http://gmv.cast.uark.edu A Method Store for Advanced Survey and Modeling Technologies Mon, 01 Apr 2013 03:29:18 +0000 en-US hourly 1 http://wordpress.org/?v=3.5.1 http://gmv.cast.uark.edu/photogrammetry/software-photogrammetry/photoscan/photoscan-workflow/photoscan-basic-processing-for-photogrammetry/ http://gmv.cast.uark.edu/photogrammetry/software-photogrammetry/photoscan/photoscan-workflow/photoscan-basic-processing-for-photogrammetry/#comments Tue, 19 Mar 2013 13:35:57 +0000 caitlin http://gmv.cast.uark.edu/?p=13131 Continue reading →]]>

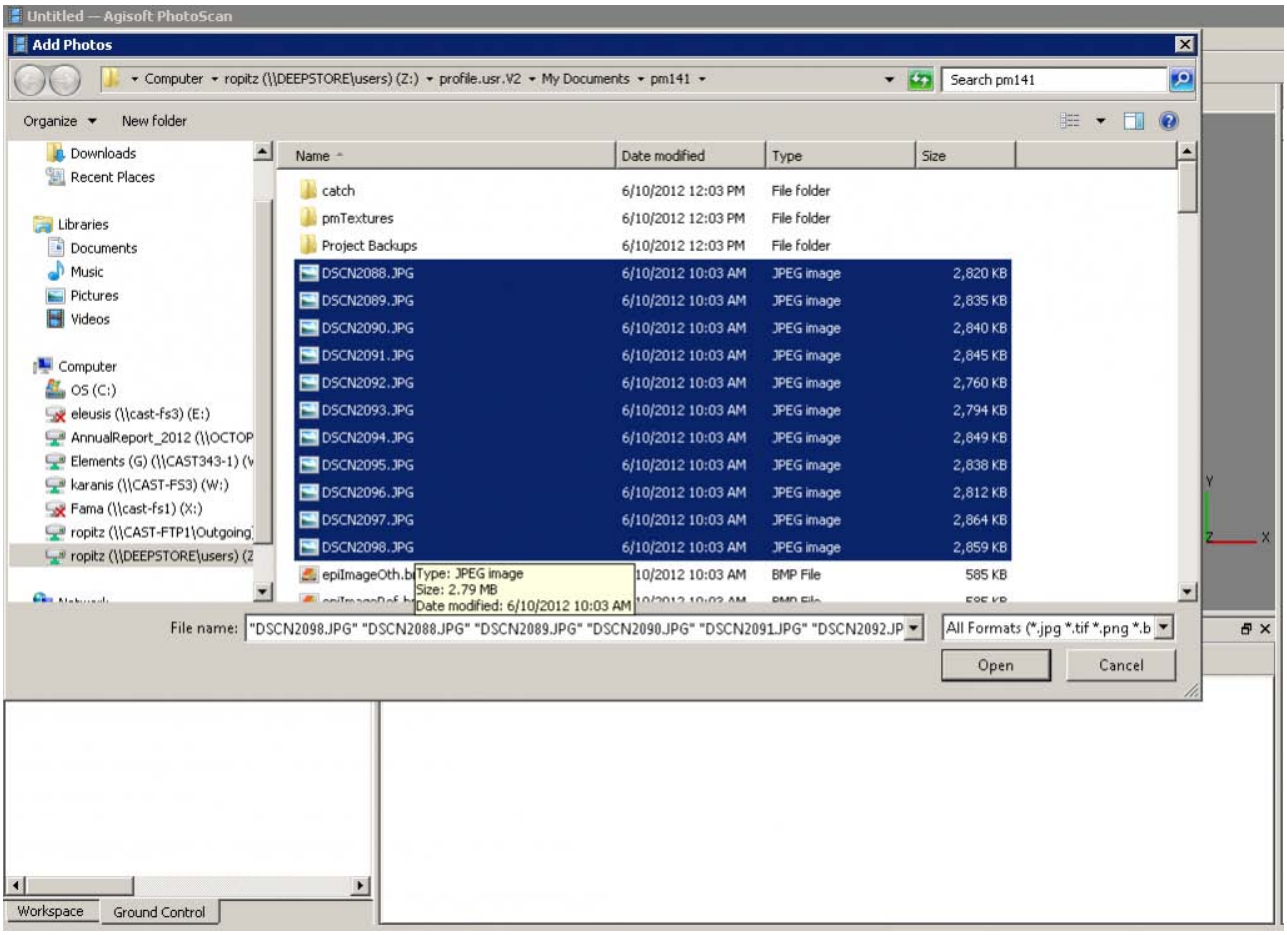
This series will show you how to create 3d models from photographs using Agisoft Photoscan and Esri ArcGIS.

Hint: You can click on any image to see a larger version.

Many archaeological projects now use photogrammetric modeling to record stratigraphic units and other features during the course of excavation. In another post we discussed bringing photogrammetric or laserscanning derived models into a GIS in situations where you don't have precise georeferencing information for the model. In this post we will demonstrate how to use bring a photogrammetric model for which georeferenced coordinates are available, using Agisoft's Photoscan Pro and ArcGIS.

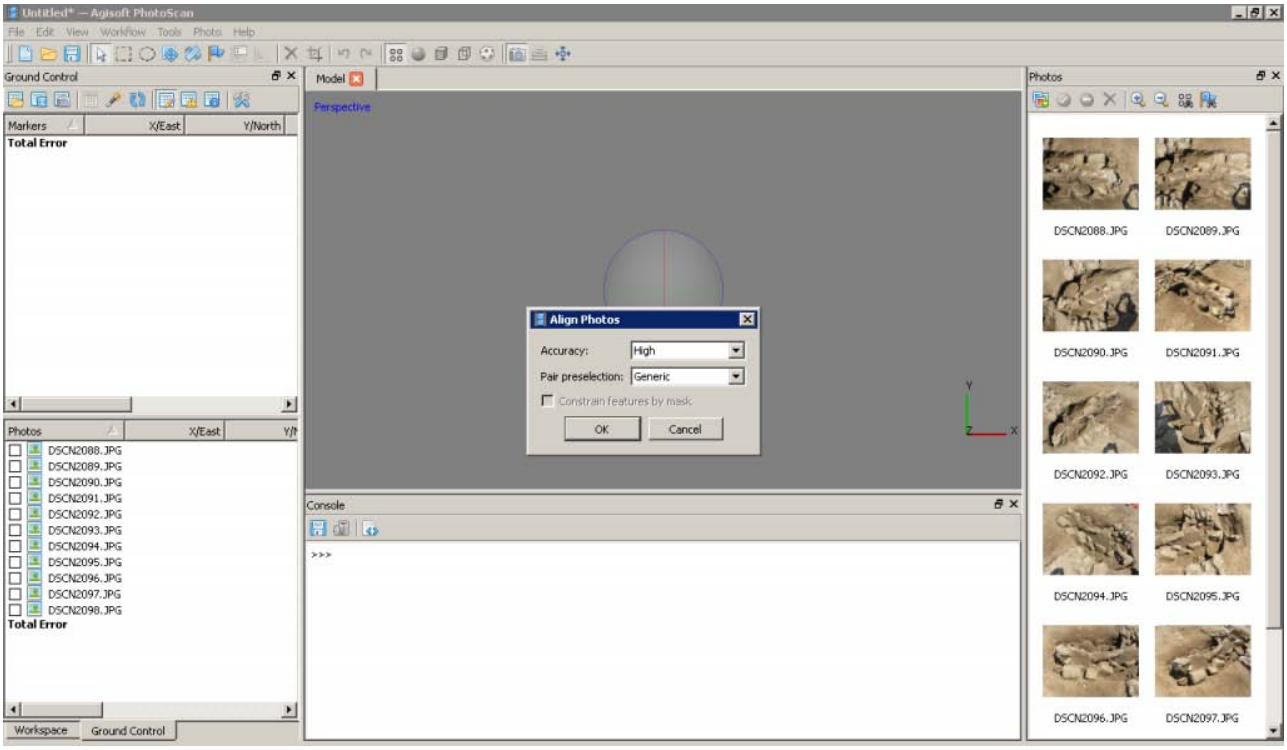
Load Photos

Begin by adding the photos used to create the model to an empty project in Photoscan.



Align Photos

Following the Photoscan Workflow, next align the images. From the menu at the top choose 'Workflow'>'Align Images'. A popup box will appear where you can input the alignment parameters. We recommend selecting 'High' for the accuracy and 'Generic' for the pair pre-selection for most convergent photogrammetry projects.

