3-D Machine Vision for factory automation, robotic vision and industrial applications -

Generate your own point clouds using 3-D Machine Vision SDK

Ben Lee – presenting (material by Blair Conner)

May 2015

This presentation is reference information and not considered a complete or official training guide

This presentation is reference information and is not considered to be a complete or official training guide.

This presentation may contain TI reference design materials. TI reference design materials are governed by TI's "Important Notice for Reference Designs" – please see http://www.ti.com/refdesignnotice for more details.

This presentation may contain references to third party materials and/or software ("Third Party Content"). Third Party Content is not under TI's control and TI is not responsible for such content, or any changes or updates to such content. TI provides information about and links to Third Party Content to you only as a convenience and TI does not endorse such content. Use of Third Party Content is governed by the applicable terms of use of the third party.

This material is prepared exclusively and NOT intended for general distribution.

Agenda

- **Introduction to 3-D machine vision**
  - What are the basics of 3-D machine vision?
- **What is the 3-D machine vision reference design using DLP® technology?**
  - What is included in the software?
  - Where do I download it?
- **Design Considerations for 3-D Machine Vision**
  - How to choose a DLP chipset
  - Scanning and reconstruction considerations
- **Developing with the DLP Structured Light SDK**
  - How to setup the development environment?
  - How to create new modules?

Introduction to 3-D Machine Vision

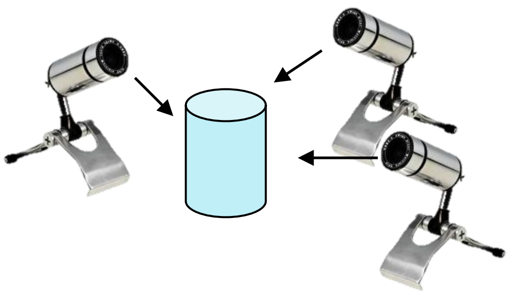
Many methods for 3-D machine vision

Use Triangulation (Geometry) to Determine the Depth of an Object

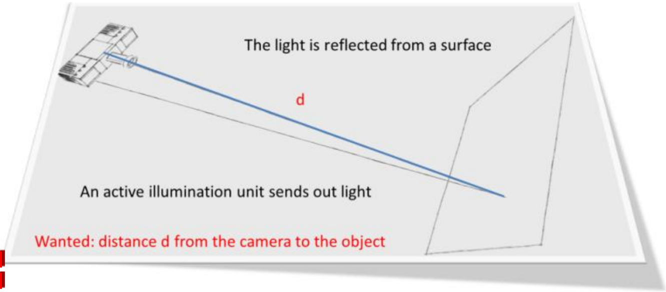
By Different Methods:

Single Line Laser Scan

Stereo Triangulation or Photogrammetry



Time of Flight



Structured Light

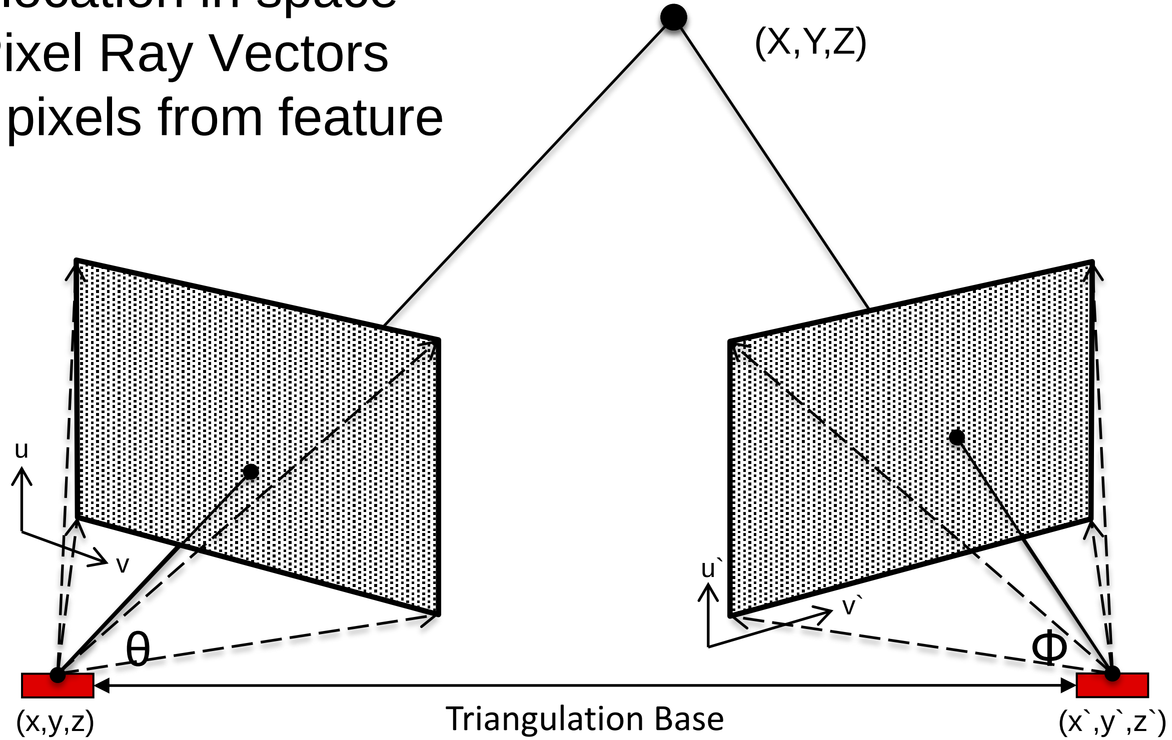
Depth calculated with triangulation

Use Triangulation (Geometry) to Determine the Depth of an Object

By Different Methods:

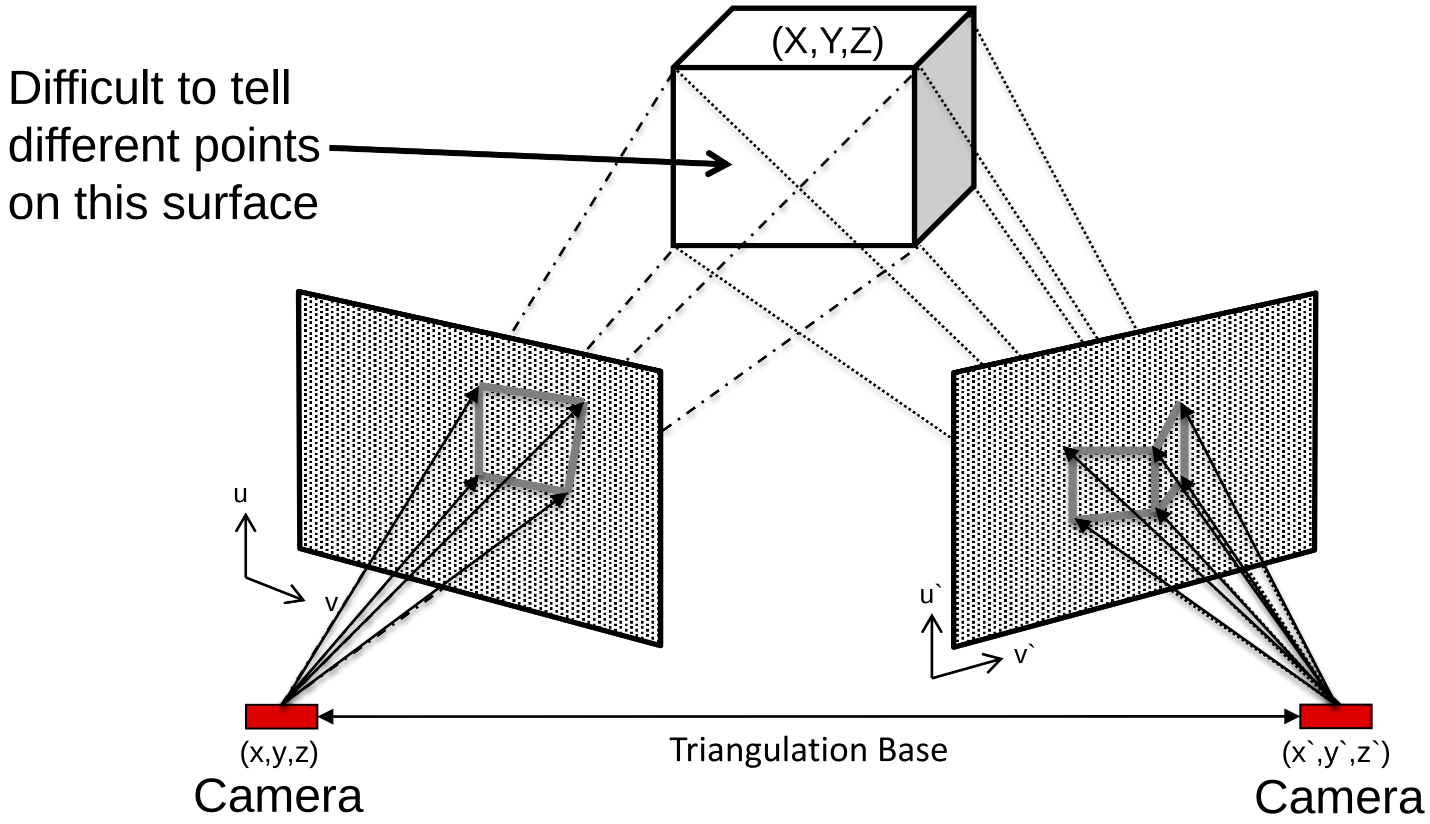
What do we need to know?

- Physical location in space
- Optical Pixel Ray Vectors
- Matched pixels from feature



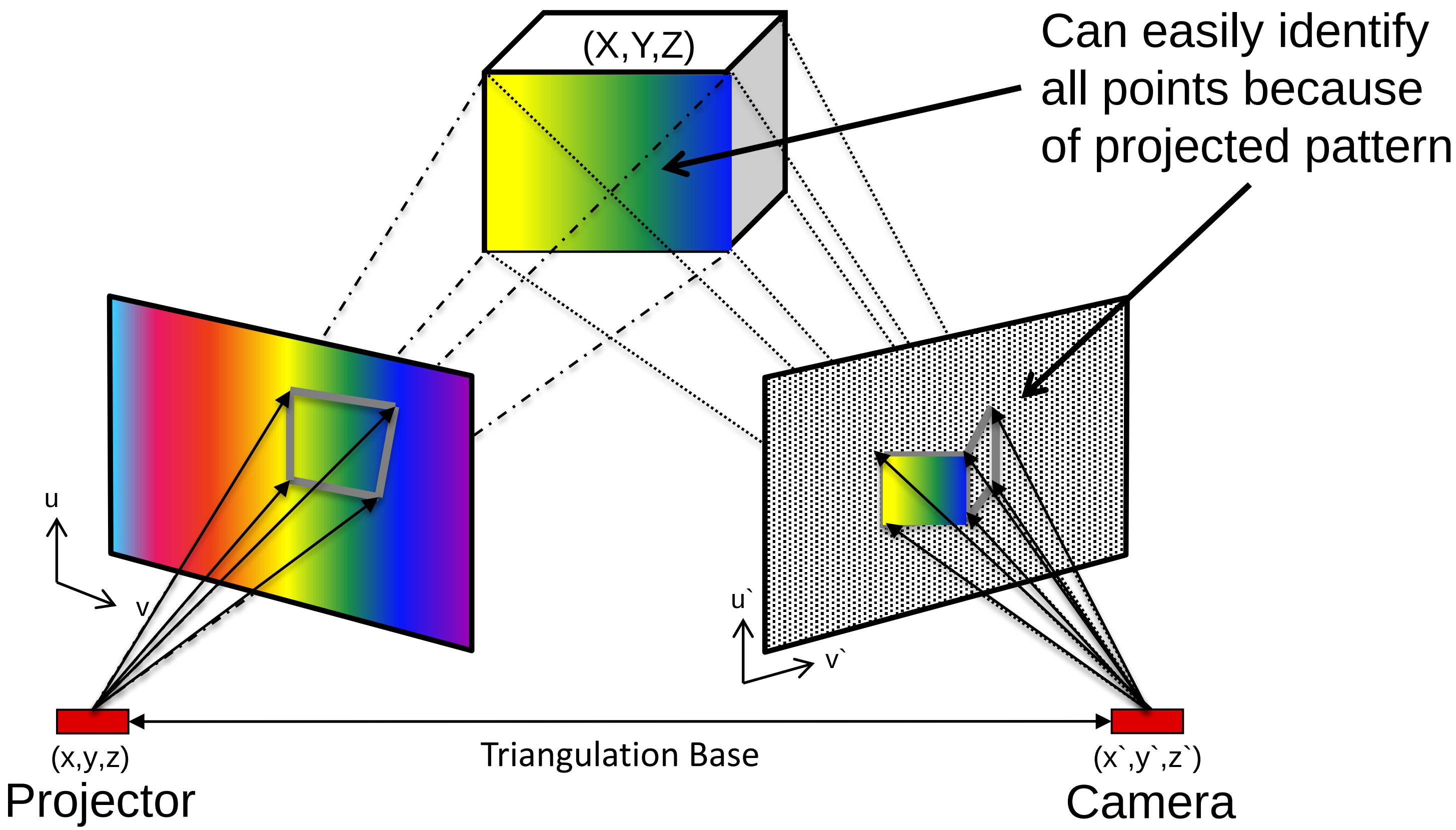
Using Cameras for 3-D Machine Vision

- Two cameras capture different viewpoints of the same object
- What if object does not have many identifiable features?



3-D Machine Vision with Structured Light

- One camera captures projected patterns
- The projected patterns inherently create identifiable features



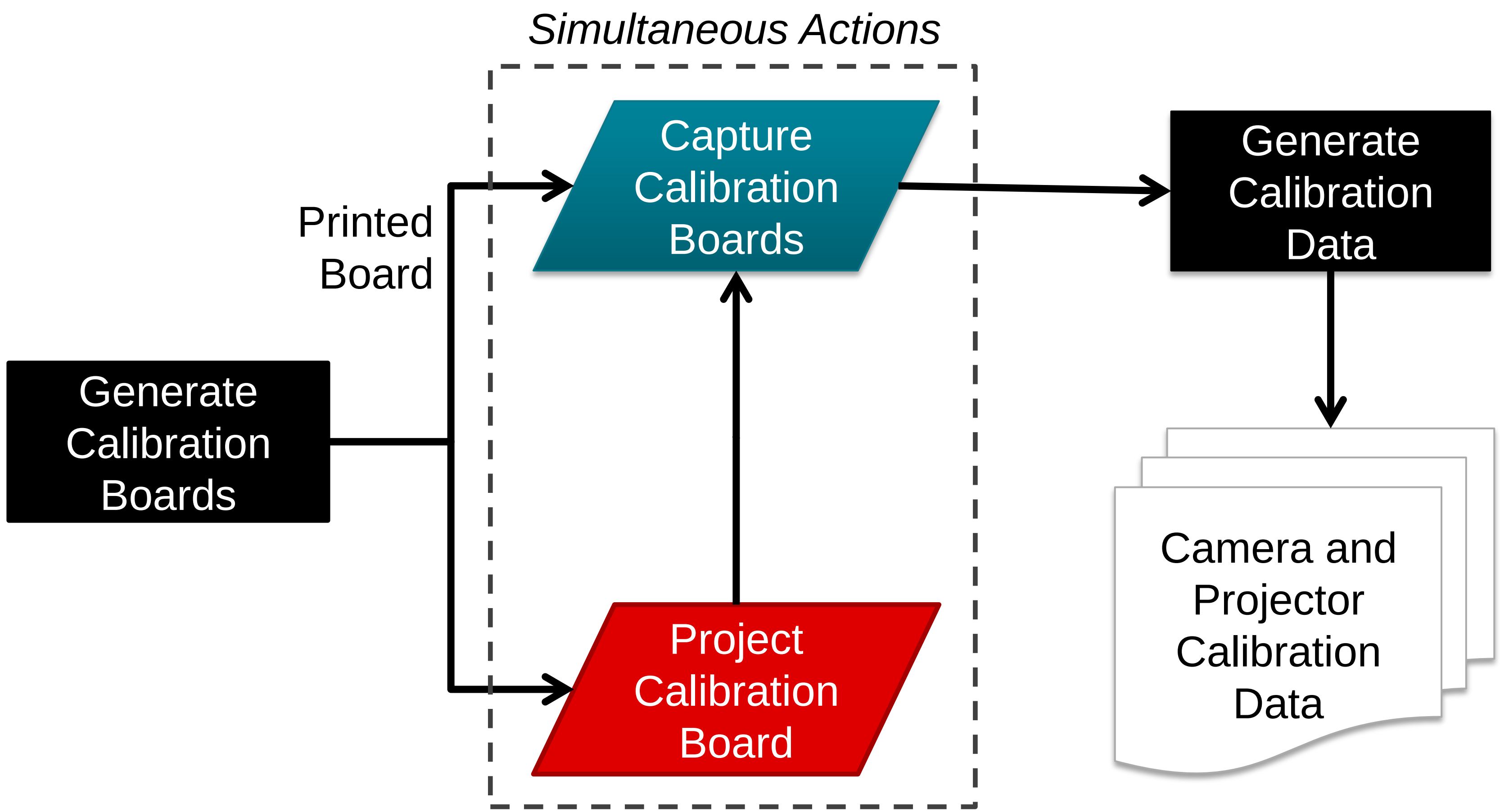
What subsystems are needed?

