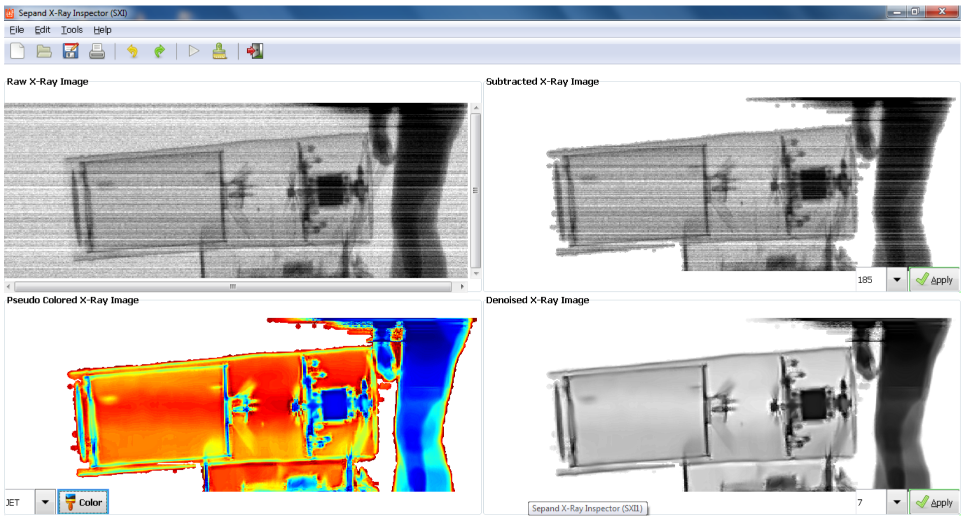
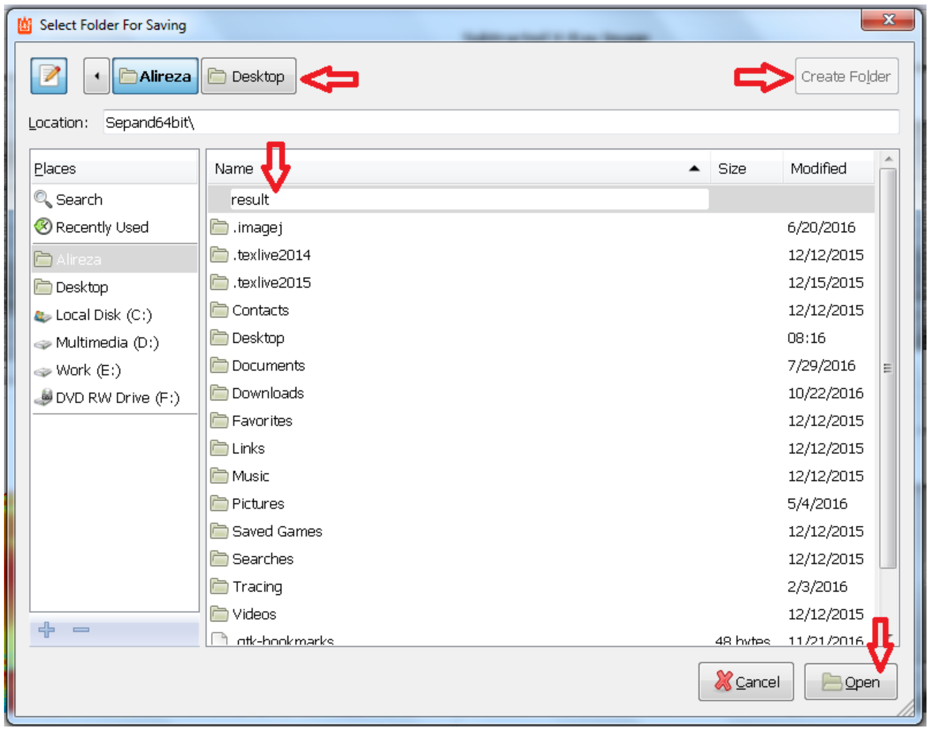


Figure 27: An example of re-compiling SXI.

push *inter key* on your keyboard. finally push Open button (Fig. 28).



Location: Sepand64bit\

Name	Size	Modified
result		
.imagej		6/20/2016
.texlive2014		12/12/2015
.texlive2015		12/15/2015
Contacts		12/12/2015
Desktop		08:16
Documents		7/29/2016
Downloads		10/22/2016
Favorites		12/12/2015
Links		12/12/2015
Music		12/12/2015
Pictures		5/4/2016
Saved Games		12/12/2015
Searches		12/12/2015
Tracing		2/3/2016
Videos		12/12/2015
.gtk-bookmarks	48 bytes	11/21/2016

Figure 28: Save as results in PC.

You can push Print button in File menu and Print results in your PC. First, select an image for printing, for example *Raw X-Ray Image* and push OK button (Fig. 29)

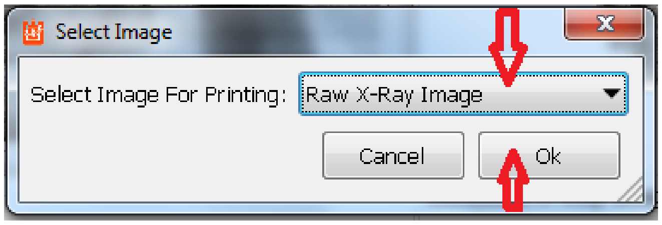


Figure 29: Print results in PC.

When you save and print results in your PC, you can clear panes and start a new image for improving and other work. For doing this work, push Clear button in Edit menu.

MAIN TOOLBAR

Most of the major tools are located in the Tool bar for easy access (Fig. 30).

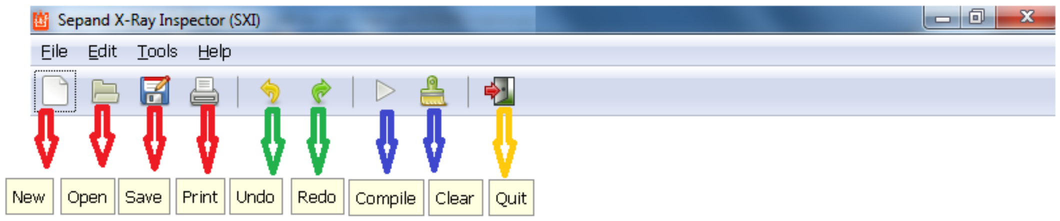


Figure 30: Main toolbar of SXI.

Bibliography

[1] Alipour Alireza and Roholla Azimi Rad. *Improving dual-energy x-ray images for better threat items detection*. Digital Signal Processing , 2016.

[2] Alipour Alireza and Roholla Azimi Rad. *Sepand X-ray Inspector (SXI) v1.0.0: A Non-intrusive Imaging Visualization Software for inspection and surveillance applications*. SotwareX, 2016.

[3] G. Bradski *The OpenCV Library*. Dr. Dobb's Journal of Software Tools, 2000. (Available from `http://www.opencv.org/`)

[4] *The GTK+ project (formerly GIMP Toolkit, sometimes incorrectly termed the GNOME Toolkit)*. (Available from `http://www.gtk.org/`)

[5] G. R. Bradski *Learning OpenCV: Computer Vision with the OpenCV Library*. Shroff Publishers & Distributors. 2008. `https://books.google.com/books?id=Um2zoQEACAAJ`