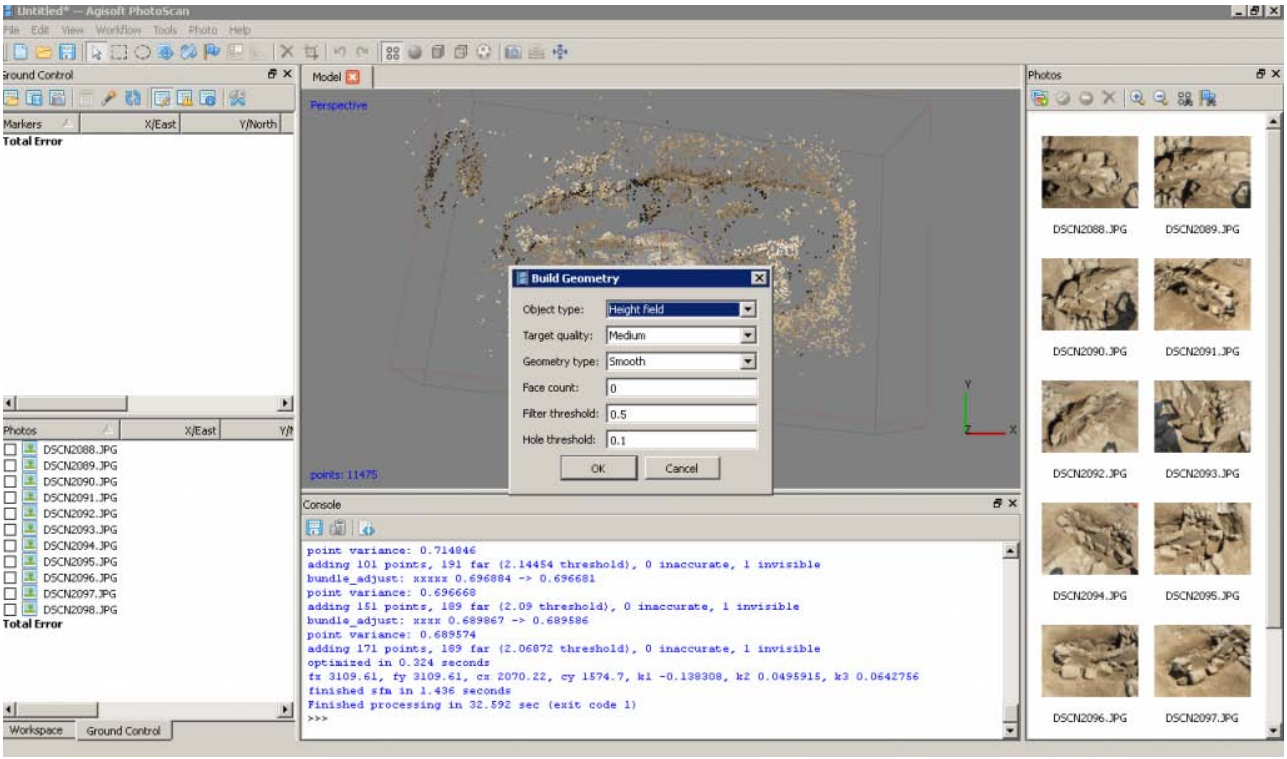


A choice

At this point there are two approaches to adding the georeferenced points to the project. You can place the points directly on each image and then perform the bundle adjustment, or you can build geometry and then place the points on the 3d model, which will automatically place points on each image, after which you can adjust their positions. We normally follow the second approach, especially for projects where there are a large number of photos.

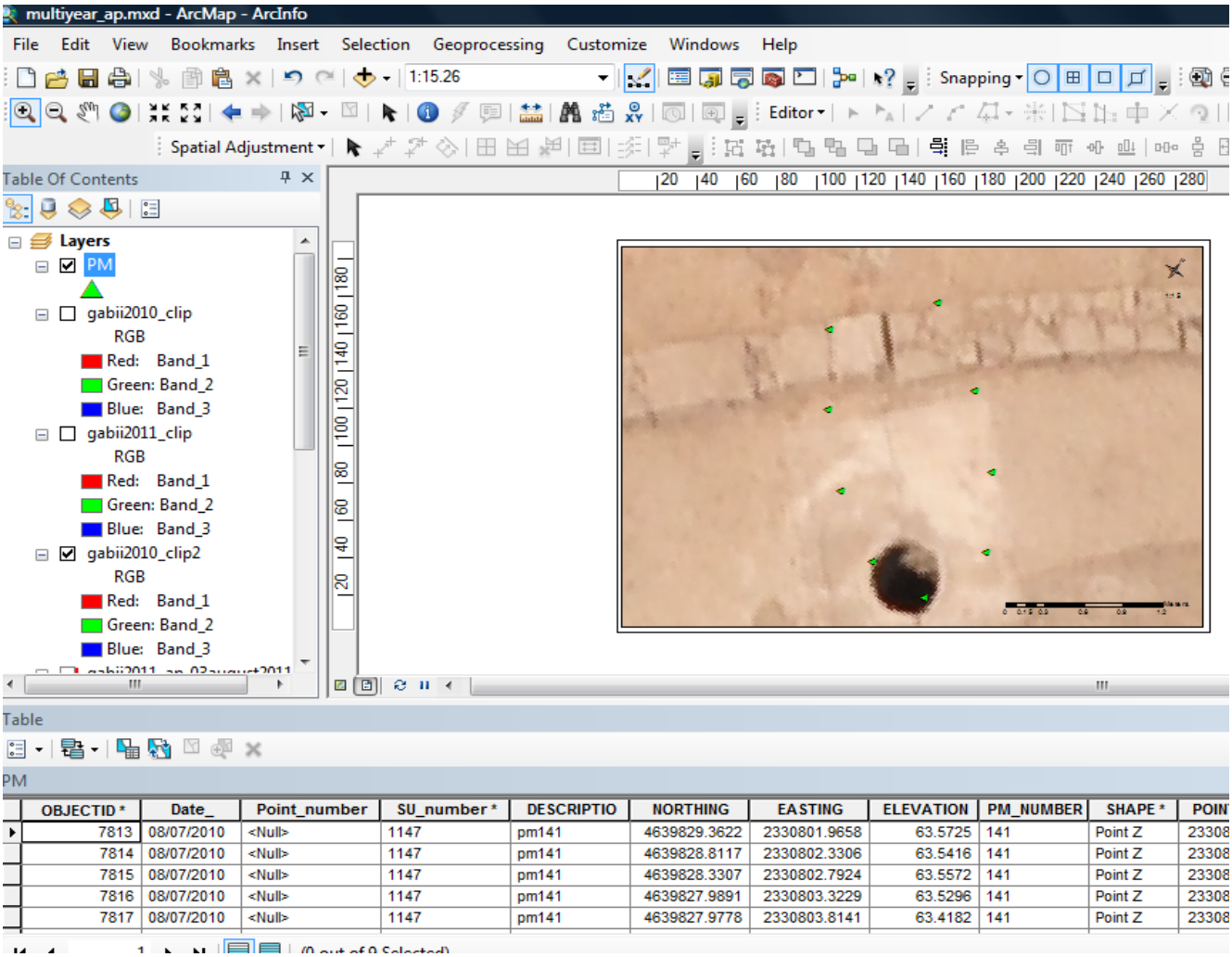
Build Geometry

Under ‘Workflow’ in the main menu, select ‘Build Geometry’. At this point we don’t need to build an uber-high resolution model, because this version of the model is just going to be used to place the markers for the georeferenced points. A higher resolution model can be built later in the process if desired. Therefore either ‘Low’ or ‘Medium’ are good choices for the model resolution, and all other parameters may be left as the defaults. Here we have selected ‘Medium’ as the resolution.