Subsystems	Purpose

Calibration	• Determine physical locations and directions
	• Determine optical parameters of camera and projector
	  ○ Focal length
	  ○ Focal point
	  ○ Radial distortion coefficients

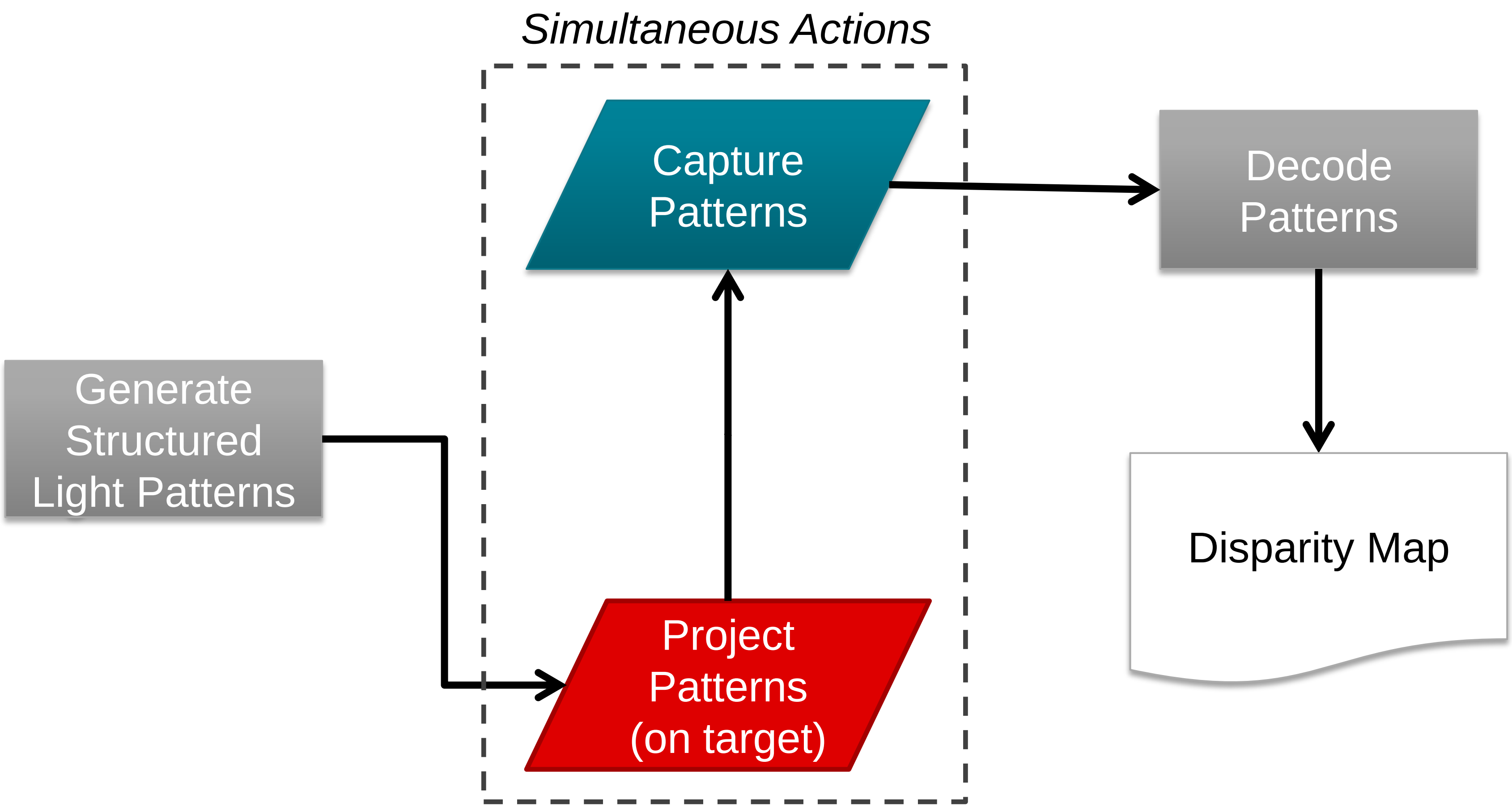
Camera	• Capture high-speed patterns for analysis
DLP Platform	• Project high-speed patterns
Structured Light	• Generate structured light patterns • Decode captured images and generate disparity map which details which projector pixels are viewed by camera
Geometry	• Uses calibration data and feature identification to reconstruct (X,Y,Z) point

Process Flow: Calibration



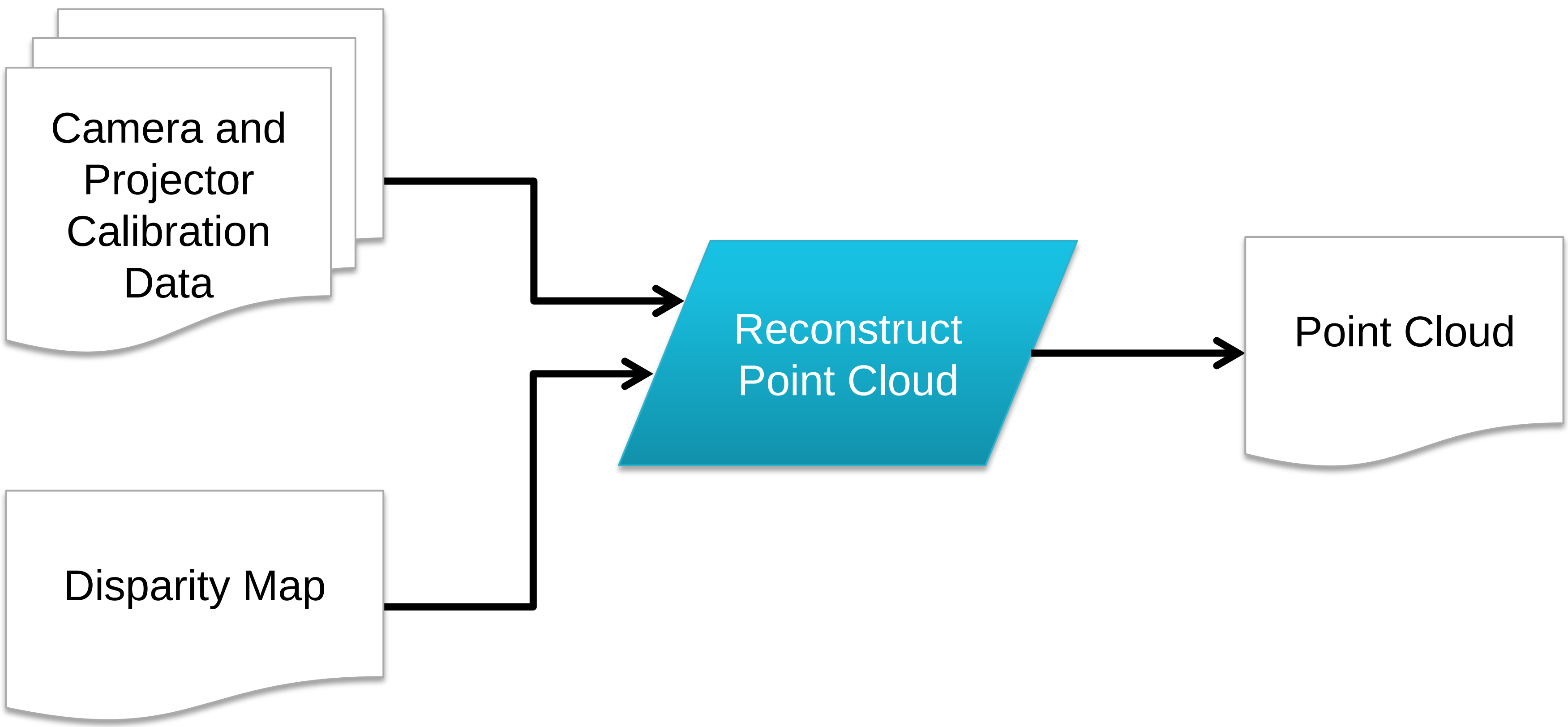
Process Flow: Scanning

Calibration | Camera | DLP® Platform | Structured Light | Geometry



Process Flow: Reconstruction

Calibration | Camera | DLP® Platform | Structured Light | Geometry

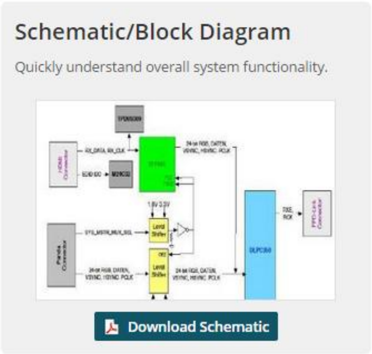


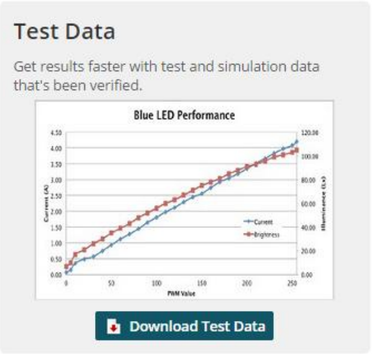
What is the DLP® 3-D machine vision reference design?

Jump Start Development with TI Designs

What are TI Designs?

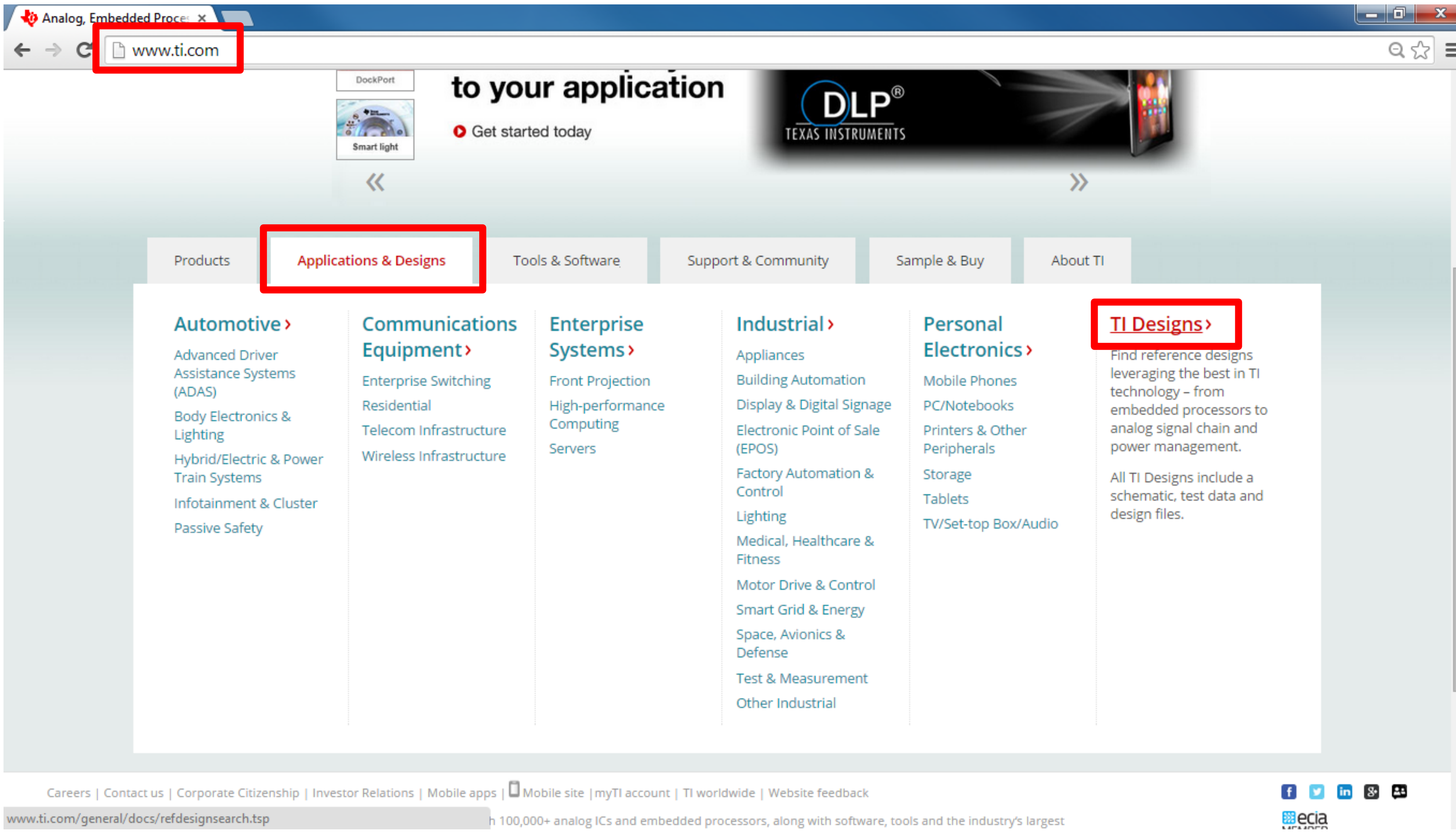
- Comprehensive designs include schematics or block diagrams, BOMs, design files and test reports
- Created by experts with deep system and product knowledge
- Spans TI's portfolio of analog, embedded processor and connectivity products
- Supports a broad range of applications including industrial, automotive, consumer, medical and more





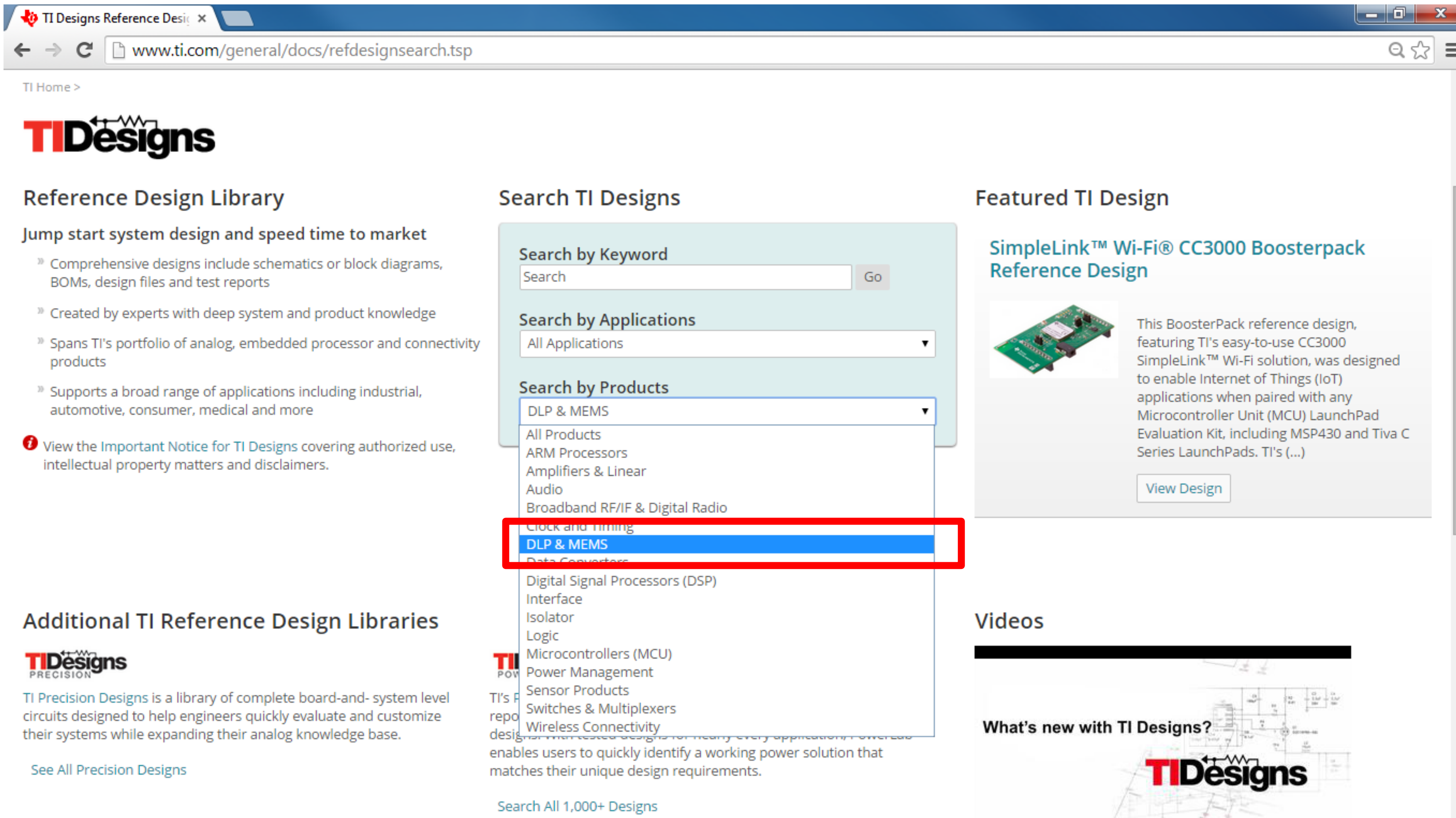
Jump Start Development with TI Designs

Where to find DLP TI Designs?



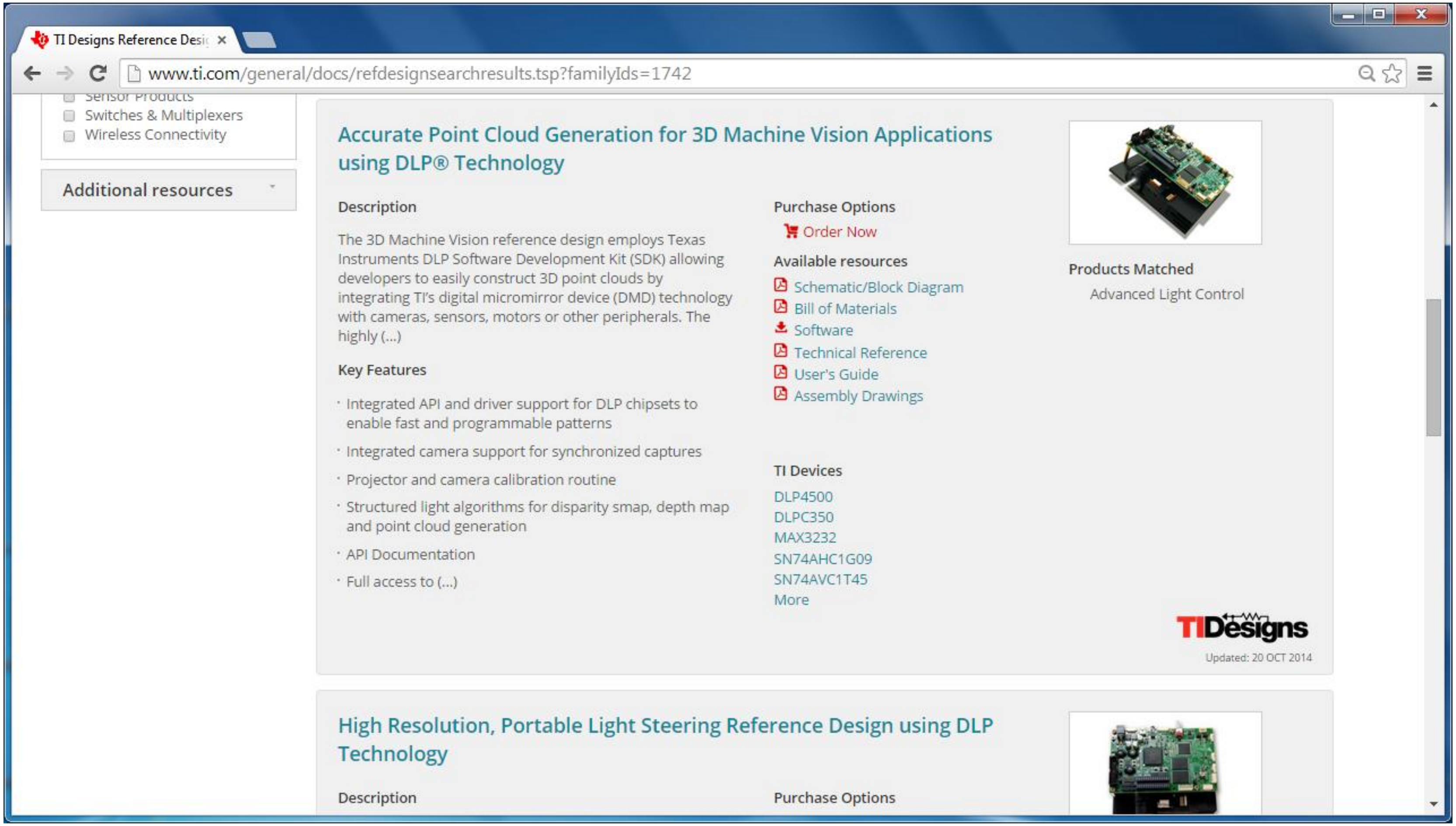
Jump Start Development with TI Designs

Where to find DLP TI Designs?