Get the georeferenced points

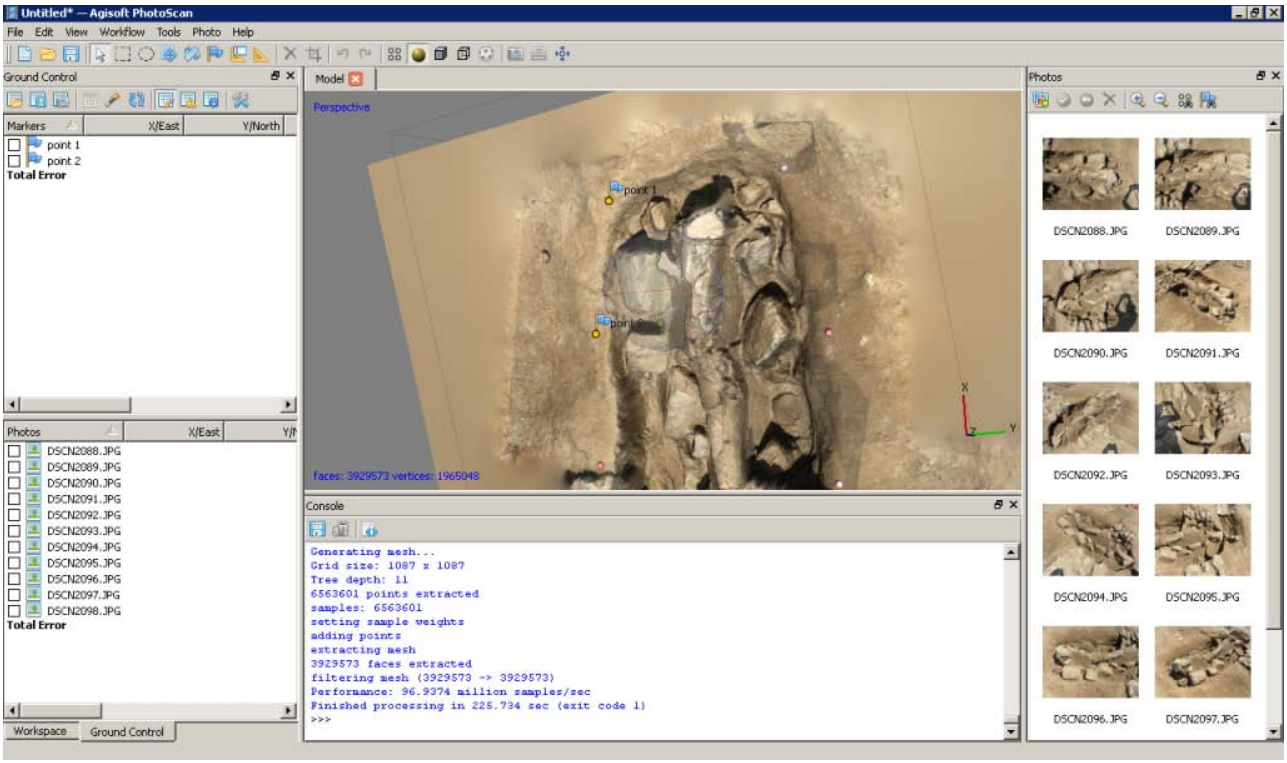
When the photos for this model were taken, targets were places around the feature (highly technical coca-cola bottle caps!) and surveyed using a total station. These surveyed targets are used to georeference the entire model. In this project all surveyed and georeferenced points are stored in an ArcGIS geodatabase. The points for this model are selected using a definition query and exported from ArcGIS.



OBJECTID *	Date_	Point_number	SU_number *	DESCRIPTIO	NORTHING	EASTING	ELEVATION	PM_NUMBER	SHAPE *	POIN
7813	08/07/2010	<Null>	1147	pm141	4639829.3622	2330801.9658	63.5725	141	Point Z	23308
7814	08/07/2010	<Null>	1147	pm141	4639828.8117	2330802.3306	63.5416	141	Point Z	23308
7815	08/07/2010	<Null>	1147	pm141	4639828.3307	2330802.7924	63.5572	141	Point Z	23308
7816	08/07/2010	<Null>	1147	pm141	4639827.9891	2330803.3229	63.5296	141	Point Z	23308
7817	08/07/2010	<Null>	1147	pm141	4639827.9778	2330803.8141	63.4182	141	Point Z	23308

Add the georeferenced points

On the left you have two tabbed menus, ‘Workspace’ and ‘Ground Control’. Switch to the the ‘Ground Control’ menu. Using the ‘Place Markers’ tool from the top menu, place a point on each surveyed target. Enter the corresponding coordinates from the surveyed points through the ‘Ground Control’ menu. *Be careful to check that the northing, easting and height fields map correctly when importing points into Photoscan, as they may be in a different order than in ArcGIS.*

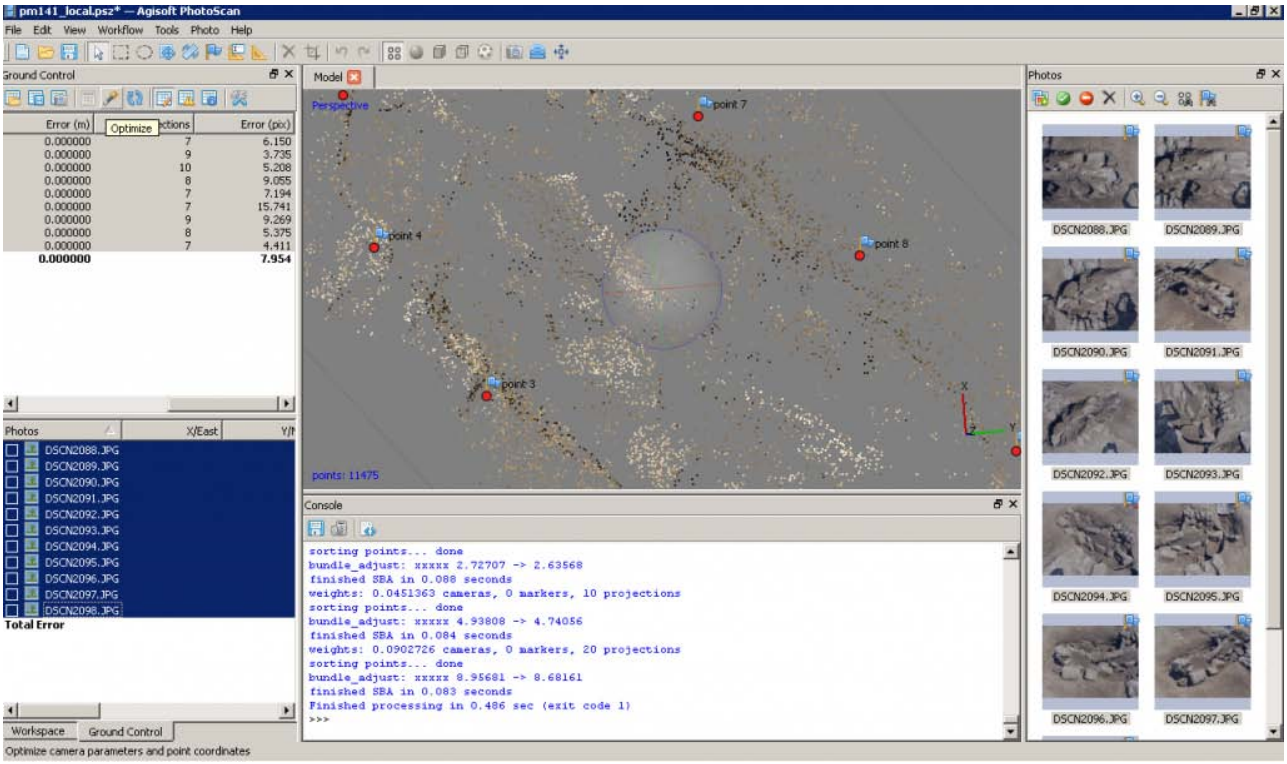


Local coordinates and projections

In practice we have found that many 3d modelling programs don't like it if the model is too far from the world's origin. This means that while Photoscan provides the tools for you to store your model in a real world coordinate system, and this works nicely for producing models as DEMs, you will need to use a local coordinate system if you want to produce models as .obj, .dae, .x3d or other modeling formats and work with them in editing programs like Rapidform or Meshlab. If your surveyed coordinates involve large numbers e.g. UTM coordinates, we suggest creating a local grid by splicing the coordinates so they only have 3-4 pre decimal digits.

Bundle Adjust – Another Choice

After all the points have been placed select all of them (checks on). If you believe the accuracy of the model is at least three time greater than the accuracy of the ground control survey you may select 'update' and the model will be block shifted to the ground control coordinates. If you believe the accuracy of the ground control survey is near to or greater than the accuracy of the model, you should include these points in your bundle adjustment to increase the overall accuracy of the model. To do this select 'optimize' from the 'Ground Control' menu after you have added the points. After the process runs, you can check the errors on each point. They should be less than 20 pixels. If the errors are high, you can attempt to improve the solution by turning off the surveyed points with the highest error, removing poorly referenced photos from the project, or adjusting the location of the surveyed points in individual images. After adjustments are made select 'update' and then 'optimize' again to reprocess the model.



Continue to…

Continue to PhotoScan – Building Geometry & Texture for Photogrammetry

]]> http://gmv.cast.uark.edu/photogrammetry/software-photogrammetry/photoscan/photoscan-workflow/photoscan-basic-processing-for-photogrammetry/feed/ 0 http://gmv.cast.uark.edu/gps/acquire-external-control-for-close-range-photogrammetry-with-trimble-survey-grade-gps/ http://gmv.cast.uark.edu/gps/acquire-external-control-for-close-range-photogrammetry-with-trimble-survey-grade-gps/#comments Fri, 15 Mar 2013 09:54:36 +0000 steph http://gmv.cast.uark.edu/?p=13098

This page is a guide for acquiring external control for close range photogrammetry using Trimble survey grade GPS.

Hint: You can click on any image to see a larger version.

Prepare for Survey

1. Begin metadata process
    1. Choose a method for documenting the project (e.g. notebook, laptop)
    2. Fill in known metadata items (e.g. project name, date of survey, site location, etc.)
    3. Create a sketch map of the area (by hand or available GIS/maps)
2. Choose and prepare equipment
    1. Decide what equipment will best suite the project
    2. Test equipment for proper functioning and charge/replace batteries

Equipment Setup

1. Base station
    1. Setup and level the fixed height tripod over the point of your choice
    2. Attach the yellow cable to the Zephyr antenna
    3. Place the Zephyr antenna on top using the brass fixture and tighten screw
    4. Attach the yellow cable to the 5700 receiver
    5. Attach the external battery to the 5700 receiver (if using)
    6. Attach the data cable to the TSCe Controller and turn the controller on
    7. Create a new file and begin the survey
    8. Disconnect TSCe Controller

2. Rover
    1. Put two batteries in the 5800
    2. Attach the 5800 to the bipod
    3. Attach TSCe Controller to bipod using controller mount
    4. Connect data cable to 5800 and TSCe Controller
    5. Turn on the 5800 and controller
    6. Create a new project file (to be used all day)

Collecting Points

1. Have documentation materials ready
    1. As you collect points, follow ADS standards
2. Base station
    1. Once started, the base station will continually collect positions until stopped
    2. When you're ready to stop it, connect the TSCe controller to the receiver and end the survey
3. Rover
    1. When you arrive at a point you want to record, set the bipod up and level it over the point
    2. Using the controller, create a new point and name it
    3. Start collecting positions for the point and let it continue for the appropriate amount of time
    4. Stop collection when time is reached and move to next position

Data Processing

1. Have documentation materials ready
    1. As you process the data, follow ADS standards
2. Transfer data
    1. Use Trimble Geomatics Office (TGO) to transfer data files from the TSCe Controller and the 5700

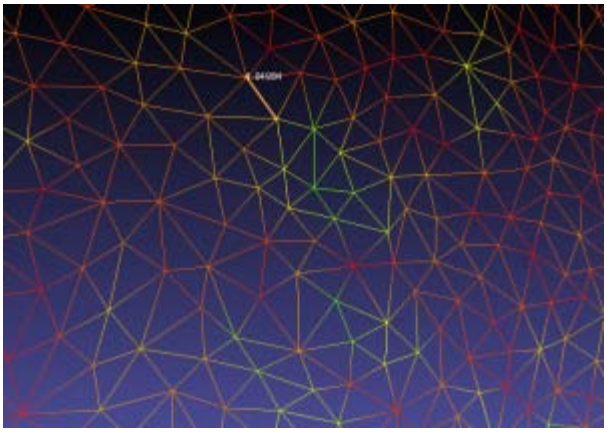
receiver to the computer

3. Calculate baselines
    1. Use TGO to calculate baselines between base station and rover points
    2. Apply adjustment and export points

]]> http://gmv.cast.uark.edu/gps/acquire-external-control-for-close-range-photogrammetry-with-trimble-survey-grade-gps/feed/ 0 http://gmv.cast.uark.edu/scanning/hardware/leica-c10/assessing-your-3d-model-effective-resolution/ http://gmv.cast.uark.edu/scanning/hardware/leica-c10/assessing-your-3d-model-effective-resolution/#comments Fri, 22 Feb 2013 14:17:08 +0000 Rachel http://gmv.cast.uark.edu/?p=12077

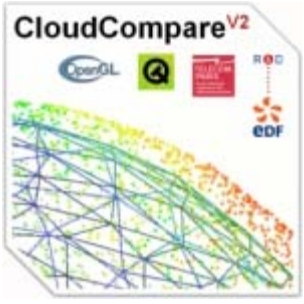
Why effective resolution?

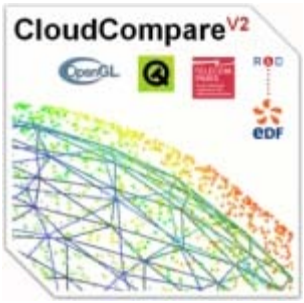
For many archaeologists and architects, the minimum size of the features which can be recognized in a 3D model is as important as the reported resolution of the instrument. Normally, the resolution reported for a laser scanner or a photogrammetric project is the point spacing (sometimes referred to as ground spacing distance in aerial photogrammetry). But clearly a point spacing of 5mm does not mean that features 5mm in width will be legible. So it is important that we understand at what resolution features of interest are recognizable, and at what resolution random and instrument noise begin to dominate the model.



Mesh vertex spacing circa 1cm.

Cloud Compare





The open source software Cloud Compare, developed by Daniel Girardeau-Montaut, can be used to perform this kind of assessment. The assessment method described here is based on the application of a series of perceptual metrics to 3D models. In this example we compare two 3D models of the same object, one derived from a C10 scanner and one from from a photogrammetric model developed using Agisoft Photoscan.

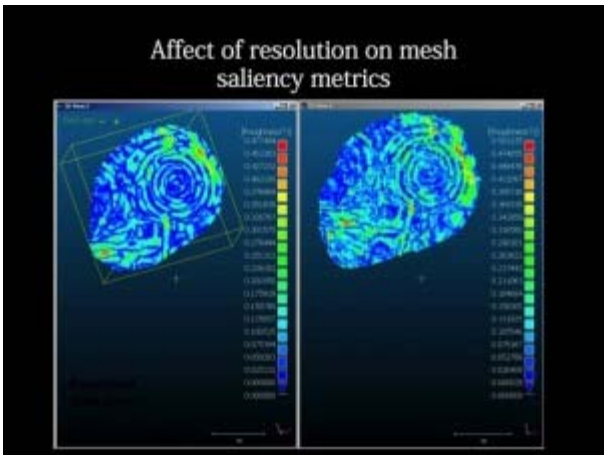
Selecting Test Features

Shallow but broad cuttings decorating stones are common features of interest in archaeology. The features here are on the centimetric scale across (in the xy-plane) and on the millimetric scale in depth (z-plane). In this example we assess the resolution at which a characteristic spiral and circles pattern, in this case from the 'calendar stone' at Knowth, Ireland is legible, as recorded by a C10 scanner at a nominal 0.5cm point spacing, and by a photogrammetric model built using Agisoft's photoscan from 16 images.

Perceptual and Saliency Metrics

Models from scanning data of photogrammetry can be both large and complex. Even as models grow in size and complexity, people studying them continue to mentally, subconsciously simplify the model by identifying and extracting the important features.

There are a number of measurements of saliency, or visual attractiveness, of a region of a mesh. These metrics generally incorporate both geometric factors and models of low-level human visual attention.

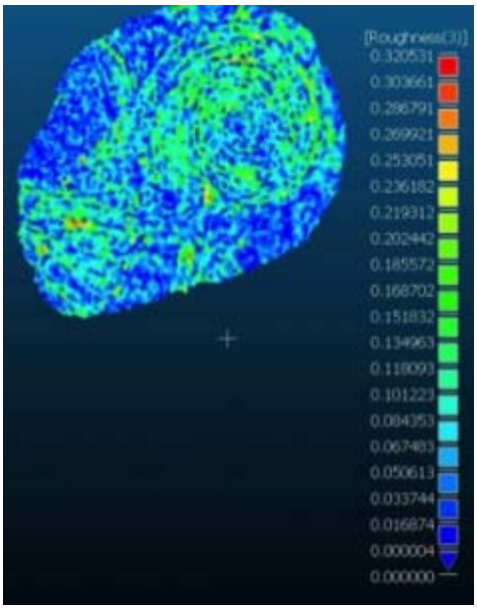


Local roughness mapped on a subsection of the calendar stone at Knowth.

Roughness is a good example of a relatively simple metric which is an important indicator for mesh saliency. Rough areas are often areas with detail, and areas of concentrated high roughness values are often important areas of the mesh in terms of the recognizability of the essential characteristic features. In the image above you can see roughness values mapped onto the decorative carving, with higher roughness values following the edges of carved areas.

Distribution of Roughness Values

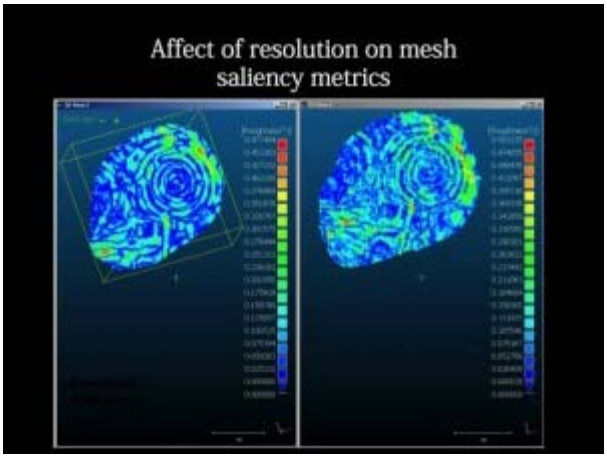
The presence of roughness isn't enough. The spatial distribution, or the spatial autocorrelation of the values, is also very important. Randomly distributed small areas with high roughness values usually indicate noise in the mesh. Concentrated, or spatially autocorrelated, areas of high and low roughness in a mesh can indicate a clean model with areas of greater detail.