Jump Start Development with TI Designs

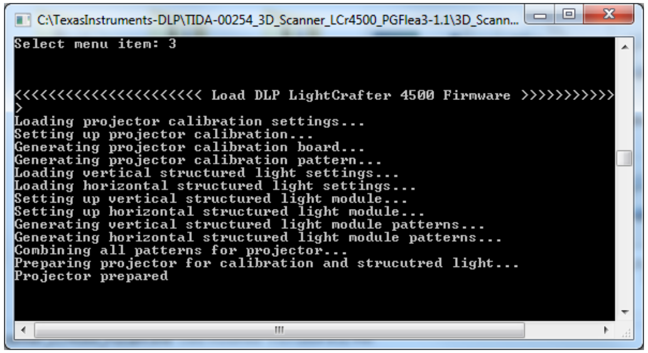
Where to find DLP TI Designs?



Accurate Point Cloud Generation for 3-D Machine Vision Applications using DLP® Technology

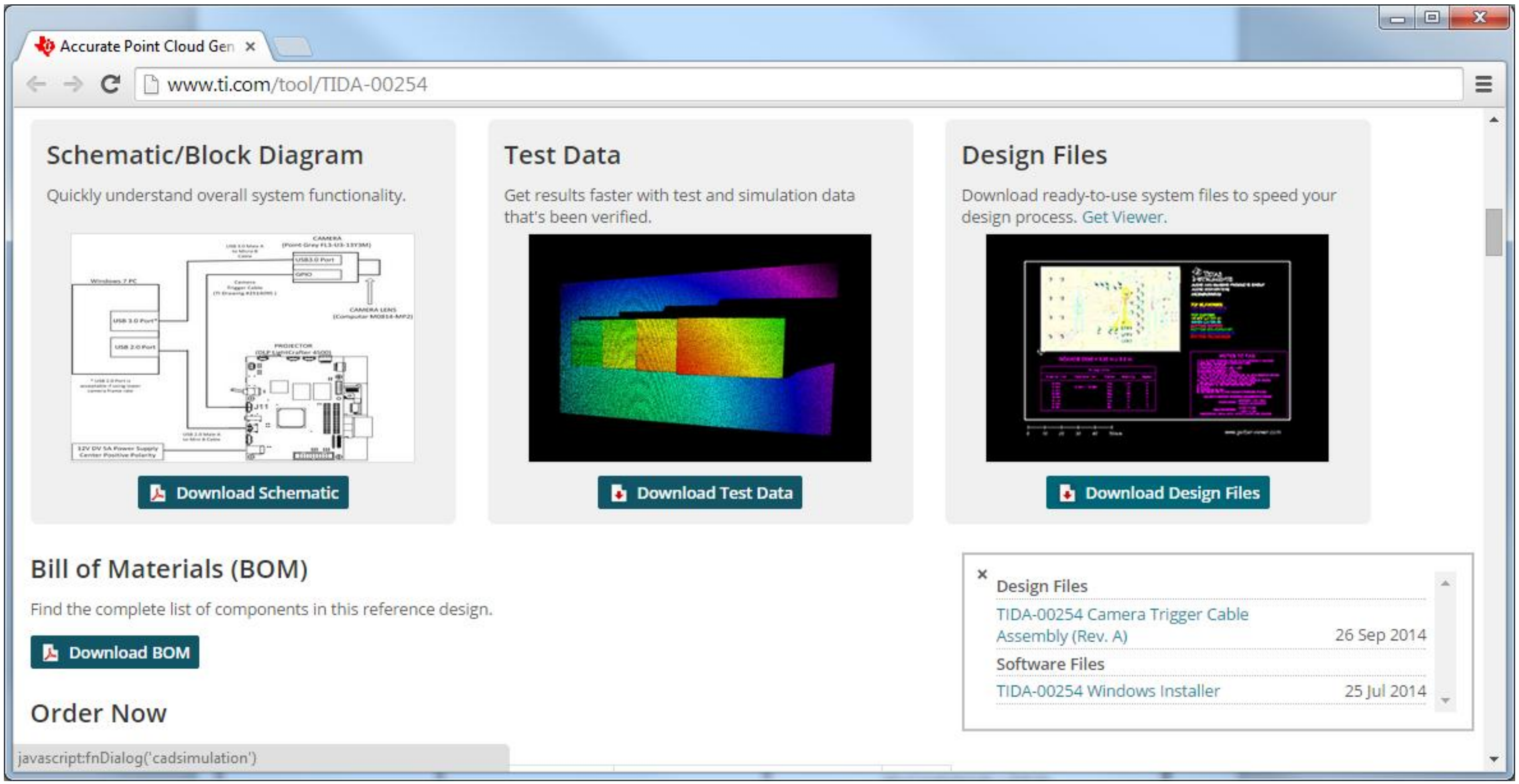
- Provides a 3-D scanner software solution to shorten development time with DLP technology
- No software or optical development required for evaluating DLP technology for generating 3-D point clouds!
  - Generates and decodes structured light patterns
  - Calibrates camera and projector
- Available design files
  - Schematic (http://www.ti.com/lit/pdf/tidr157)
  - Bill of Materials (http://www.ti.com/lit/pdf/tidr958)
  - Command-line 3-D Scanner with DLP Structured Light SDK (http://www.ti.com/lit/zip/tidc535)
  - Camera Trigger Cable Assembly Guide (http://www.ti.com/lit/df/tidu457a/tidu457a.pdf)
- Available from TI Designs
  - http://www.ti.com/tool/TIDA-00254

Point Grey Research: http://ww2.ptgrey.com/USB3/Flea3



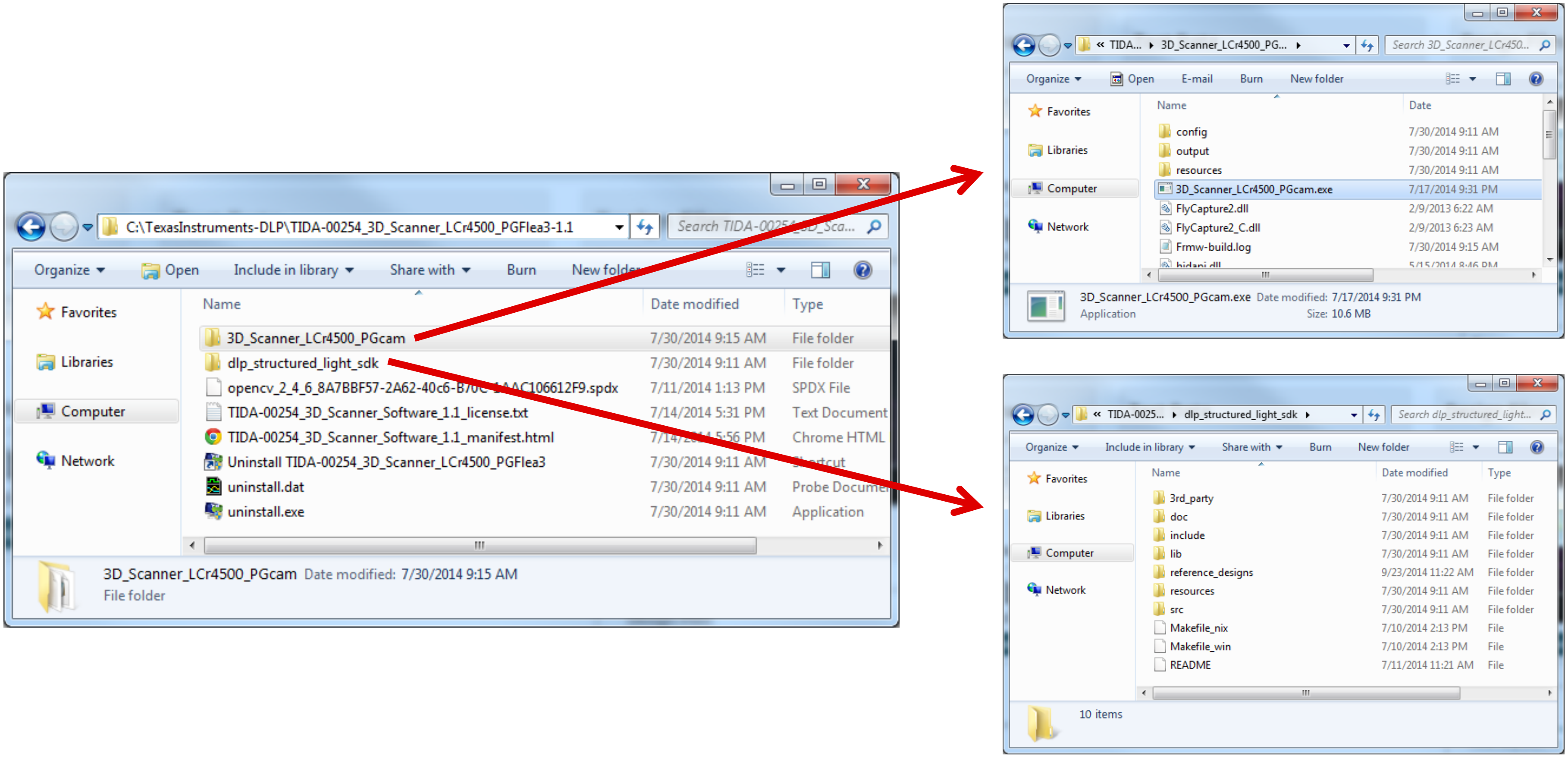
Where to find software, BOM, and trigger cable assembly guide?

1. Go to the TIDA-00254 tool page (http://www.ti.com/tool/TIDA-00254)
2. Software installer listed under Download Design Files button
3. Schematic and BOM listed under their respective buttons



Where is the 3-D Scanner Application and DLP Structured Light SDK source code?

1. Go to the TIDA-00254 tool page (http://www.ti.com/tool/TIDA-00254)
2. Software installer listed under Download Design Files button (http://www.ti.com/lit/zip/tidc535)



Design Considerations for 3-D Machine Vision

How to choose a DLP® Chipset?

- What am I trying to scan?
  - Darkly colored objects → Need more light
  - Lightly colored/shiny objects → Need less light
  - Moving objects → Need fast scan times
  - Static objects → Scan time less important
  - Biometrics → May need non-visible wavelengths
  - Clear objects → Need non-visible wavelengths
- How portable should scanner be?
  - Smaller chipsets need smaller optics and are more portable
- Are fast scan times needed?
  - Camera is typically the limiting factor if binary patterns are used
  - Non-binary patterns such as three phase methods can cause the projector speed to limit overall system
- What is the desired accuracy of the point cloud?
  - Determined by camera and projector resolutions, baseline distance, focal lengths, and structured light method

How to determine point cloud resolution?

Spatial X and Y

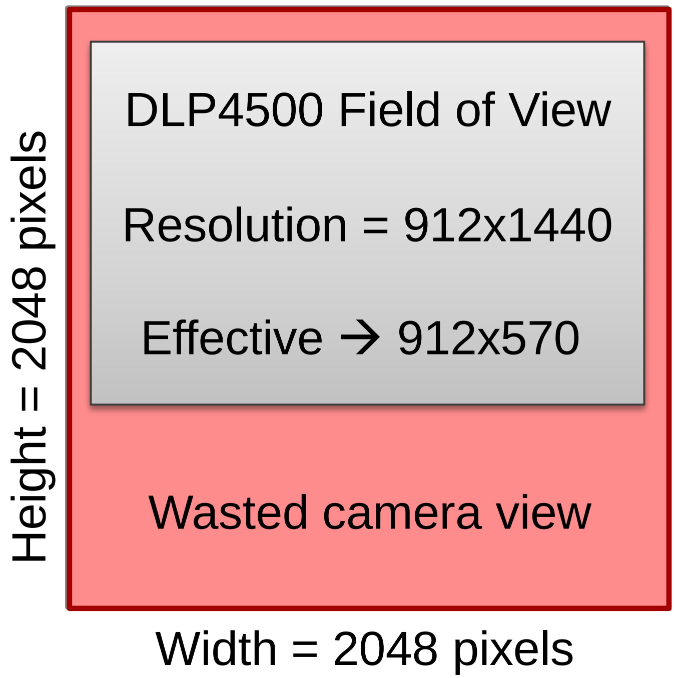
- Inversely proportional to field of view
  - Scanning larger areas worsens spatial resolution
  - Scanning smaller areas improves spatial resolution
- Proportional to camera and projector resolution
  - Increasing camera or projector resolution improves spatial resolution
  - Also increases number of points in cloud

Z-Depth

- Inversely proportional to focal length and baseline
  - Longer focal lengths improve accuracy
  - Increasing the baseline distance improves accuracy
- Proportional to the object distance and disparity resolution
  - Accuracy decreases as distance increases
  - Increasing camera and projector resolutions improves accuracy

System Resolutions & Field of Views

- Nyquist theorem requires at least 2x sampling
  - Camera width resolution must be double projector width resolution
  - Camera height resolution must be double projector height resolution
  - Camera pixel count should be at least 4 times larger than projector's!
- Field of view and "effective resolution" must be considered



Field of view mismatch means smaller effective resolution…

$$2048\, pxls * 60\% = 1228\, effective\, pxls$$

Check pixel sampling…

$$\frac{1228\, effective\, pxls}{1140\, projector\, pxls} = 1.07 < 2$$

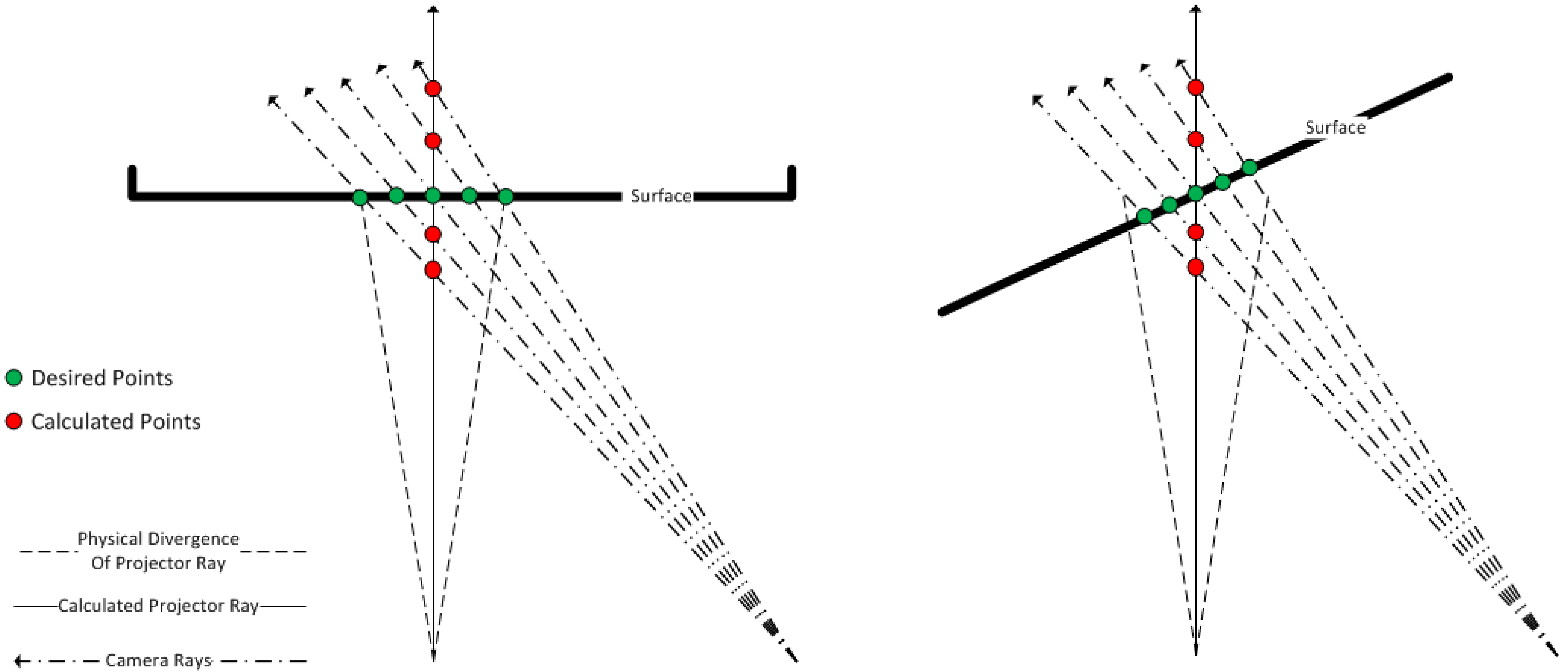
Cannot resolve all projector rows!!