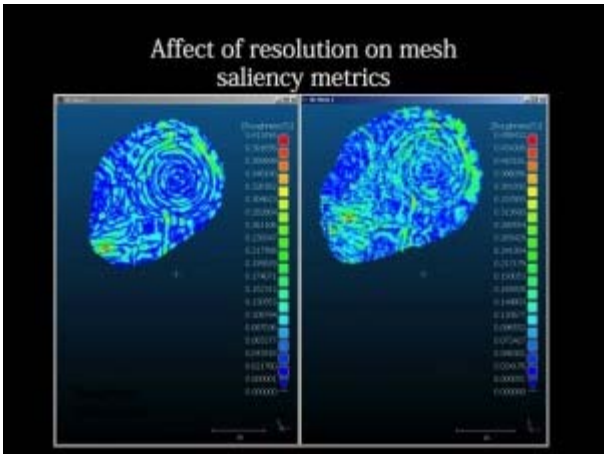
High roughness values combined with low spatial autocorrelation of these values indicates noise in the model.

Picking Relevant Kernel Sizes

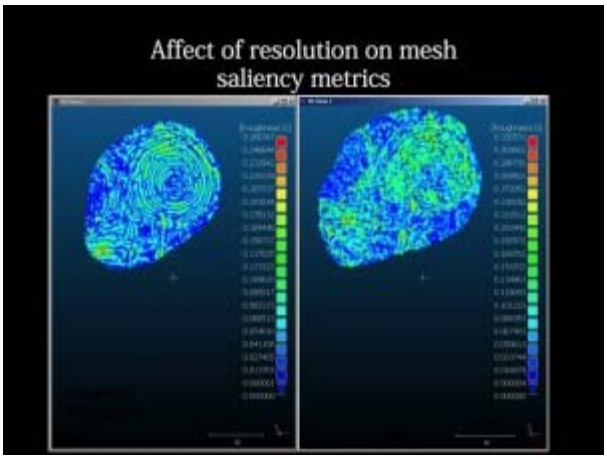
To use the local roughness values and their distribution to understand the scale at which features are recognizable, we run the metric over our mesh at different, relevant, kernel sizes. In this example, the data in the C10 was recorded at a nominal resolution of 5mm. We run the metric with the kernel at 7mm, 5mm, and 3mm.



Local roughness value calculated at kernel size: 7mm.



Local roughness value calculated at kernel size: 5mm.



Local roughness value calculated at kernel size: 3mm.

Visually we can see that the distribution of roughness values becomes more random as we move past the effective

resolution of the C10 data: 5mm. At 7mm the feature of interest -the characteristic spiral- is clearly visible. At 5mm it is still recognizable, but a little noisy. At 3mm, the picture is dominated by instrument noise.

]]> http://gmv.cast.uark.edu/scanning/hardware/leica-c10/assessing-your-3d-model-effective-resolution/feed/ 0 http://gmv.cast.uark.edu/photogrammetry/software-photogrammetry/photoscan/photoscan-workflow/basic-operation-of-the-epson-10000xl-flatbed-scanner-with-epson-scan-utility-software/ http://gmv.cast.uark.edu/photogrammetry/software-photogrammetry/photoscan/photoscan-workflow/basic-operation-of-the-epson-10000xl-flatbed-scanner-with-epson-scan-utility-software/#comments Wed, 13 Feb 2013 16:14:05 +0000 adam http://gmv.cast.uark.edu/?p=12336

This document will guide you through using the Epson 10000XL Flatbed Scanner to scan photographs and other media for many applicationsÂ including use in photogrammetry and archival storage.
Hint: You can click on any image to see a larger version.

GETTING STARTED

A current version of EPSON Scan Utility software can be freely downloaded from the Epson website and used to scan a variety of media, including transparent film and photographic prints.

Epson 10000XL Flatbed Scanner

To get started, make sure the scanner is connected to the computer and turn both the scanner and computer on. Log in to the computer and start the EPSON Scan software.

1. Mode – In the EPSON Scan dialog (Figure 1), change the “Mode” to “Professional Mode.”

2. Media – If scanning transparent film media, choose “Film” in the “Document Type” drop-down menu. If scanning paper, prints, or other reflective type media choose “Reflective.”

SETTINGS

3. The “Document Source” should always be set to “Document Table.”

4. In the “Image Type” drop-down menu, choose the appropriate setting for the media you’re scanning.
-When scanning transparent film media we recommend using a 16-bit Grayscale (for B&W film) or 24-bit Â Â Â Color (for color natural or false color film).

Figure 1: Settings for scanning with EPSON Scan software

5. Choose a resolution that is appropriate for the media you're scanning.

-Â Â When scanning transparent film media we recommend using a minimum resolution of 1200 dpi
-Â Â For high quality film, we recommend using 2400 or 3200 dpi in order to capture all of the available Â Â Â detail contained within the film
-Â Â When scanning print or paper media, a scanning resolution of 300-350 dpi should capture all of the Â Â Â available detail contained within the print.

6. Un-check the "Thumbnail"Â check box. Â All other settings in the EPSON Scan dialog will depend on the media you're scanning, or on your personal preference.

SCANNING

Figure 2: Transparent Film on Scan Bed

7. Placement – Carefully place the media face down in the upper left corner of the scan bed (Figure 2). We

recommend using clean gloves when handling transparent film or print media.

8. Click the "Preview" button at the bottom of the dialog and the scanner will begin scanning.

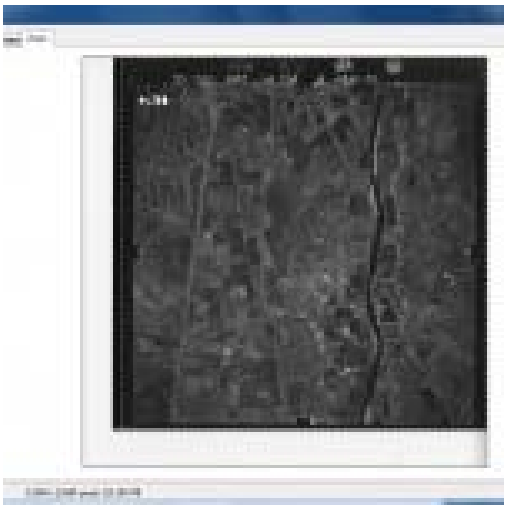


Figure 3: EPSON Scan software Preview

9. Once the preview scan is complete, the "Preview" dialog should appear (Figure 3). Use the Marquee tools to select the area of the media you would like to include in your scan. Be sure not to crop an image you plan on using for photogrammetry, and to include any visible fiducial marks.

10. Begin Scan – In the "EPSON Scan" dialog window, click "Scan" to start the scanning process.

SAVING YOUR FILE

11. In the "File Save Settings" dialog, choose a location, format, and name for output file.

> **NOTE:** *For best practice (and especially projects considering archival), we recommend scanning to the TIFF (.tif) file format.*

12. Time - Depending on the size of your media and the resolution you chose, the scanning process could take up to 1-2 hours.

]]> http://gmv.cast.uark.edu/photogrammetry/software-photogrammetry/photoscan/photoscan-workflow/basic-operation-of-the-epson-10000xl-flatbed-scanner-with-epson-scan-utility-software/feed/ 0 http://gmv.cast.uark.edu/photogrammetry/software-photogrammetry/photomodeler/workflow-photomodeler/pre-processing-digital-images-for-close-range-photogrammetry-crp/ http://gmv.cast.uark.edu/photogrammetry/software-photogrammetry/photomodeler/workflow-photomodeler/pre-processing-digital-images-for-close-range-

photogrammetry-crp/#comments Tue, 05 Feb 2013 20:19:06 +0000 adam http://gmv.cast.uark.edu/?p=12226

This page will show you how pre-process digital images for use in Close-Range Photogrammetry (CRP).
Hint: You can click on any image to see a larger version.

A BASIC INTRODUCTION

Why is pre-processing necessary?

For most close-range photogrammetry projects digital images will need to be captured in a RAW format, preserving the maximum pixel information which is important for archival purposes. Therefore it will likely be necessary to do some pre-processing in order to convert RAW images into a file format accepted by the photogrammetry software being used for the project.