$$\frac{1228\, effective\, pxls}{570\, projector\, pxls} = 2.15 > 2$$

Can resolve projector row pairs

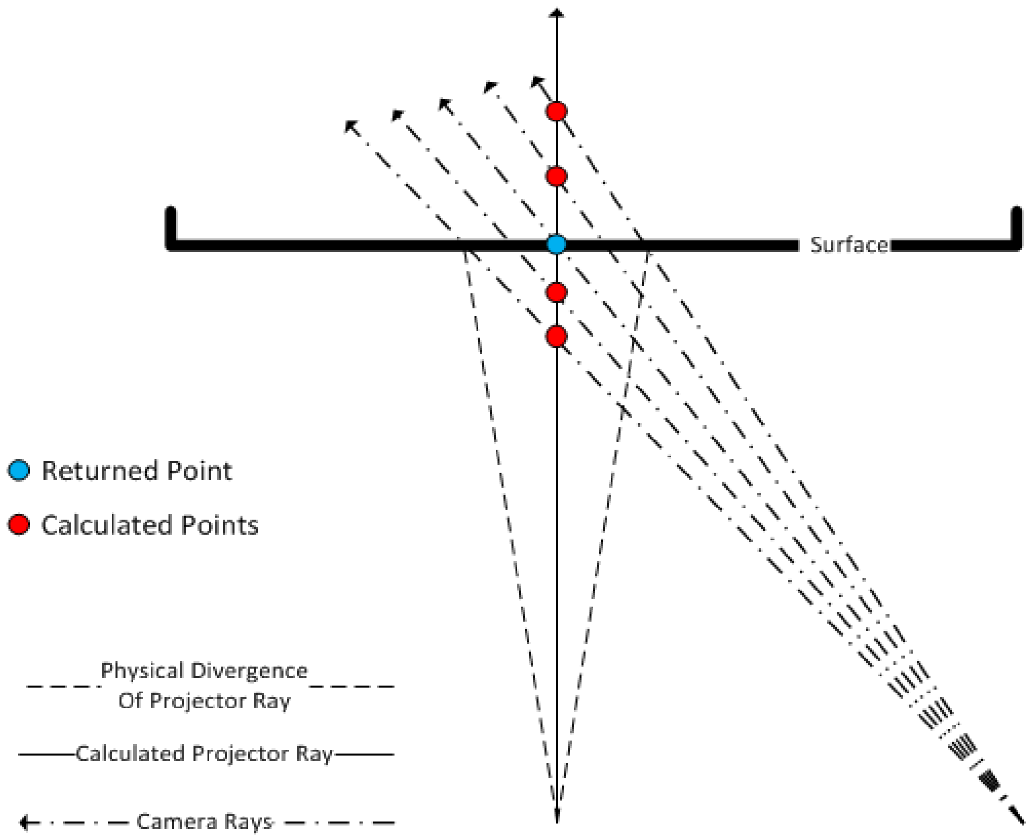
Effects of pixel over-sampling

- Decreased point cloud accuracy from divergent beams
  - Projector rays are straight lines rather than divergent beams
  - Over-sampling the projector rays leads to incorrect point reconstructions



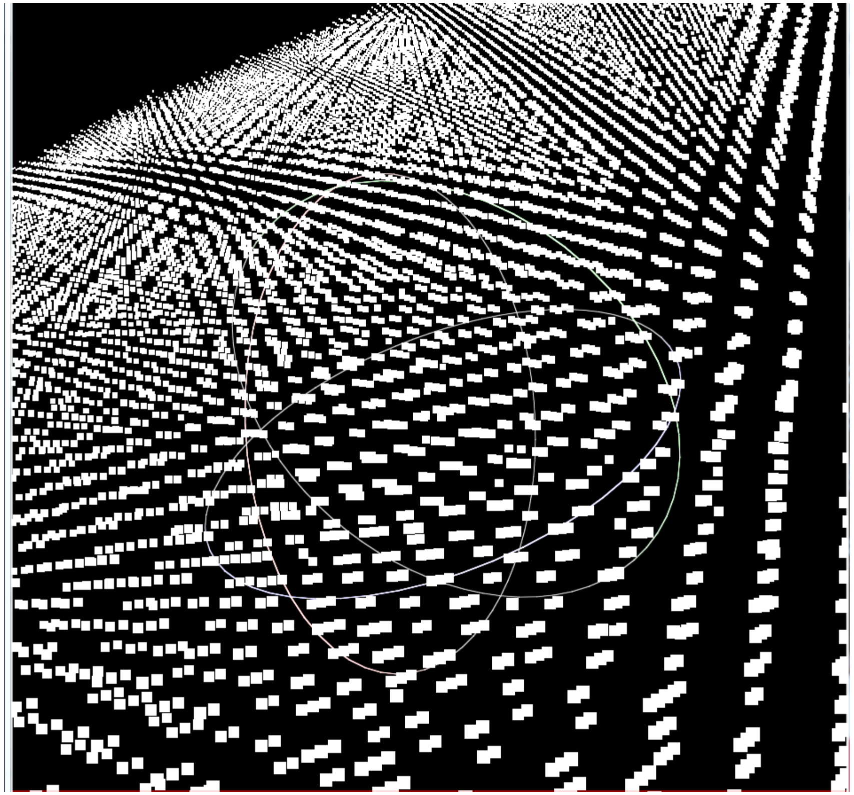
Overcoming pixel over-sampling

- When both vertical and horizontal patterns are used every camera ray is associated to a specific projector ray
- During reconstruction, group points according to their projector ray and filter!
- This method limits the number of points to the number of projector pixels
- This method will not work for scans with a single orientation of patterns

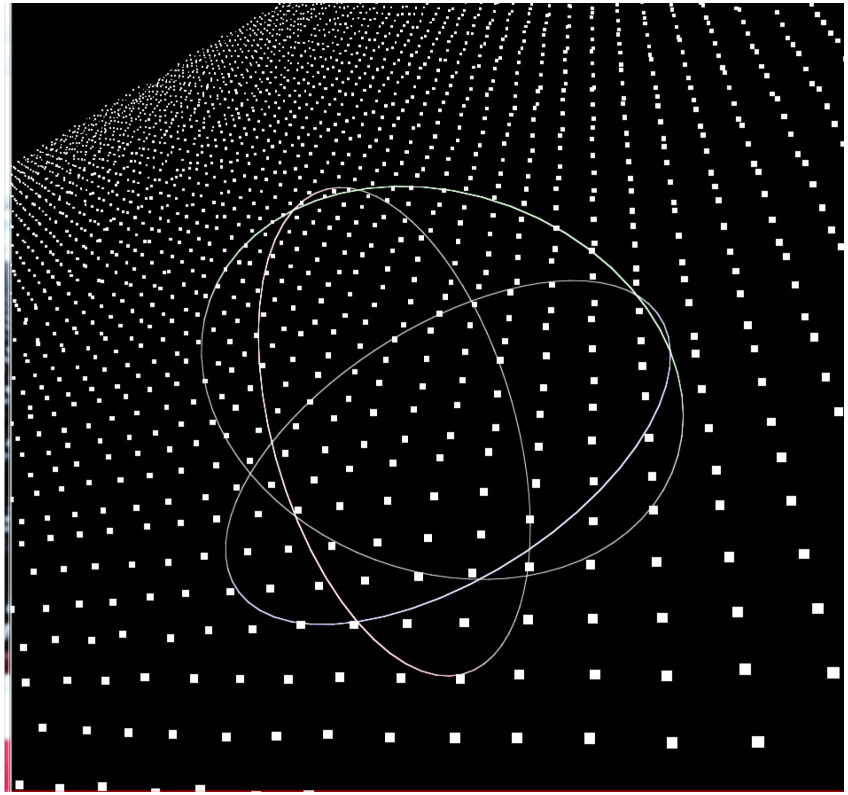


Example of filtering points by projector ray

No Filtering



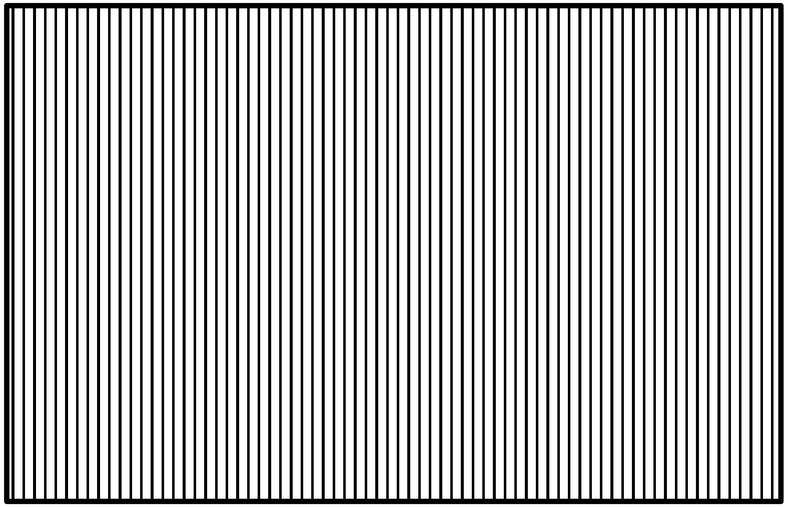
Points Filtered Per Projector Ray

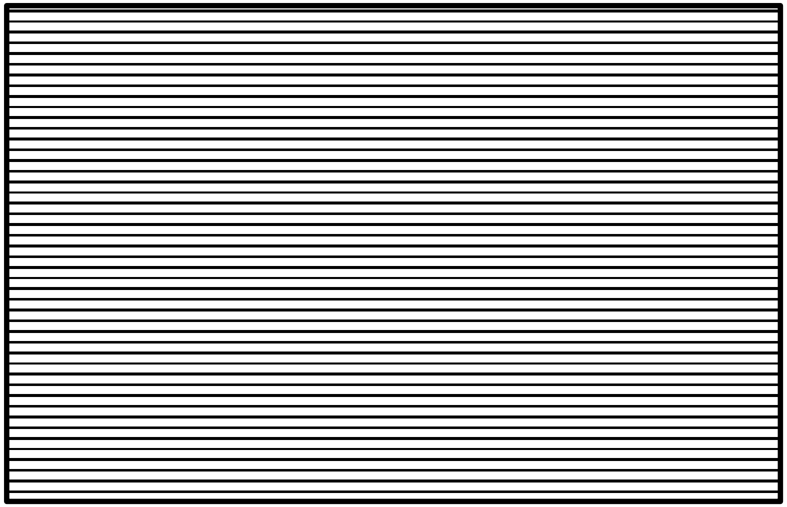


Note the lower density and higher accuracy of the point cloud

What about using one pattern orientation?

- If only vertical or horizontal patterns are used, filtering by projector ray is not possible since only planes are identified

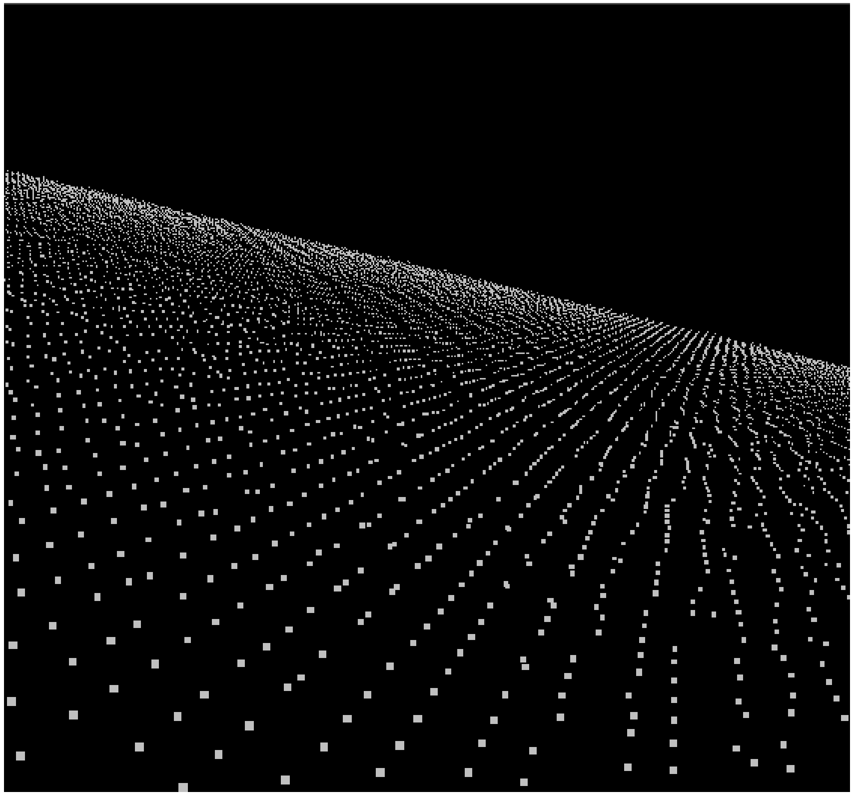




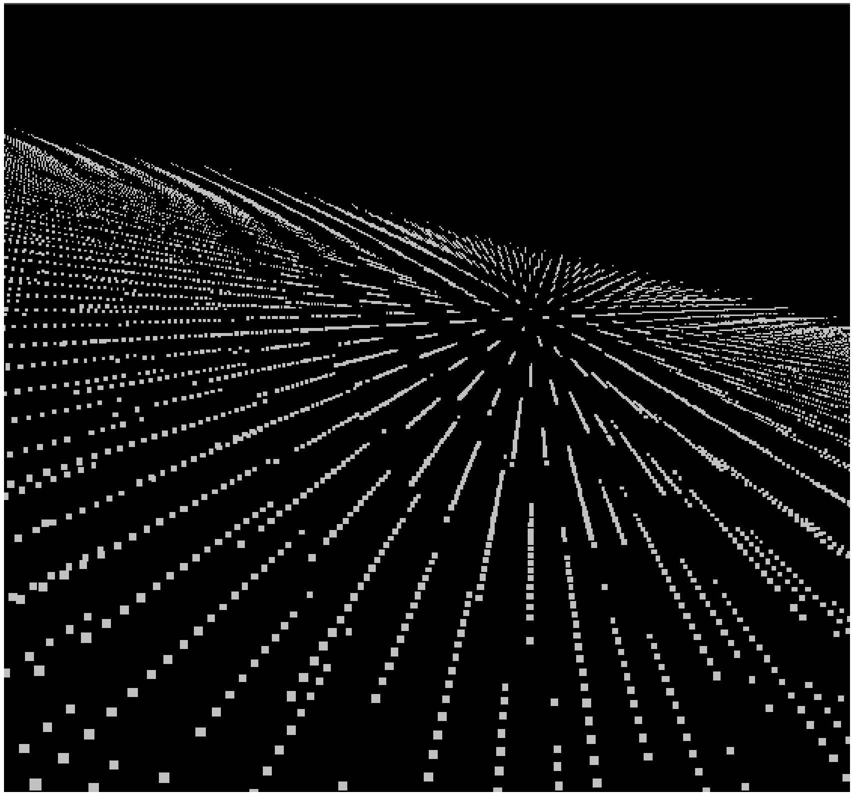
- Single pattern orientation scans offer several advantages
  - Faster scan times
  - Point cloud density a function of projector planes (rows or columns) and camera resolution rather than only projector pixels