If a color chart or gray card was using during image capture, it may also be useful to perform a white balance on the image set. There are a number of tools/software packages available for this purpose, but below we will describe a potential workflow using Adobe products for batch processing.

Overall steps of this workflow:

- Â Batch convert RAW to DNG (Adobe DNG Converter)
- Â Batch white balance (Camera Raw)
- Â Batch image adjustments (Camera Raw)
- Â Batch save to JPEG (or TIFF) format (Camera Raw)

BATCH CONVERT RAW DATA

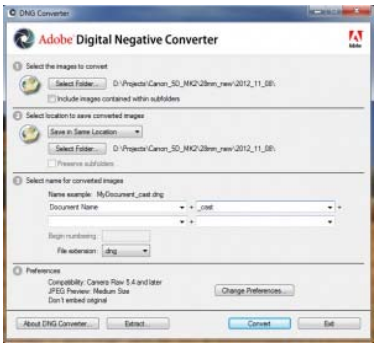
Batch RAW to DNG with Adobe Digital Negative (DNG) Converter Software

As an open extension of the TIFF/EP standard with support for EXIF, IPTC and XMP metadata, the Adobe DNG format is rapidly becoming accepted as a standards for storing raw image data (primarily from digital photography).

For more information about file formats for archival, see the Archaeological Data Service (ADS) Guides to Good Practice.

Steps to Batch Convert:

1. Download and install Adobe DNG Converter. As of the date this workflow was published, version 7.2 of Adobe DNG Converter is a free tool available for download on the Adobe website.



Adobe DNG Converter is a free tool available for download on the Adobe website.

2. This tool converts an entire folder (aka batch) of images at one time. Use the tool interface to select the appropriate input folder containing the RAW images.

3. If needed, use the interface to design a naming scheme to be used for the new file names.

4. Set preferences for compatibility (e.g. Camera Raw 5.4 and later) and JPEG Preview (e.g. medium size). As an option, you can embed the original RAW file inside the new DNG files. This will, or course, increase the file size of the new DNG file.

5. Click â€œConvertâ€ to start the process. Wait for this to finish.

BATCH WHITE BALANCE – 1

Batch white balance, image processing, and exporting with Adobe – Part 1: Adobe Bridge

It is considered best practice to (correctly) use a quality color chart or gray card when capturing digital images for any CRP project. **Â Performing a white balance for each image set (or each lighting condition) can dramatically enhance the appearance of a final product** (i.e. ortho-mosaic). This particular workflow uses Adobe Bridge and the Adobe Camera Raw tool, but a similar process can be done in other (free) software as well.

Adobe Bridge – Open in Camera Raw

1. Open Adobe Bridge and navigate to the folder containing the digital images (DNG files).

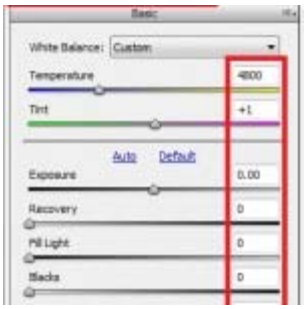
2. Select the appropriate images (including images with color chart/gray card).

3. Use the â€œFileâ€ menu to select â€œOpen in Camera Rawâ€

BATCH WHITE BALANCE – 2

Batch white balance, image processing, and exporting with Adobe – Part 2 : Camera Raw tool

4. Camera Raw will open and all of the selected images will appear on the left side of the window. Select the image with the color chart/gray card you would like to use for white balancing and other adjustments. Do all adjustments to this one image. We will apply the same changes to all images in the following slide ‘Batch Image Adjustment’.



Adobe Camera Raw – Image Processing Settings

5. **By default, Camera Raw may attempt to apply a number of image processing settings that you should remove**. This can be done using the interface on the right hand side of the screen. Check that all settings (with the exception of Temperature and Tint, which are set by the white balance tool in the next step) are set to zero. Be sure to check under each of the tabs.

6. Select the â€œColor Sampler Toolâ€ Â found in tool bar at the top of the window and:

A. If using a color chart, add a color sample inside the black and white squares. After adding these you should see the RGB pixel values for each sample.

B. If using a gray card, add a color sample inside the gray portion of the card.

7. Select the â€œWhite Balance Toolâ€ from the tool bar at the top of the window and click on the gray portion of the chart (or card) to apply a white balance. At the same time, notice how the RGB values of the color sample(s) change. The RGB values should not differ by more than five or six (e.g. the white sample could be R: 50, G: 50, B: 51). If they differ by too much there could be a problem with the white balance. Try clicking a slightly different spot in the gray portion of the chart.

8. If other adjustments need to be made (i.e. exposure, brightness, contrast) make them now.

BATCH IMAGE ADJUSTMENTS

Applying adjustments to all

Once the white balance and adjustments have been made to this one image, we can apply the same to all the other images open in Camera Raw.

To do this, click â€œSelect Allâ€ in the top left corner of the window â€“ then click â€œSynchronize.â€ Wait for this to finish.

BATCH SAVE TO JPEG OR TIFF

Saving

Once the Synchronization is complete, click the â€œSave Imagesâ€ in the bottom left corner of the window (make sure all images are still selected). The â€œSave Optionsâ€ dialog allows you to choose a folder for the images to be saved to, a naming scheme, a file extension and format, and a quality/compression. Choose the settings you prefer and click â€œSave.â€

[/wptabcontent]

CONTINUE TO…

Continue to PhotoScan – Basic Processing for Photogrammetry

]]> http://gmv.cast.uark.edu/photogrammetry/software-photogrammetry/photomodeler/workflow-photomodeler/pre-processing-digital-images-for-close-range-photogrammetry-crp/feed/ 0 http://gmv.cast.uark.edu/photogrammetry/hardware-photogrammetry/canon-5d-mark-ii/canon-5d-checklist/good-photos-vs-bad-photos-for-close-range-photogrammetry/ http://gmv.cast.uark.edu/photogrammetry/hardware-photogrammetry/canon-5d-mark-ii/canon-5d-checklist/good-photos-vs-bad-photos-for-close-range-photogrammetry/#comments Thu, 31 Jan 2013 16:37:42 +0000 adam http://gmv.cast.uark.edu/?p=12144 Continue reading →]]>

"Good" close-range photogrammetry example from Ostia Antica, Italy. Note that the object (the temple) is framed tightly, and that all objects (both near and far) are in sharp focus.

When it comes to close-range photogrammetry, the difference between â€œgoodâ€ photos and â€œbadâ€ photos can be the difference between getting useful 3D information and having complete failure. There are many different variables contributing to success or failure of a project, but to help avoid the most common mistakes a photographer can follow the general guidelines outlined below.

Basic photographic concepts that, when followed, generally produce acceptable digital images:

Camera/lens Properties:
-Use a mid to high resolution camera (at least 12-15MP)
-Use a fixed (non-zoom) lens
-Tape the focus ring (and set to manual focus)

-If using a zoom lens, tape the zoom ring and use one focal length for the entire project

Camera Placement:
-Use a tripod and stable tripod head
-Frame the subject tightly, making use of the entire sensor area
-Maintain 60-80% overlap between photos
-Ensure all important areas of the object are visible in at least three images
-Be aware of camera geometry required by software (baseline, convergent angles)

Camera Settings:
-Use aperture priority mode (set to between f/8 and f/16)
-Use a timer or wired/wireless shutter release to minimizes motion blur
-Use mirror lock-up, if available, to further minimizes motion blur

A list of common mistakes made while capturing digital images for a close-range photogrammetry project:

Camera/lens:

"Bad" close-range photogrammetry example. Note that the object (the temple) is not framed tightly, and that most objects are blurry and out of focus.

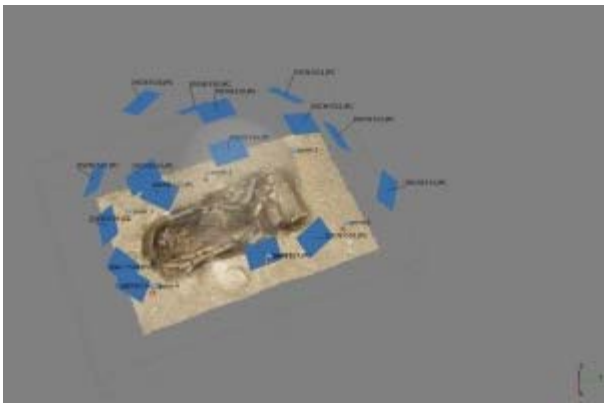
-Use of low resolution camera (8MP or less)
-Changing zoom (focal length) between images
-Use of loose/damaged lens
-Significant re-focusing due to varying distance from object

Camera placement:
-Handheld camera (no tripod)
-Insufficient overlap between images
-Inefficient use of sensor area (too far from subject)
-weak camera geometry (multiple images from one position, short baseline, overall poor network of image locations/orientations)

Camera settings:
-shallow depth of field (below f/8)
-manual shutter release (causes motion blur)

]]> http://gmv.cast.uark.edu/photogrammetry/hardware-photogrammetry/canon-5d-mark-ii/canon-5d-checklist/good-photos-vs-bad-photos-for-close-range-photogrammetry/feed/ 0 http://gmv.cast.uark.edu/region-data/data-photogrammetry/gabii-photogrammetry/ http://gmv.cast.uark.edu/region-data/data-photogrammetry/gabii-photogrammetry/#comments Mon, 14 Jan 2013 18:00:42 +0000 Rachel http://gmv.cast.uark.edu/?p=11943 Continue

reading →]]>



The Gabii Project is an international archaeological project, directed by Nicola Terrenato of the University of Michigan. The Gabii Project began in 2007, seeking to study the ancient Latin city of Gabii through excavation and survey. Gabii, located in central Italy, was a neighbor of and rival to Rome, and flourished during in the first millennium BC.