Example of Pattern Orientation Clouds

Filtering w/ Both Orientations



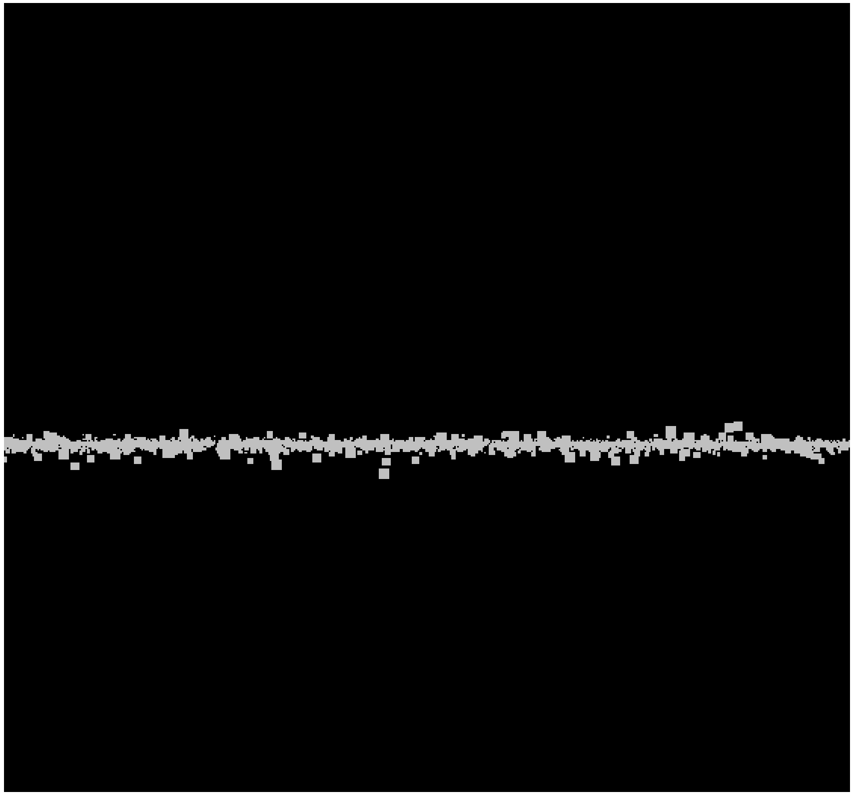
Only Vertical Orientation



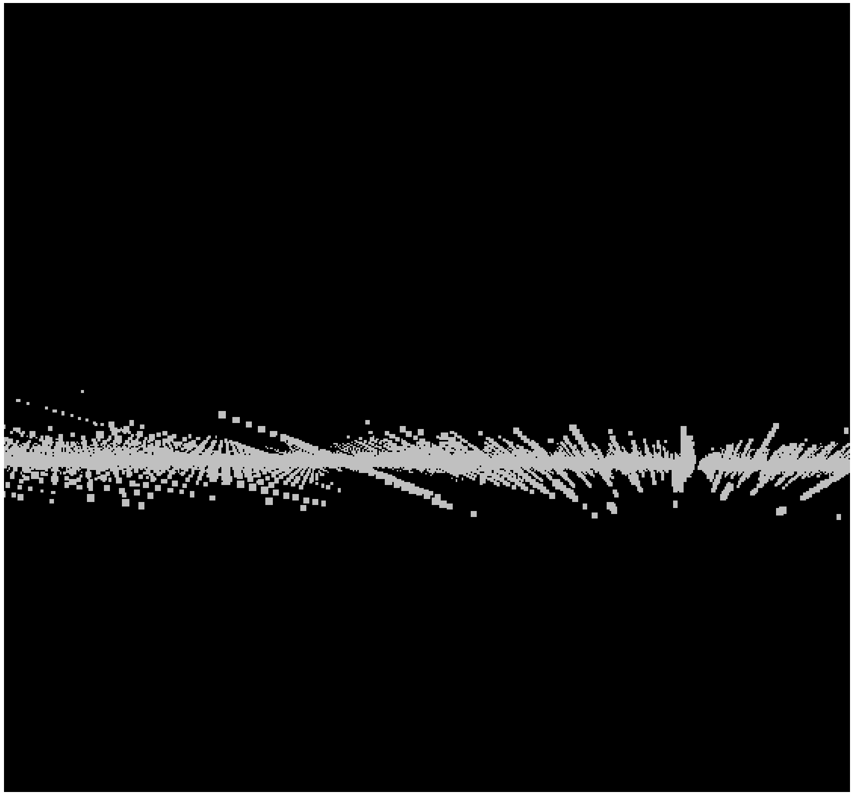
Note the higher point cloud density

Example of Pattern Orientation Clouds

Filtering w/ Both Orientations



Only Vertical Orientation

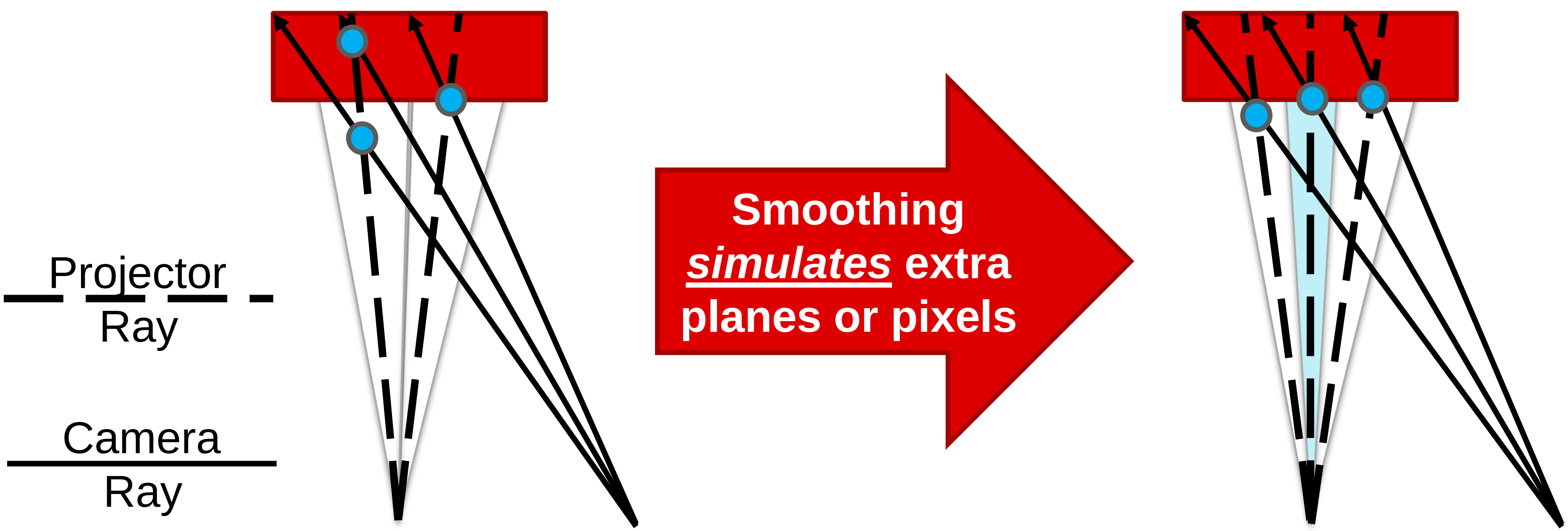


Note the higher point cloud density but lower depth accuracy

Smoothing the Disparity Map

- How to solve over sampling when using single orientation patterns?
  - After constructing the disparity map with projector planes, apply a smoothing filter on the disparity map
- Consider three camera pixels which see two projector planes

{ 1, 1, 2 } ←Disparity Map Values → { 1, 1½ , 2 }



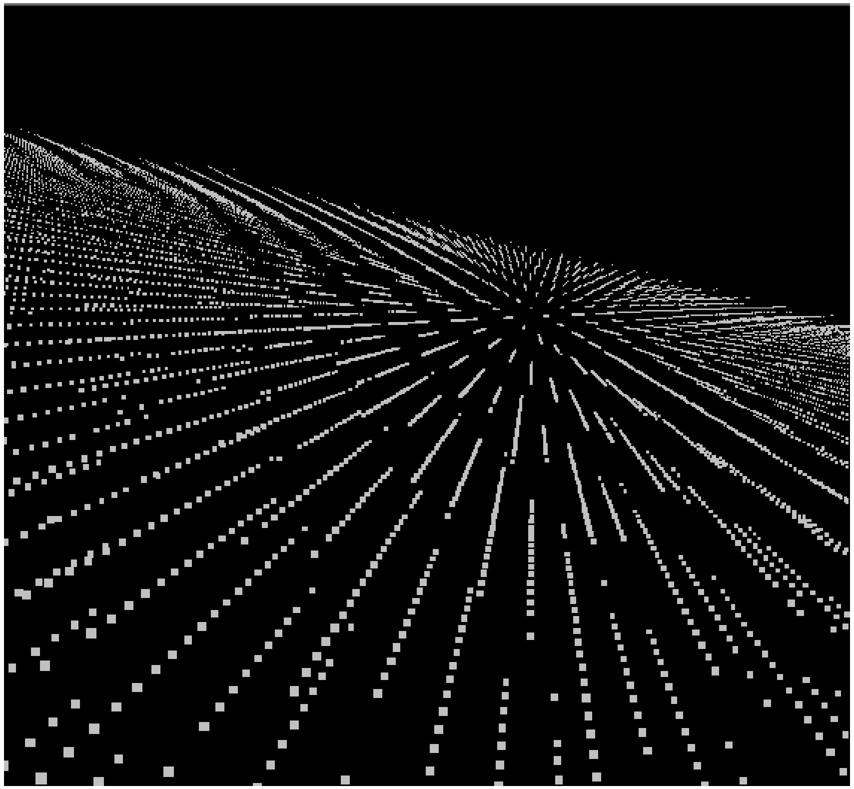
- Note: This can smooth edges that should be sharp!

Smoothing a Disparity Map Example

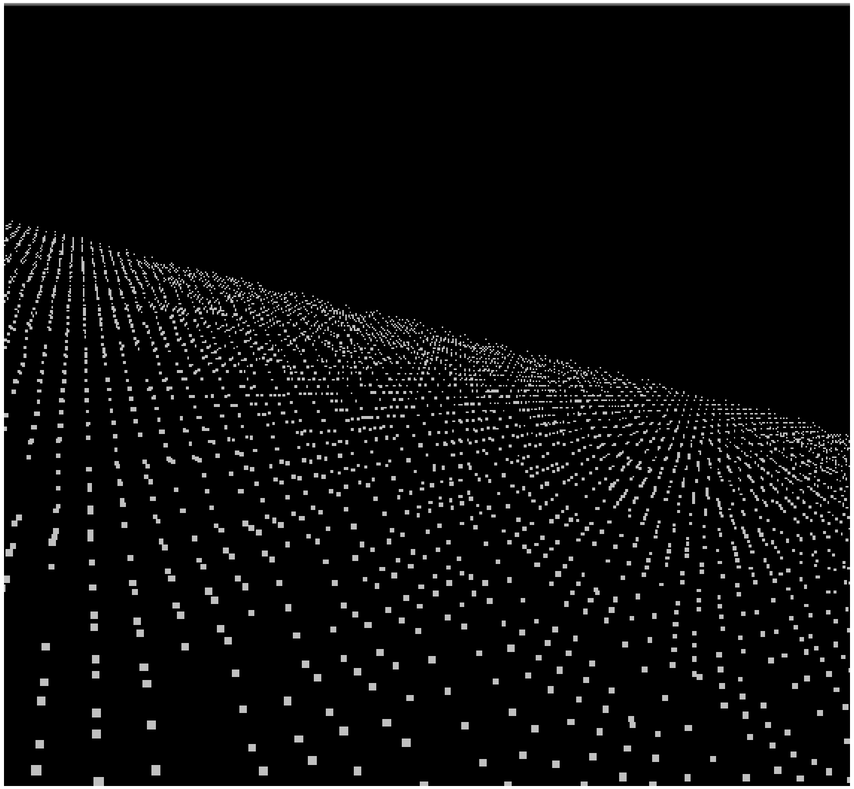
Note: This is an ***extreme*** example!

Example of Disparity Smoothing

No Smoothing



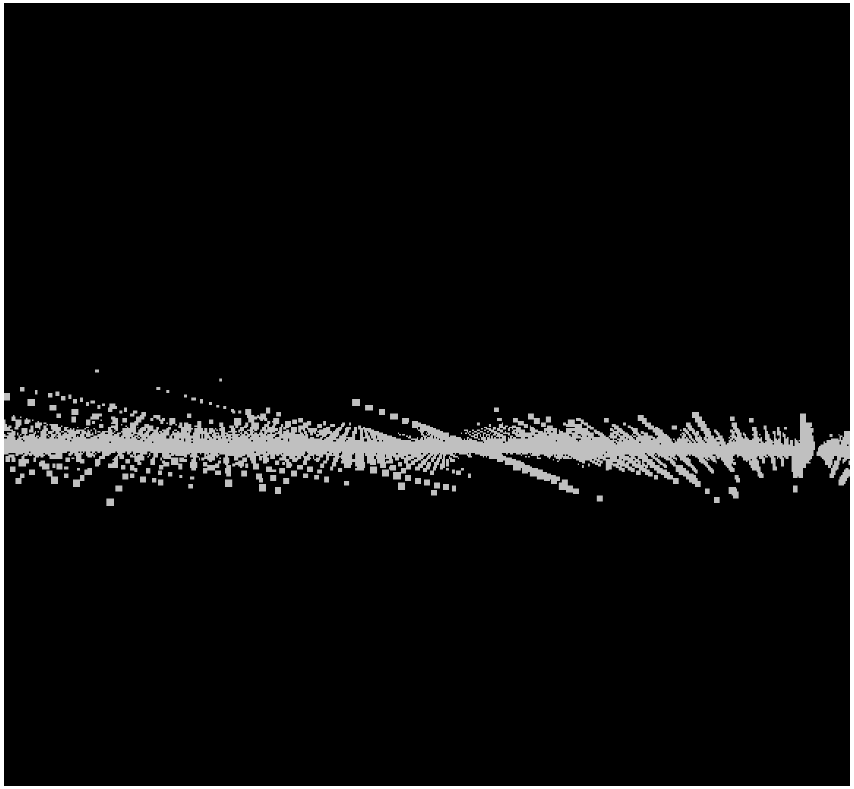
Smoothing Applied



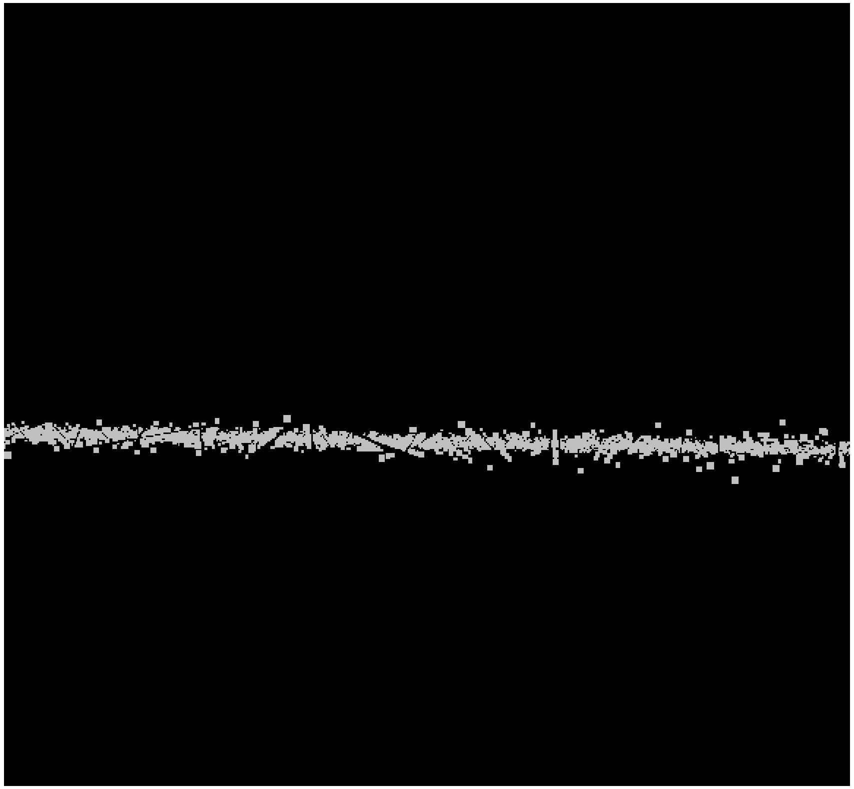
Helps but does not completely remove cloud "thickness"

Example of Disparity Smoothing

No Smoothing



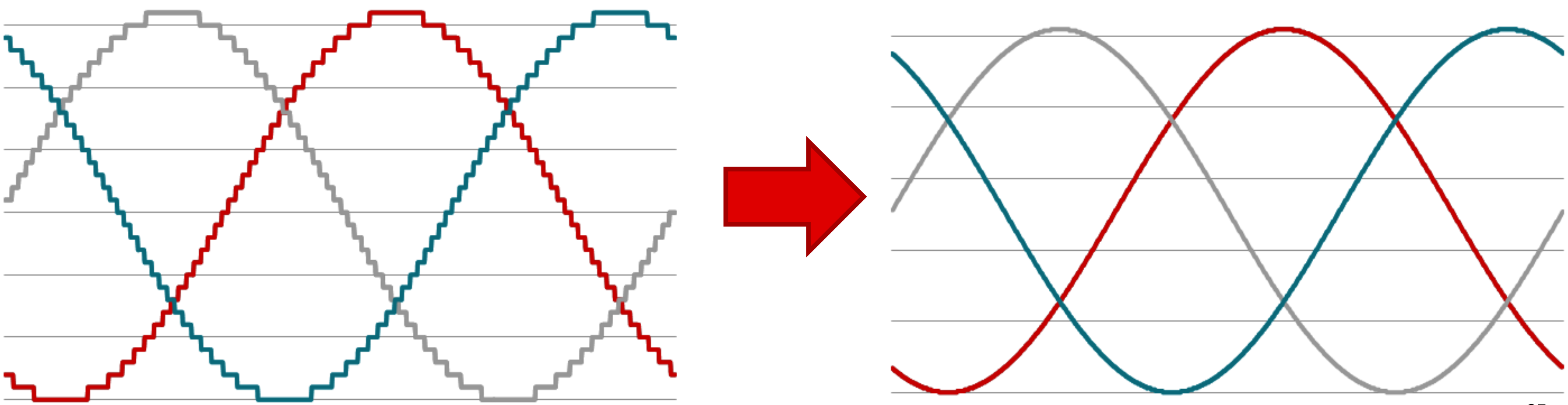
Smoothing Applied



Helps but does not completely remove cloud "thickness"

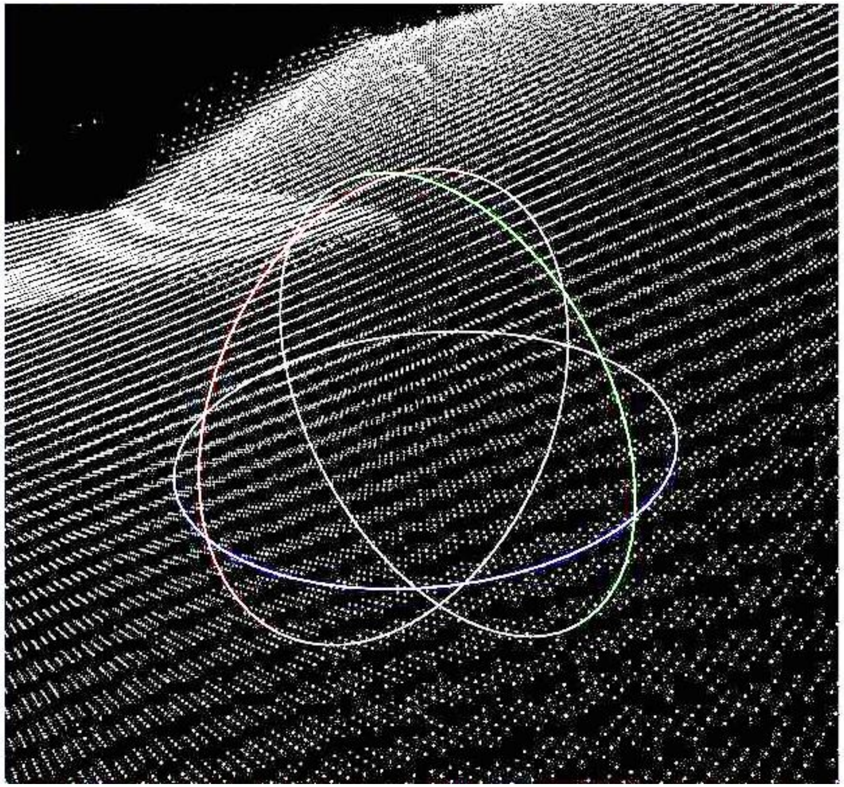
Alternative to Disparity Map Smoothing?