The excavations at Gabii are uncovering extensive and complex remains within the city's urban core. Convergent photogrammetry is essential to the project's recording strategy. At Gabii, this technique is used to document features with complex geometries or large numbers of inclusions, including walls, pavements, rubble collapse, and architectural elements. These types of features can be quite time-consuming to document thoroughly by hand or using conventional surveying in the field. The 3D models collected in the field are georeferenced. They are subsequently simplified for incorporation into the project's GIS, and compiled into models for distribution online using Unity3D.

d install the free Unity

View Gabii Project in a larger map

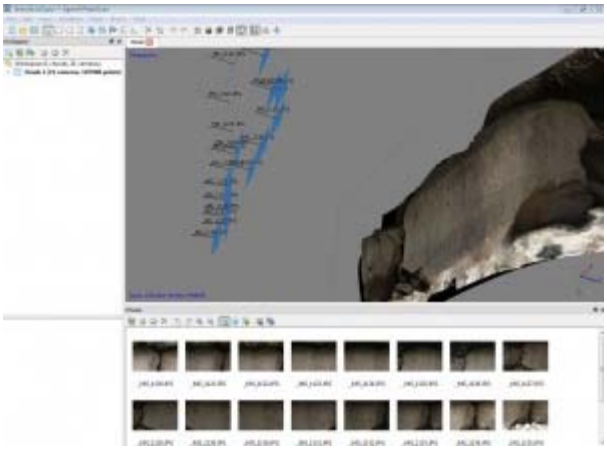
]]> http://gmv.cast.uark.edu/region-data/data-photogrammetry/gabii-photogrammetry/feed/ 0 http://gmv.cast.uark.edu/region-data/data-photogrammetry/knowth-photogrammetry/ http://gmv.cast.uark.edu/region-data/data-photogrammetry/knowth-photogrammetry/#comments Mon, 24 Dec 2012 17:13:30 +0000 Rachel http://gmv.cast.uark.edu/?p=10317 Continue reading →]]>

Detail from the model of the K11 kerbstone, showing decorative carving on the rock surface.

The archaeological complex at Knowth, located in the BrÃº na BÃ³inne World Heritage Site, consists of a central mound surrounded by 18 smaller, satellite mounds. These monuments incorporate a large collection of megalithic art, primarily in the form of decorated stones lining the mounds' internal passages and surrounding their external bases. The megalithic art found at this site constitutes an important collection, as the Knowth site contains a third of the of megalithic art in all Western Europe. The kerbstones surrounding the main mound at Knowth, while protected in winter, sit in the open air for part of the year, and are consequently exposed to weather and subject to erosion. The Researchers at CAST, in collaboration with UCD Archaeologists and Meath County Council, documented the 127 kerbstones surrounding the central mound at Knowth over the course of two days using close range convergent photogrammetry. This pilot project aims to demonstrate the validity of photogrammetry as the basis for monitoring the state of the kerbstones and to add to the public presentation of the site, incorporating the models into broader three dimensional recording and documentation efforts currently being carried out at Knowth and in the BrÃº na BÃ³inne, including campaigns of terrestrial laserscanning and aerial lidar survey.

The k15 kerbstone is available here as a sample dataset. You can download the 3D pdf (low res) or the DAE file (high res).

Photogrammetry data from this project was processed using PhotoScan Pro.



Photoscan Pro processing of the model for the K15 kerbstone.

View Knowth TLS and Photogrammetry in a larger map

]]> http://gmv.cast.uark.edu/region-data/data-photogrammetry/knowth-photogrammetry/feed/ 0 http://gmv.cast.uark.edu/uncategorized/microsoft-kinect-setting-up-the-development-environment/ http://gmv.cast.uark.edu/uncategorized/microsoft-kinect-setting-up-the-development-environment/#comments Tue, 11 Dec 2012 11:59:04 +0000 Matt T http://gmv.cast.uark.edu/?p=11061

Using Eclipse IDE

Since there is a plethora of existing tutorials guiding how to set up various development environments in C++, I will show you how to set up the 32-bit OpenNI JAR (OpenNI Java wrapper) in Eclipse IDE and to initialize a production node to begin accessing Kinect data via the Java programming language.
To continue we will be working with the open-source and fantastic piece of software known as Eclipse that you can find here: www.eclipse.org. You will want to download the IDE for Java programmers located on their â€œdownloadsâ€ page(about 149 mb). Take note of the many other software solutions that they offer and the vast amount of resources on the site.

> NOTE: Even though we are downloading the â€œJavaâ€ Eclipse IDE you can easily add plugins to use this same piece of software with Python, C/C++, and many other applications.

Additionally, we are assuming that you have already gone through the OpenNI installation located here.

You also need to have the Java JDK installed (www.oracle.com).

Finally, to gain access to one of the best open-source computer vision libraries available, you will need to download and install OpenCV(http://opencv.org/) and the JavaCV(http://code.google.com/p/javacv/). The installation instructions located on each of these sites are excellent.

Setting Up Eclipse with OpenNI: Before You Start

Important Note: As you may already be aware, these tutorials are focused on the Beginner Level user, not only

to using the Kinect but also to programming. Before going any further I should also remind you that if jumping “head first” into the new domain of programming isnâ€™t something for which you have the interest or the time, there are many things you can accomplish with the â€œready to use softwareâ€ solutions located here.

Also, before starting, make sure that you are using the same platform (32 â€“bit to 32-bit/64 to 64) on the Eclipse IDE, Java JDK, and OpenNI installation.

Eclipse with OpenNI: Starting a New Java Project

Starting a New Java Project â€¦
Once you have downloaded Eclipse, installed the java JDK and the OpenNI/Primesense, you will need to start a new Java Project. Following the wizard is the easiest way to do this.

Check the box that says â€œpublic static void main(String[] args)â€ so that Eclipse will add a few lines of code for us.

> **NOTE: For this tutorial I have kept the names fairly vague – be sure to use names that you will remember and understand. Remember that if you use a different naming convention than shown here, you will need to make corrections in the sample code to fit to your specifications.**

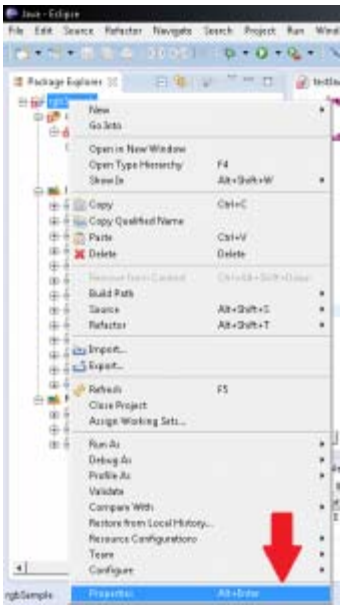
Eclipse with OpenNI: Adding the OpenNI Libraries Part 1

Adding the OpenNI librariesâ€¦

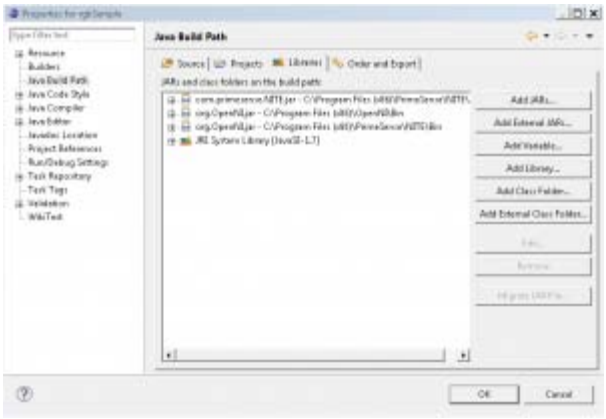
Next we will need to add the OpenNI libraries to the project. This is a pretty straight forward process in Java and Eclipse, simply being a matter of adding the pre-compiled JAR file from the â€œbinâ€ folder of your OpenNI installation directory.

> NOTE: If you plan on using User Tracking or another Primesense middleware capability you will need to add the JAR in the Primesense directory

To do so right-click on the project we just created:



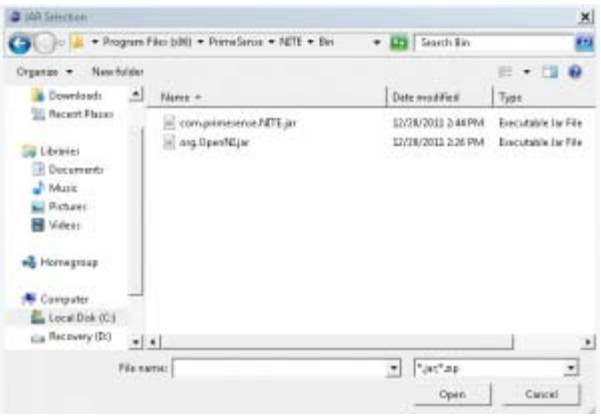
And select the â€œPropertiesâ€ menu item.
Then we will want to select the â€œJava Build Pathâ€ and â€œAdd External Jarâ€™sâ€ button.



Repeat the same steps as above for the JavaCV JARâ€™s that you previously installed somewhere on your machine.

Eclipse with OpenNI: Adding the OpenNI Libraries Part 2

Navigate to the â€œbinâ€ folders of the install directories for OpenNI and Primesense.
On my Windows 7 64-bit machine with the 32-bit install it is located here:



Note: There are TWO OpenNI â€œJARâ€ files â€“ one in the bin folder of the OpenNI install directory as well as one in the Primesense directory. I havenâ€™t noticed any difference in using one over the other; as long as your environment paths in Windows are set up to locate the needed files, they should both work.

After this, you should see these files in the â€œReferenced Librariesâ€ directory on the â€œPackage Explorerâ€ tool bar in Eclipse.

Eclipse with OpenNI: Projects

We should now be able to access the Kinect via Java and Eclipse.

In the following projects we will introduce and attempt to explain the necessary steps for initializing the Kinect via OpenNI and for getting basic access to its data feeds in Java.

Each project goes through setting up the Kinect in OpenNI and includes comments to explain line-by-line what is going on.

For information on Visual Studio 2010 & Microsoft C# SDKâ€¦.

Using the Microsoft SDK provides a lot of advantages and ease of access, but it is also only applicable to the â€œKinect for Windowsâ€ hardware and not the Xbox Kinect (as of v1.5).

There are a lot of existing tutorials on the web about setting up your development environment with plenty of sample projects. Below is a list of links to a few of them in no particular order as to avoid reinventing the wheel.

1. http://channel9.msdn.com/Series/KinectQuickstart/Setting-up-your-Development-Environment
2. http://social.msdn.microsoft.com/Forums/el/kinectsdk/thread/7011aca7-defd-445a-bd3c-66837ccc716c
3. http://msdn.microsoft.com/en-us/library/hh855356.aspx
4. Power Point from Stanford

]]> http://gmv.cast.uark.edu/uncategorized/microsoft-kinect-setting-up-the-development-environment/feed/ 0
http://gmv.cast.uark.edu/uncategorized/example-projects-for-the-kinect/
http://gmv.cast.uark.edu/uncategorized/example-projects-for-the-kinect/#comments Thu, 06 Dec 2012 15:32:35 +0000 Matt T http://gmv.cast.uark.edu/?p=11060

Overview

As previously mentioned, the OpenNI API is written in C++ but once you follow the installation procedures covered, here, you will have some pre-compiled wrappers that will give you access to use OpenNI in a few other languages if you need.

Since there is a plethora of existing tutorials regarding setting up various development environments in C++ and corresponding example projects, this article will show you how to setup the 32-bit OpenNI JAR (OpenNI Java wrapper) in Eclipse IDE. We will then initialize an OpenNI production node to begin accessing Kinect data and to get the RGB stream into OpenCv, which is a popular computer vision library.

Before going on to the following project, make sure that you have all of the dependent libraries installed on your

machine. For the instructions on getting the 3rd party libraries and for setting up the development environment check out this post.

Also, I want to clarify that this code is merely one solution that I managed to successfully execute. This said, it may have bugs and/or mayb be done more successfully or more easily using a different solution. If you have any suggestions or find errors, please donâ€™t hesitate to contact us and I will change the post immediately. These posts follow and continue to follow exploration and collaboration.

Using the Kinectâ€™s RGB feed

In this project we will:

1. Make a simple program to capture the RGB feed from the Kinect in Java
2. Get the data into an OpenCV image data structure
3. Display the data on the screen

A high-level overview of the steps we need to take are as follows:

1. Create a new â€˜contextâ€™ for the Kinect to be started
2. Create and start a â€˜generatorâ€™ which acts as the mechanism for delivering both data and metadata about its corresponding feed
3. Translate the raw Kinect data into a Java data structure to use in native Java libraries
4. Capture a â€œframeâ€ and display it on screen

The next tab is the commented code for you to use as you wish.

> NOTE: For extremely in-depth and excellent instruction on using JavaCV, the Kinect, along with various other related projects I extremely the book(s) by Andrew Davison from the Imperial College London. A list of his works can be found here http://www.doc.ic.ac.uk/~ajd/publications.html and here http://fivedots.coe.psu.ac.th/~ad/.

Sample RGB Project – Part 1

First, let's import the required libraries:

```
import org.OpenNI.*;

import com.googlecode.javacv.*;
import static com.googlecode.javacv.cpp.opencv_imgproc.*;
import com.googlecode.javacv.cpp.opencv_core.IplImage;
```

Then eclipse nicely fills out our class information.

```
public class sampleRGB {
```

We define some global variables

```
static int imWidth, imHeight;

static ImageGenerator imageGen;
static Context contex
```

Eclipse will also fill out our "main" statement for use by checking the box on the project set up. One addition we

will need to make is to surround the following code block with a statement for any exceptions that may be thrown when starting the data feed from the Kinect. Here we are starting a new "context" for the Kinect

```
public static void main(String[] args) throws GeneralException {
```

Create a "context"

```
context = new Context();
```

Sample RGB Project – Part 2

We are manually adding the license information from Primesense. You can also directly reference the xml documents located in the install directory of both the OpenNI and Primesense.

```
License license = new License("PrimeSense", "0KOIk2JeIBYClPWVnMoRKn5cdY4=");
context.addLicense(license);
```

Create a "generator" which is the machine that will pump out RGB data

```
imageGen = ImageGenerator.create(context);
```

We need to define the resolution of the data coming from the image generator. OpenNI calls this mapmode (imageMaps, depthMaps, etc.). We will use the standard resolution.

First initialize it to null.

```
MapOutputMode mapMode = null;
mapMode = new MapOutputMode(640, 480, 30);
imageGen.setMapOutputMode(mapMode);
```

We also need to pick the pixel format to display from the Image Generator. We will use the Red-Green-Blue 8-bit 3 channel or "RGB24"

```
imageGen.setPixelFormat(PixelFormat.RGB24);
```

Sample RGB Project – Part 3

OpenNI also allows us to easily mirror the image so movement in 3d space is reflected in the image plane

```
context.setGlobalMirror(true);
```

Create an Iplimage(opencv image) with the same size and format as the feed from the kinect.

```
IplImage rgbImage = IplImage.create(imWidth,imHeight, 8, 3);
```

Next we will use the easy route and utilize JFrame/Javacv optimized canvas to show the image

```
CanvasFrame canvas = new CanvasFrame("RGB Demo");
```

Now we will create a never-ending loop to update the data and frames being displayed on the screen. Going line by line we will, update the context every time the image generator gets new data.
Set the opencv image data to the byte buffer created from the imageGen.

NOTE: For some reason the channels coming from the Kinect to the opencv image are ordered differently so

we will simply use the opencv convert color to set the "BGR" to "RGB". We tell the canvas frame that we created to show image.

Sample RGB Project – Part 4

Finally, we need to also release the Kinect context or we will get an error the next time we try to start a node because the needed files will be locked

```
while (true){
context.waitOneUpdateAll(imageGen);
rgbImage.imageData(imageGen.createDataByteBuffer());
cvCvtColor(rgbImage, rgbImage, CV_BGR2RGB);
canvas.showImage(rgbImage);

canvas.setDefaultCloseOperation(CanvasFrame.EXIT_ON_CLOSE);
}
```

****Note that you can add an argument to the canvas frame to reverse the channels of the image**

]]> http://gmv.cast.uark.edu/uncategorized/example-projects-for-the-kinect/feed/ 0