- Consider if the projector is in focus or not
  - Binary patterns are less robust against de-focus and “fuzzy” pixels are determined invalid during decoding
    - Smoothing disparity map helps, but softens edges due to manipulating the pixels of the captured image
- Phase shift patterns often times use sinusoidal patterns
  - Fuzzy pixels, over-sampling, and diamond pixels help smooth sine waves!
  - Edges are retained because camera *captures* the already averaged projection

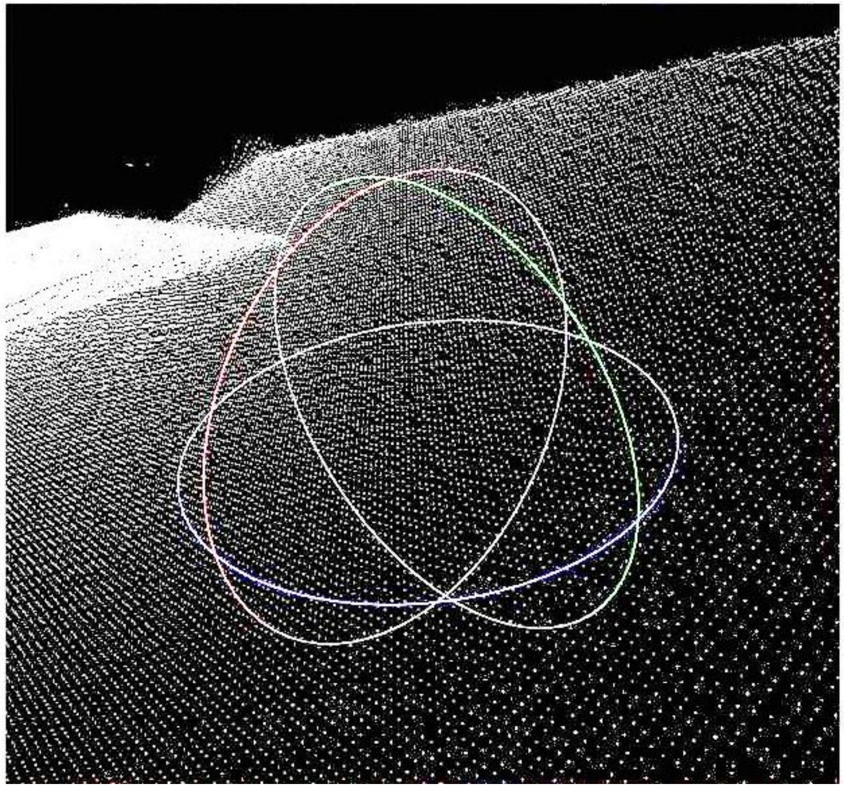


Example of Three Phase with Sub-Pixels

No Sub-Pixels



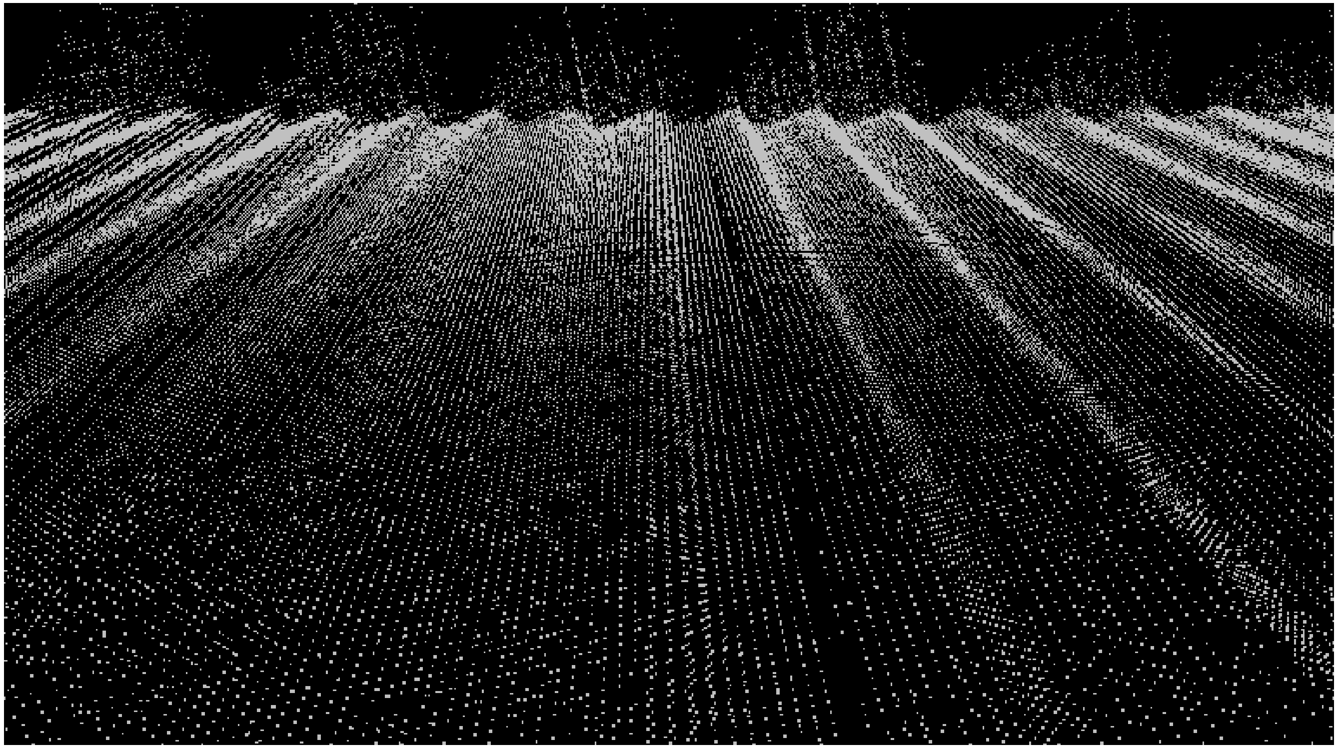
Sub-Pixels Calculated



No loss in point cloud density or edge definition!

Phase Shift Scans and Ambient Light

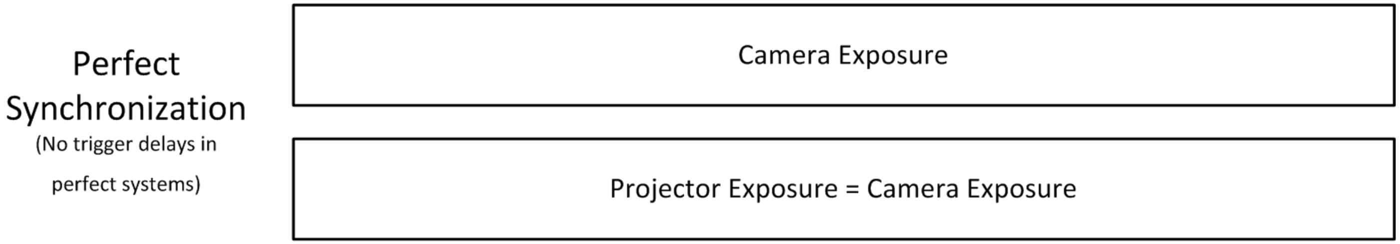
- Many indoor light sources pulse a 60 Hz!
- Notice the effect of fluorescent lighting on this 120 Hz speed scan



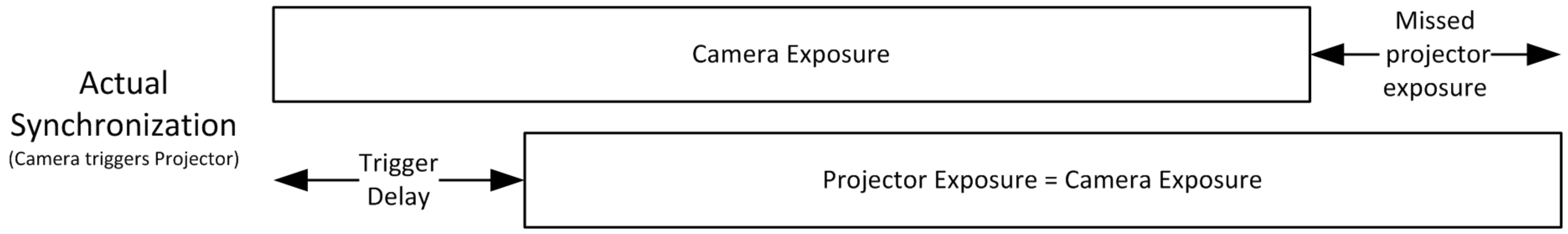
- Camera and pattern exposures must be very long to average the ambient light pulses or scan must be taken in darkness

Synchronizing the Camera with Projector

- Ideally everything matches up perfectly



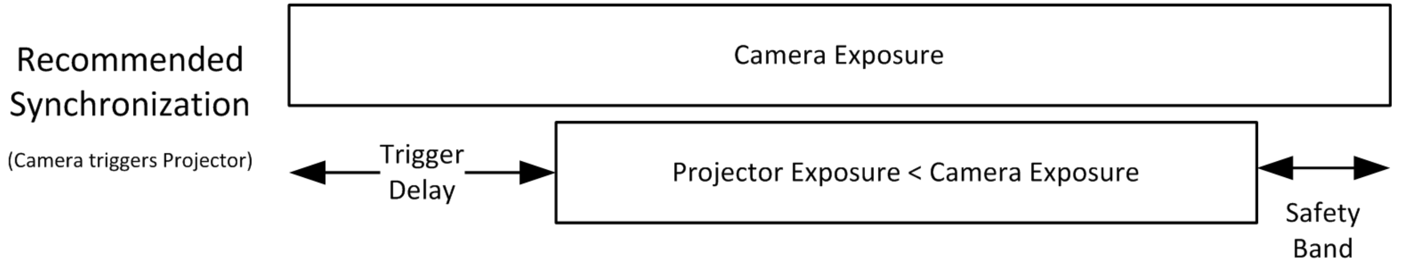
- There is *<u>always</u>* a delay in the triggers



  - Camera exposure may capture part of the next or previous projected pattern
  - Delay could cause the camera exposure to miss part of the pattern
    - This could cause issues in the linearity of non-binary patterns

Synchronizing the Camera with Projector

- Recommended synchronization setup



- Camera triggers projector
  - Most cameras with triggers run faster in a free run mode rather than a triggered mode
  - If the projector triggered the camera, the trigger delay would cause the camera exposure to miss part of the exposed pattern
- Projector exposure shorter than the camera exposure
  - Ensures the camera exposure captures the entire projected pattern and thus greyscale linearity remains intact

Developing with the DLP® Structured Light SDK

Developing the DLP® Structured Light SDK

- Setting up the development environment
  - Install the DLP Structured Light SDK
  - Install and compile OpenCV
- Scalable solutions with object oriented programming (OOP) in C++
  - Case Study: Consider that the 3-D Scanner Demo software is practically identical for the DLP LightCrafter™ 4500 EVM and DLP LightCrafter 6500 EVM (and now the DLP LightCrafter 3000 EVM)
  - What are the primary abstract modules?
  - How to use abstracted modules?
- Creating new modules
  - Where should source code go?
  - How to creating a new camera module?

Setting up the development environment

- DLP® Structured Light SDK
  - Currently included with 3-D Machine Vision Reference Design
  - As new DLP evaluation modules are added (DLP LightCrafter, LightCrafter 6500 EVM, etc.) the DLP Structured Light SDK source code will move to its own tool page
    - This is to prevent duplication of code on ti.com
- OpenCV
  - Download the source code from www.opencv.org
  - Brief instructions
    - Install CMake
    - Use CMake to create Makefile
    - Compile with make
  - Detailed compilation instructions are available in the Machine Vision Reference Design User's Guide

Scalable Solutions with OOP

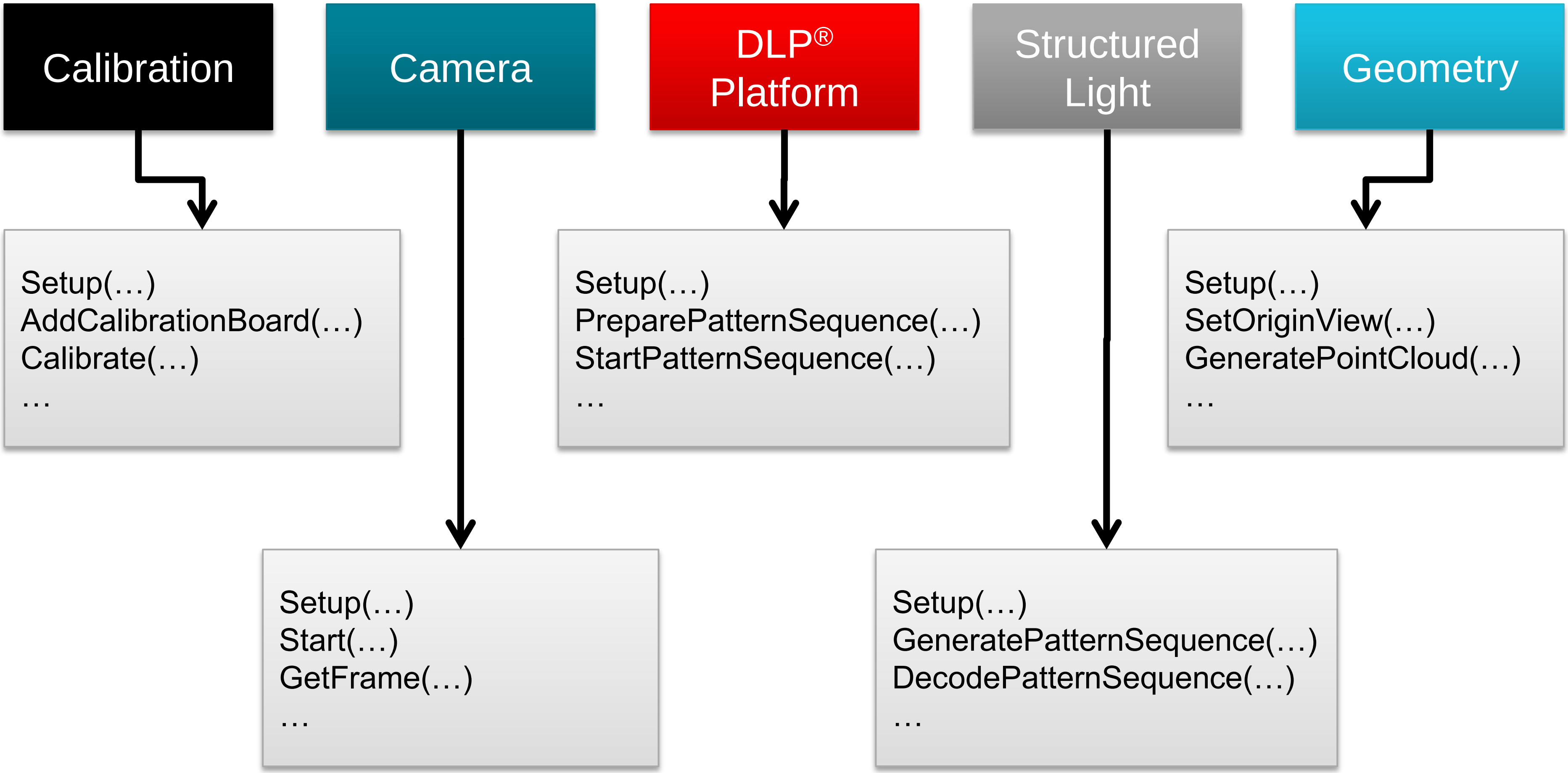
- Consider the DLP LightCrafter™ 4500 and LightCrafter 6500 EVMs
  - Both chipsets have different API, resolutions, speeds, etc.
  - Did I need to rewrite each application with all of the chipset specific API ?
    - NO! Only a single line of code needed to change

```
// System Variables
dlp::PG_FlyCap2_C    camera;
dlp::LCr4500         projector;
```

```
// System Variables
dlp::PG_FlyCap2_C    camera;
dlp::LCr6500         projector;
```

  - **The DLP LightCrafter 3000 & 6500 EVM SDK to be added soon!**
- How is this possible?
  - DLP Structured Light SDK contains modules which define interfaces
  - C++ allows you to reference sub-classes as their parent class

What are the primary abstract modules?



- Each module (base-class) defines an interface which all sub-modules (sub-classes) must follow

How to use abstracted modules?

- Function Declaration uses the abstracted base-class modules

```
void ScanObject(dlp::Camera                *camera,
                const std::string          &camera_calib_data_file,
                dlp::DLP_Platform          *projector,
                const std::string          &projector_calib_data_file,
                dlp::StructuredLight *structured_light_vertical,
                dlp::StructuredLight *structured_light_horizontal,
                const bool                 &use_vertical,
                const bool                 &use_horizontal,
                const std::string          &geometry_settings_file){
```

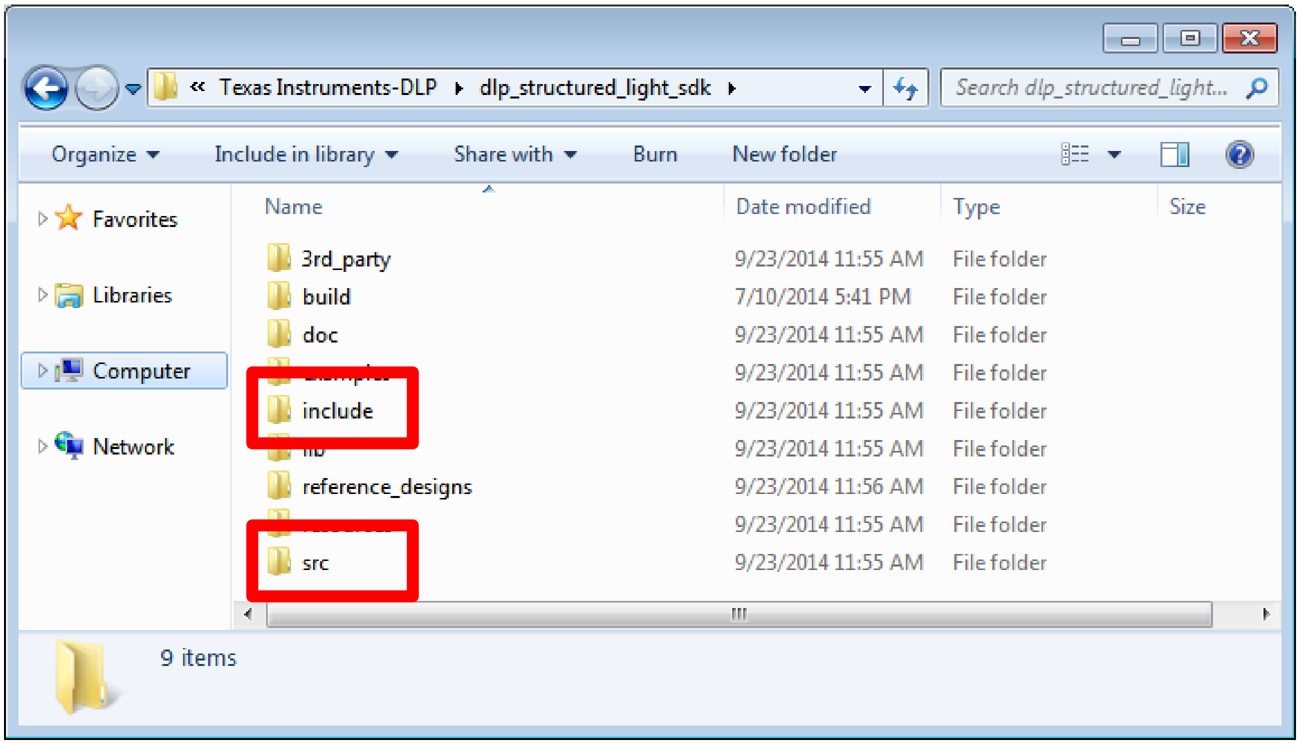
- Specific sub-modules can be passed as the parent type though

```
// System Variables
dlp::PG_FlyCap2_C   camera;
dlp::LCr6500        projector;

ScanObject(&camera,
            calibration_data_file_camera,
           &projector,
            calibration_data_file_projector,
           &structured_light_vertical,
           &structured_light_horizontal,
           true,
           true,
            geometry_settings_file);
```

Where should source code go?

- Use the current sub-modules for reference
  - Header files should be located in the /include directory
  - Source files should be located in the /src directory



- Add new source files to QT PRO file or CMakeLists.txt

How to creating a new camera module?

- Reference the module base-class header files to identify what functions need to be written for a sub-class
  - All virtual functions must be written by the sub-class!

```
class Camera: public dlp::Module{
    public:

    // Define by subclass
    virtual ReturnCode Connect(int camera_id) = 0;
    virtual ReturnCode Disconnect() = 0;

    virtual ReturnCode Start() = 0;
    virtual ReturnCode Stop() = 0;

    virtual ReturnCode GetFrame(Image* ret_frame) = 0;
}
```

NOTE: This is only an example!! Please reference camera.hpp for the complete camera module declaration

```
class PG_FlyCap2_C : public Camera
{
public:

    // Define pure virtual functions
    ReturnCode Connect(int camera_id);
    ReturnCode Disconnect();

    ReturnCode Start();
    ReturnCode Stop();

    ReturnCode GetFrame(Image* ret_frame);
}
```

Notice that the sub-class contains the exact same methods

Conclusion

- Industrial 3-D Machine Vision systems are complicated as a whole, but the different processes can be broken down into manageable sub-systems
- The TI design featuring the DLP® LightCrafter™ 4500 EVM enables faster evaluation periods for assessing DLP technology with 3-D machine vision (about 30 minutes!)
- Choosing the best DMD and other system factors is completely dependent on the application
- The DLP Structured Light SDK provides a scalable and easy to use framework for developing various applications using DLP technology

Questions

Live Demonstration!

Steps to use to 3-D Machine Vision design

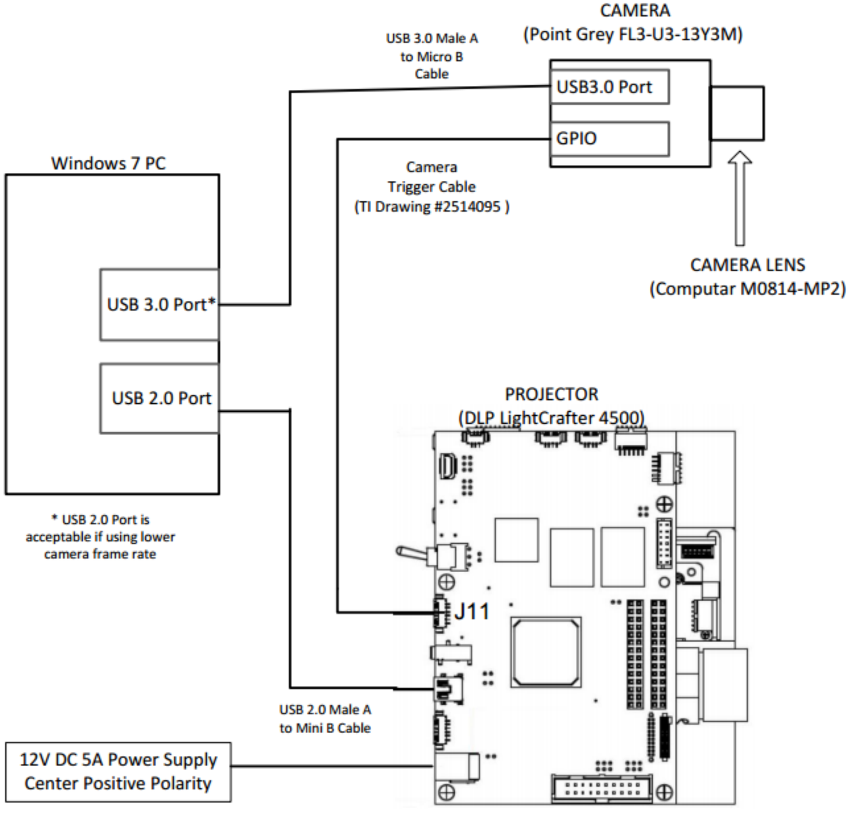
1. Install software → Done!
2. Connect the hardware
3. Make calibration board → Done!
4. Prepare the software and projector
5. Calibrate Camera
6. Calibrate System
7. Perform Scan
8. Develop your own applications!

Installing software

Software	Purpose	Required for application	Required for development
Point Grey FlyCapture2 SDK http://www.ptgrey.com/	• Camera drivers • Camera interface code libraries	☑	☑
DLP Machine Vision Reference Design Download http://www.ti.com/lit/zip/tidc535	• Precompiled application • DLP Structured Light SDK	☑	☑
Meshlab http://meshlab.sourceforge.net/	• Point Cloud Viewer • Surface Reconstruction • Analysis	(optional)	
OpenCV (source code) http://opencv.org/	• Image data management and processing code libraries		☑

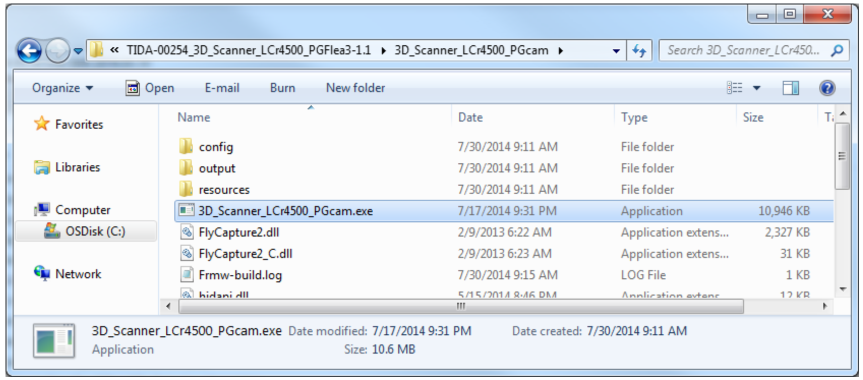
Connecting the Hardware

- Connect Camera to USB3.0 port if available
- Connect DLP® LightCrafter™ 4500 EVM to any USB port
- Connect Camera trigger cable to DLP LightCrafter 4500 EVM input trigger

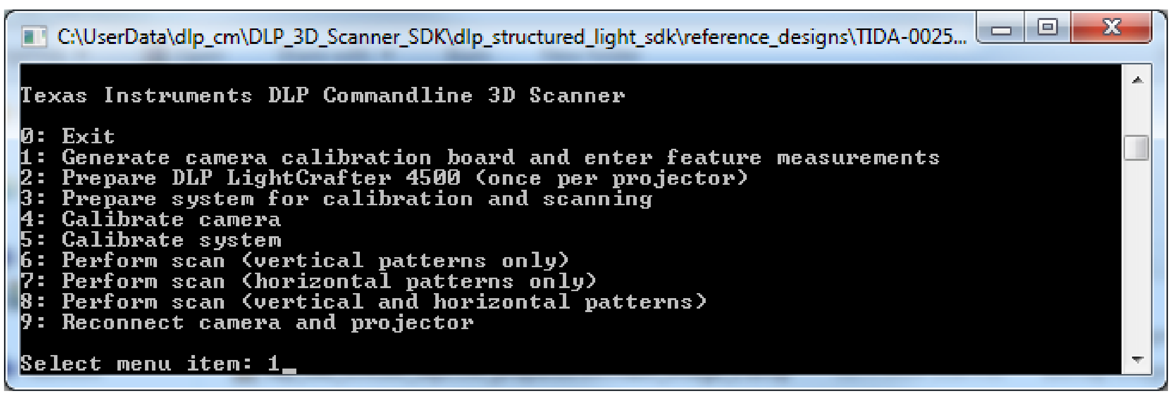


How to make the calibration board

- Open application directory and start executable

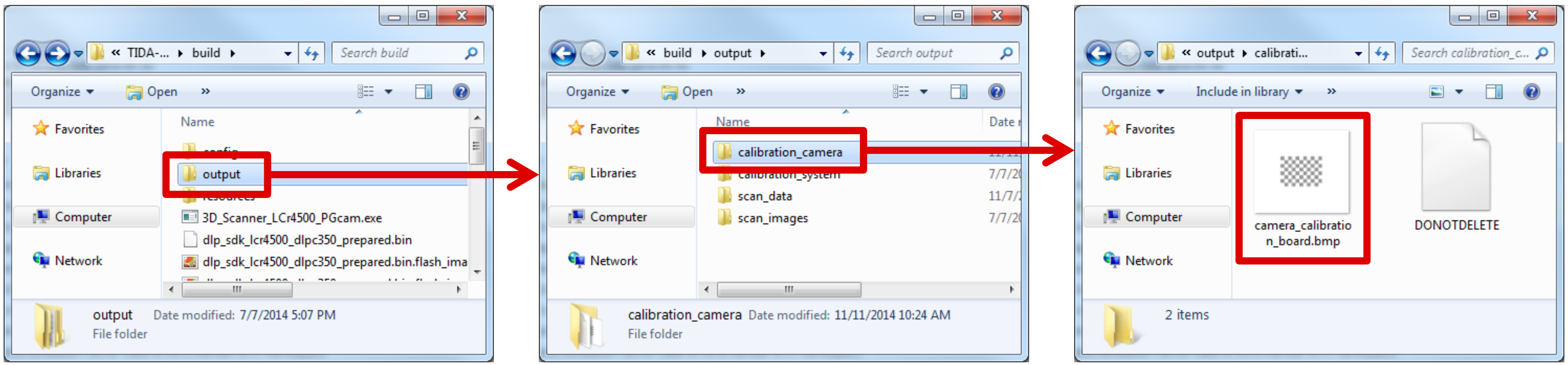


- Enter menu item “1: Generate camera calibration board and enter feature measurements”



How to make the calibration board

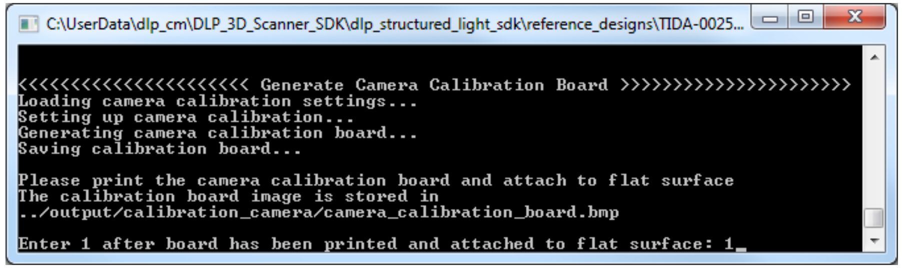
- After selecting menu item 1, a BMP file with the chessboard is generated in the “output/calibration_camera” directory



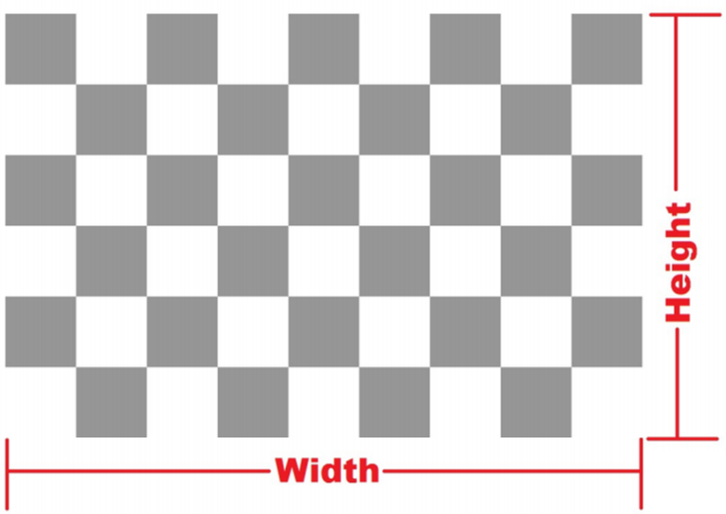
- Print the BMP file (at high DPI) and attach it to a flat surface
  - ¼” Foam core board, aluminum sheet stock, etc. all work well
  - Use spray adhesive to attach printed chessboard
  - **Your point cloud data will only be as good as your calibration board! Flatness is critical!**

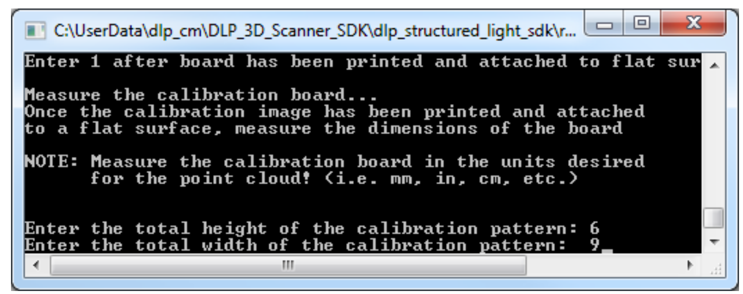
Entering calibration board measurements

- Enter “1” after printing and attaching the flat board



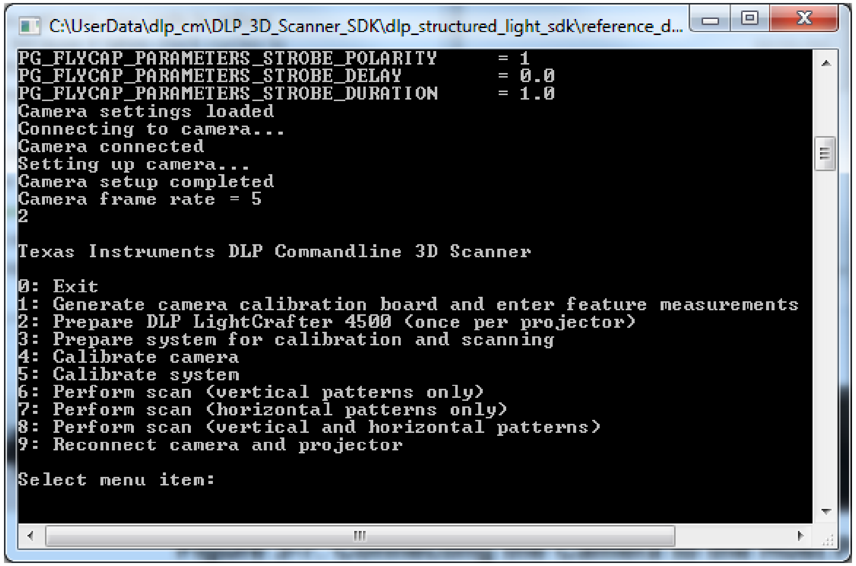
- Measure and enter the height and width of the calibration pattern
  - Note: Point cloud data units will be in the same units as are entered here





Preparing software and projector

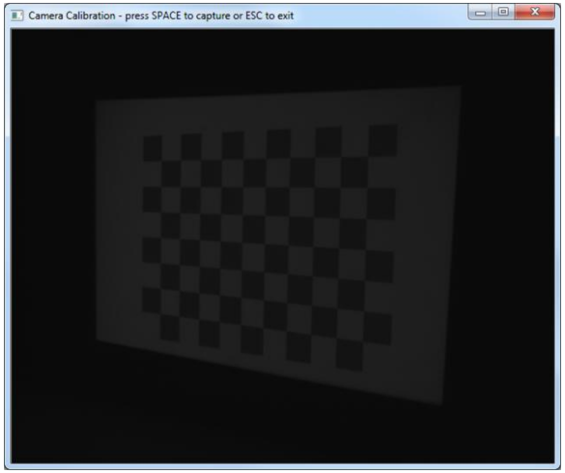
- Preparing the software and projector does the following:
  - Loads calibration and structured light settings
  - Generates projector calibration pattern
  - Generates structured light patterns
  - Uploads images to LightCrafter 4500
- The first time you use the projector with the software or change any structured light settings, use option 2: “Prepare DLP LightCrafter 4500 (once per projector)”
  - Performs all steps listed above
- If settings have not changed and the projector was previously prepared, use option 3: “Prepare system for calibration and scanning”
  - Performs all steps above, except uploading images to LightCrafter 4500
  - Must be run every time the application is run



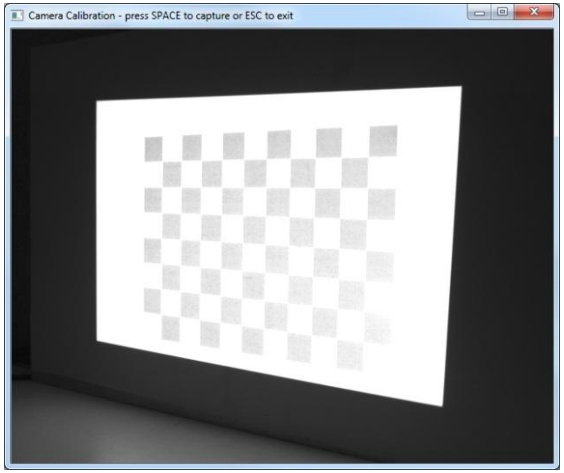
Calibrating the Camera - Setup

- Before capturing any board positions, set the aperture and focus
  - Aperture determines how much light reaches the sensor
  - Focus ensures the image plane is at the exact level of the sensor so that the image is sharp and not blurry
  - Lock everything into place!

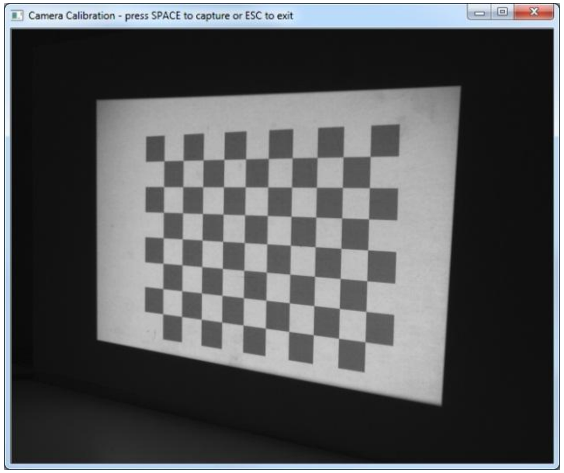
Under exposed



Over-exposed

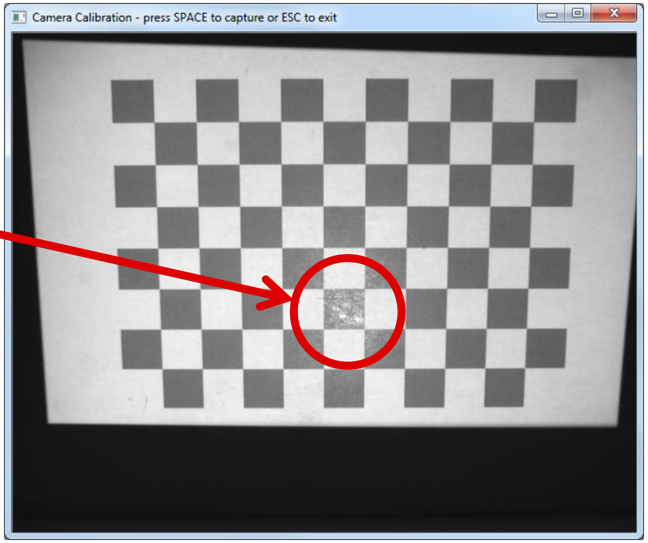


Good exposure

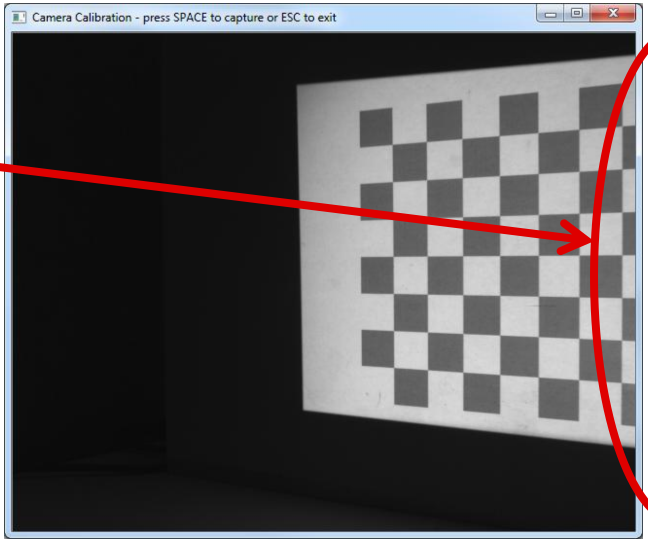


Calibrating the Camera – Watch out for…

- Software won't find the chessboard if…
  - There is too much glare
    - To remove glare, angle the calibration board

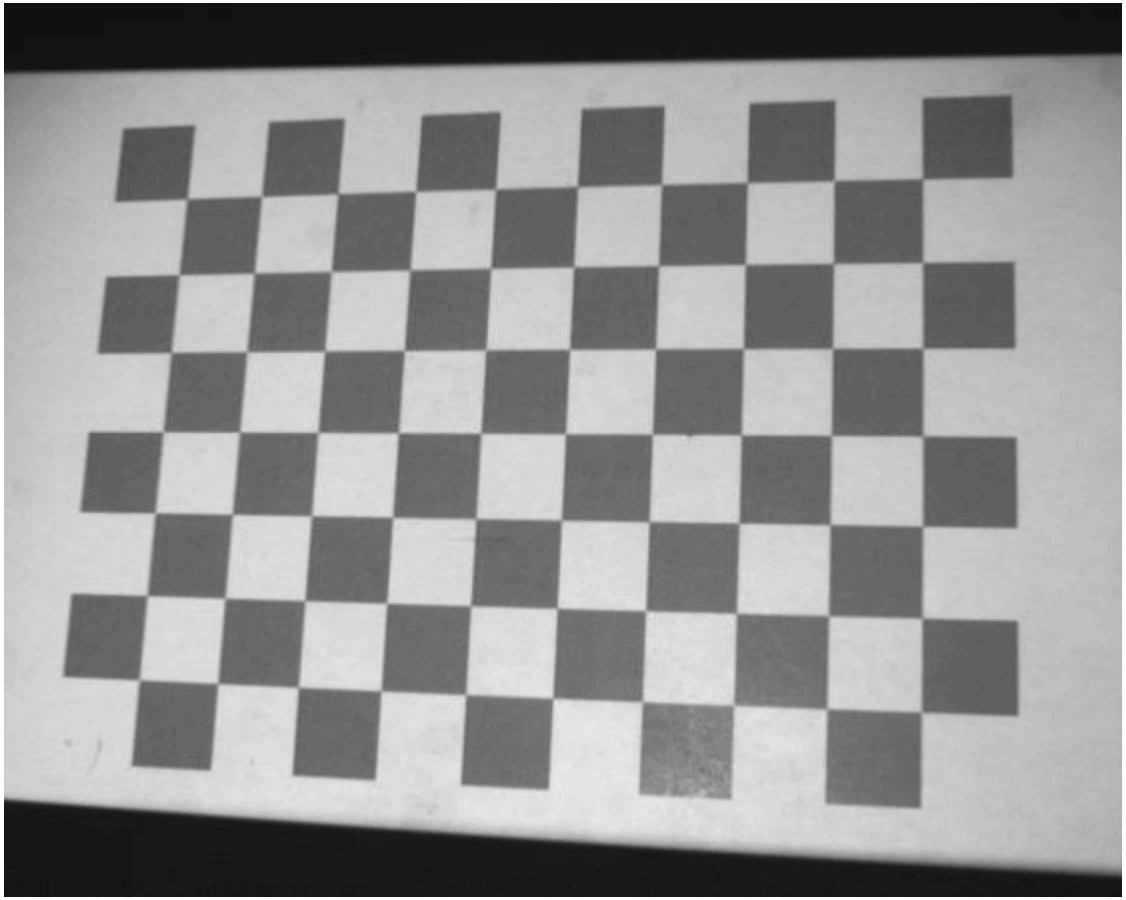


  - Part of the chessboard is missing from within the captured image
    - Parts of the squares on the border square can be cutout, so long as the inside corners are still visible



Calibrating the Camera – Example Images

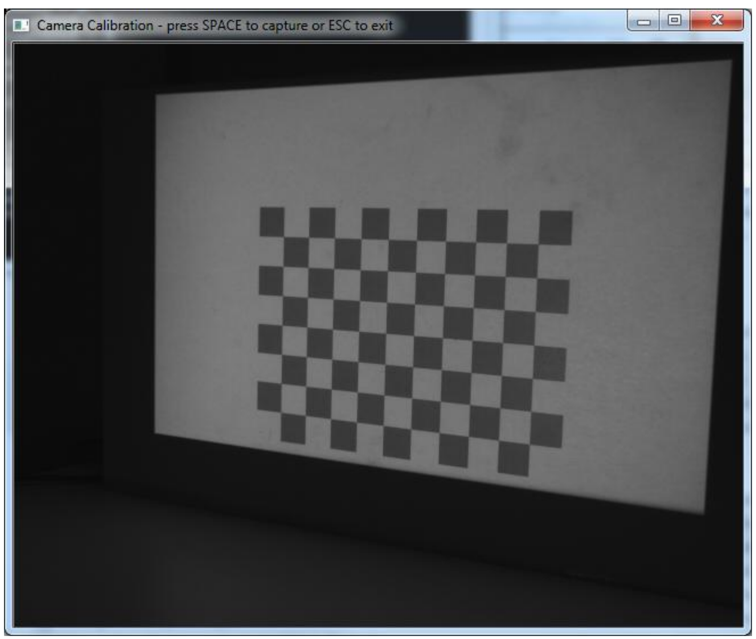
- Calibration image examples
- ?????? Measured camera reprojection error = 0.166341 ??????



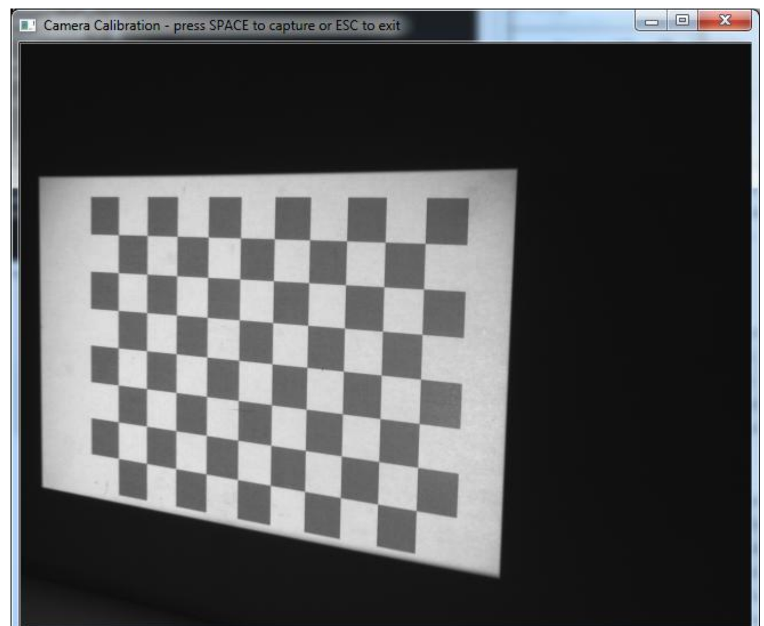
Calibrating the system - Setup

- Mount the camera so that the projected area can be seen within the camera at the minimum and maximum scanning distance
  - Try to utilize the entire camera frame if possible
- If the camera or projector are moved relative to each other, this calibration process must be redone

Furthest distance

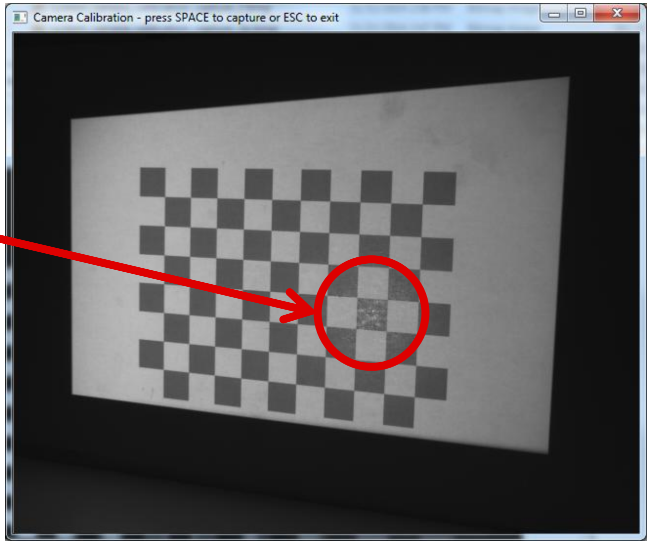


Closest distance

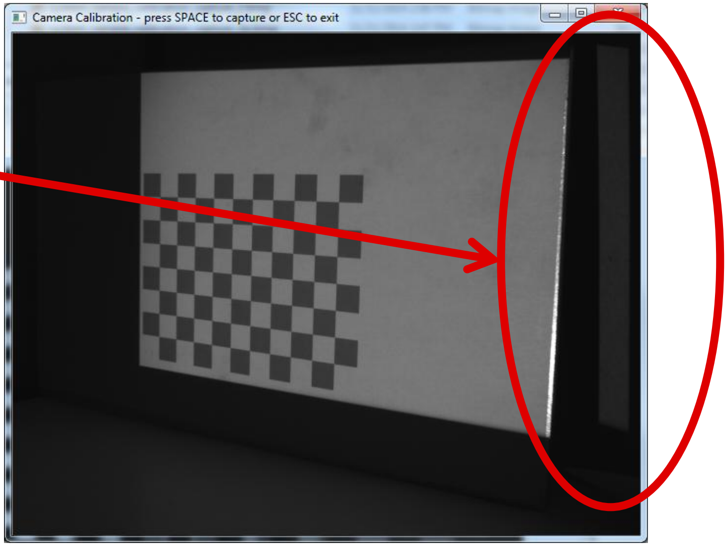


Calibrating the system – Watch out for…

- Software won't find the chessboard if…
  - There is too much glare
    - To remove glare, angle the calibration board

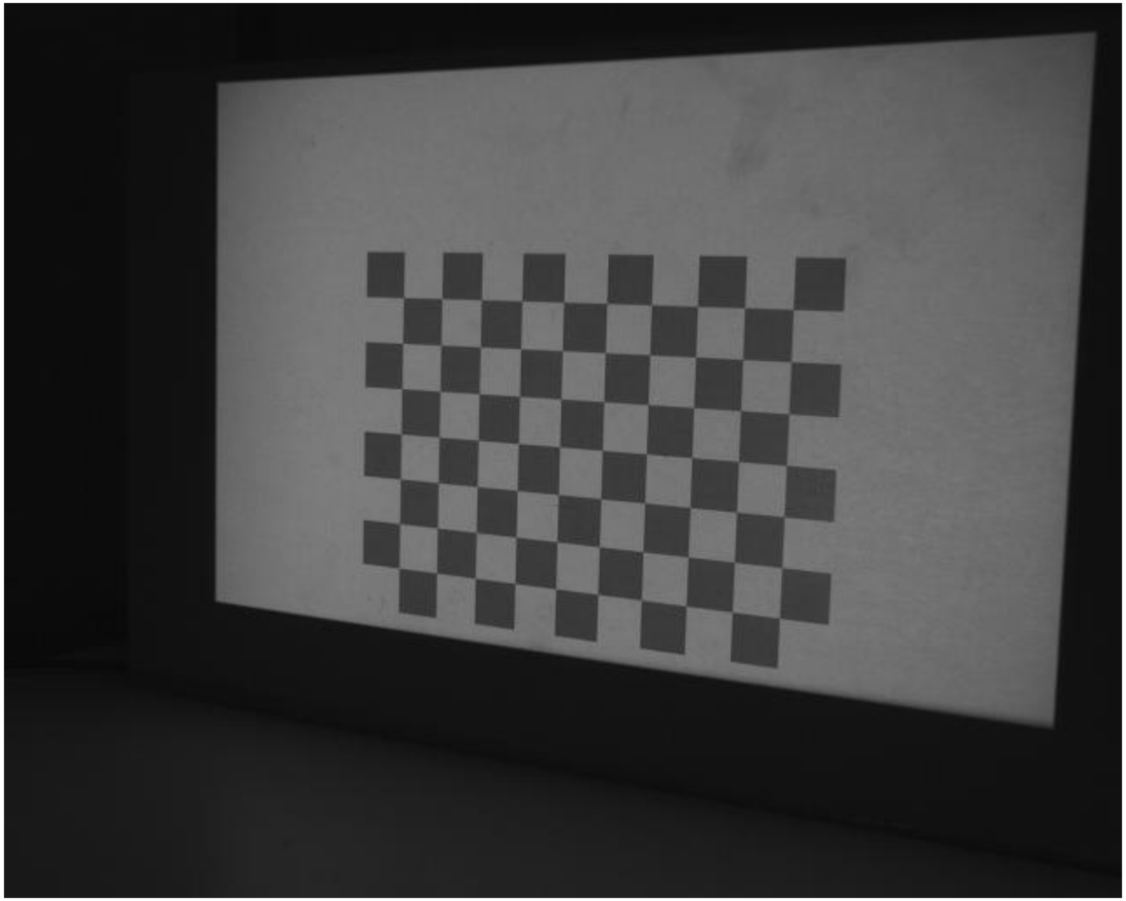


  - Part of the projected image falls off of the calibration board
    - This will cause squares to be missing on the projected chessboard calibration pattern



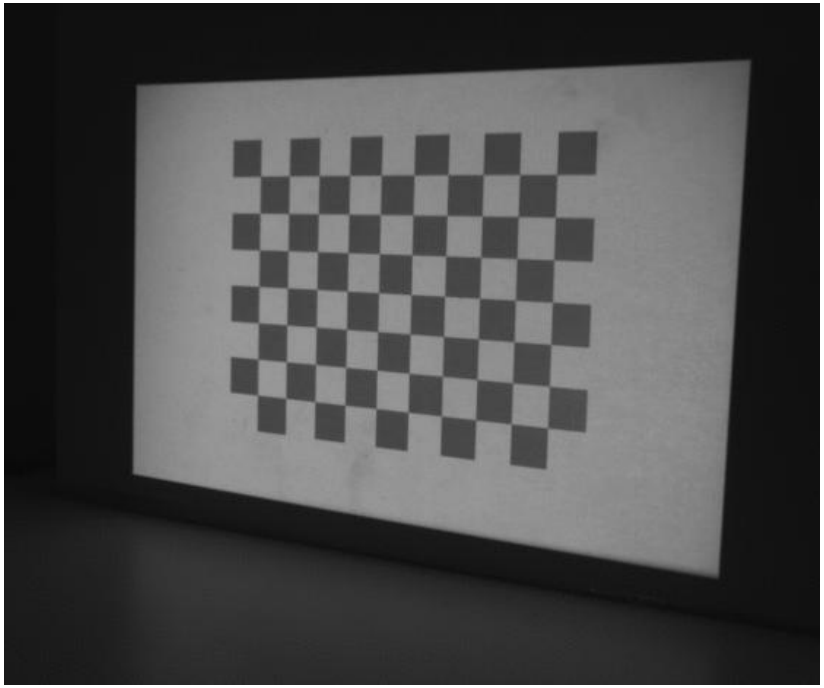
Calibrating the System – Example Images

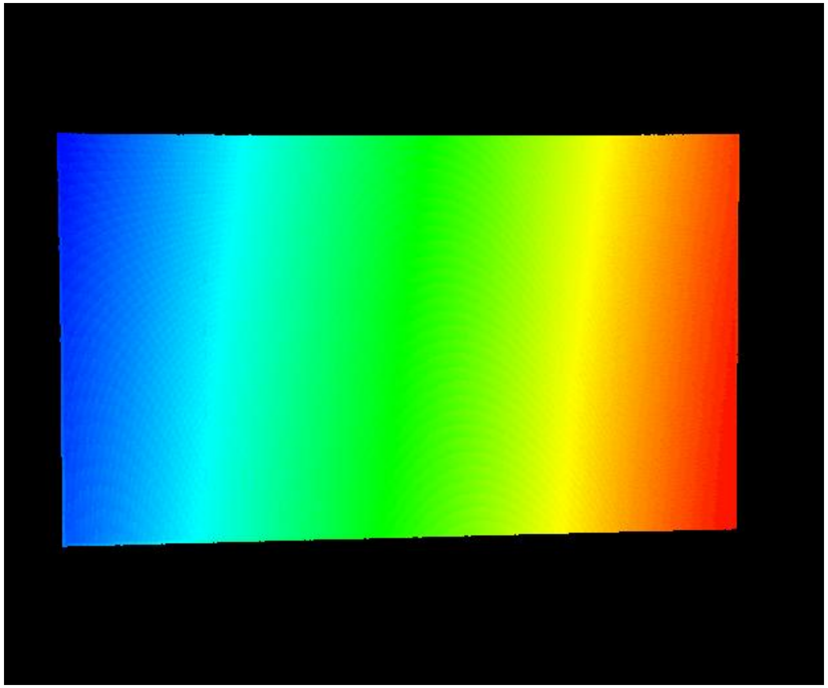
- Calibration image examples
- Measured projector reprojection error = 0.325859



Perform Scan

- After preparation and calibration, the system is ready for scanning!
  - Use one of the “Perform Scan” menu options 6, 7, or 8





Point Cloud Example

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265