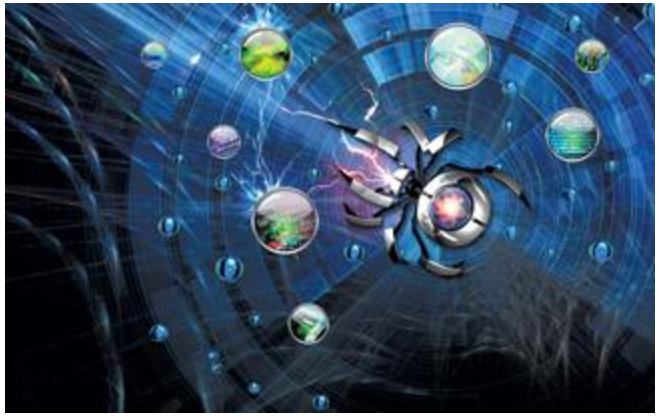
BAE SYSTEMS
INSPIRED WORK

GPU-accelerated 3-D point cloud generation from stereo images

Dr. Bingcai Zhang



BAE SYSTEMS
INSPIRED WORK

Contents

- Graphics processing unit (GPU)
  - Programming massively parallel processors
  - Thoughts after one year of GPU programming (C++ Accelerated Massive Parallelism [AMP])
  - Share experiences
  - Hopefully provide guidelines
- Automatic Spatial Modeler (ASM)
  - Review of state-of-the-art 3-D point cloud generation technology, image understanding, Automatic Feature Extraction (AFE)/identification
  - ASM
  - UAV images
  - Frame images

BAE SYSTEMS
INSPIRED WORK

A typical software development case: GXP case

- Geospatial eXploitation Products® (GXP) case
  - A few millions lines of code in C++
  - Linking to many third-party dlls
  - Visual Studio® 2008
  - Microsoft Windows® 7
- C++ AMP requirements
  - Microsoft Windows 8
  - Visual Studio 2012

BAE SYSTEMS
INSPIRED WORK

A typical software development case: GXP case …2

- GXP cannot upgrade to Visual Studio 2012 and Microsoft Window 8 yet
- Workaround solution
  - Developing GxpAsmAmp.dll with Visual Studio 2012 on Microsoft Windows 8
  - GxpAsmAmp.dll is totally independent from GXP
  - GxpAsmAmp.dll has advanced new stereo image matching algorithms that utilize GPU through AMP
  - GXP links to GxpAsmAmp.dll and calls functions inside GxpAsmAmp.dll

BAE SYSTEMS
INSPIRED WORK

A typical software development case: GXP case …3

- *C++ AMP: Accelerated Massive Parallelism with Microsoft® Visual C++®* by Kate Gregory and Ade Miller (Microsoft Press, 2012, 358 pages) is a great resource for our work
- Response from Ade Miller (co-author)
  - "What's far more likely is that you are compiling with Visual Studio 2008. I'm not even sure this will work at all. The VC 2008 compiler does not support the tile_static and restrict language extensions needed to build C++ AMP code. I also think you would need to link with the latest C++ runtime."
  - "How are you building your application? From what you say above I would not expect this to work regardless of your synchronization issues."
- Workaround solution
  - Works
  - Extremely inefficient

BAE SYSTEMS
INSPIRED WORK

Workaround solution

- Visual Studio 2008 GXP application, ASM, links to GxpAsmAmp.dll
  - Cannot debug GxpAsmAmp.dll
  - Cannot run Purify (Purify crashes right away)
- GXP application can only link to a released version of GxpAsmAmp.dll, not a debug version of GxpAsmAmp.dll
- Workaround for debug issue
  - GXP application writes memory/data into a binary file
  - A simple application on Microsoft Windows 8 with Visual Studio 2012
    - Reads the binary file
    - Calls the functions in GxpAsmAmp.dll with the same memory/data
    - Debug GxpAsmAmp.dll
- No workaround for crashes
  - OMG
  - Crashes in GXP application, but not in simple Microsoft Windows 8 application

BAE SYSTEMS
INSPIRED WORK

GPU speed up

- NVIDIA® Quadro K5000 card and Intel® Xeon® CPU (3.00GHz)
- Average ratio between CPU and GPGPU is 5.78
- Stereo image matching algorithm is very parallel
  - Two key algorithms run with >10,000 GPU threads
- Globe memory access is the bottleneck to achieve even high speed-up
- We hope the Maxwell Unified Virtual Memory can resolve the globe memory access issue
- Shortening development time for application developers is the key for the future of GPGPU

BAE SYSTEMS
INSPIRED WORK

CPU and GPU timing table

– One of the most expensive new image matching algorithms in ASM
– Unit is in microsecond

CPU	986.77	1322.69	2509.95	1156.34	2412.70	571.07	1356.52	878.70
GPGPU	185.57	141.19	552.87	209.96	645.65	122.97	221.21	125.73
Ratio	5.32	9.37	4.54	5.51	3.74	4.64	6.13	6.99

BAE SYSTEMS
INSPIRED WORK

GPU programming cost

- GPU software engineering is much more expensive than CPU software engineering.
- At last GTC conference, Lawrence-Berkeley Lab gave a 4:1 ratio
- Our case is even higher than a 4:1 ratio
- AMP questions, issues, and solutions at: Parallel Computing in C++ and Native Code Forum
- Only a very small portion of a large application needs to run on GPU

BAE SYSTEMS
INSPIRED WORK

GPU programming guidelines

- Is your algorithm parallelizable?
  - Can your algorithm run with hundreds or thousands of GPU threads concurrently?
  - If the answer is NO, GPU is not for your algorithm
  - It is much more challenging to develop parallel algorithms
- Does your algorithm need to access a lot of memory/data?
  - AMP has limited sum of registers (4096)
  - Global memory access is slow
  - If it needs a lot of global memory access and little computation, then GPU is not for your algorithm
- Separate CPU code from GPU code
  - Into different classes
  - Into different files
  - Compiling GPU code is EXTREMELY slow

Automatic image object recognition/extraction

- It is a trivial task for a five-year-old child to recognize and name an object such as a car, house, or building. However, it is a challenging software problem to identify and label these same objects automatically in a digital image.
- Geospatial information technology, such as digital photogrammetry, can answer the “where” question accurately. The next breakthrough may be the “what” question, which is to identify and label objects automatically in digital imagery.
- Automatic 3-D building extraction from digital imagery is considered the “holy grail” in photogrammetry
- It is very difficult to extract buildings automatically from images using only their radiometric properties

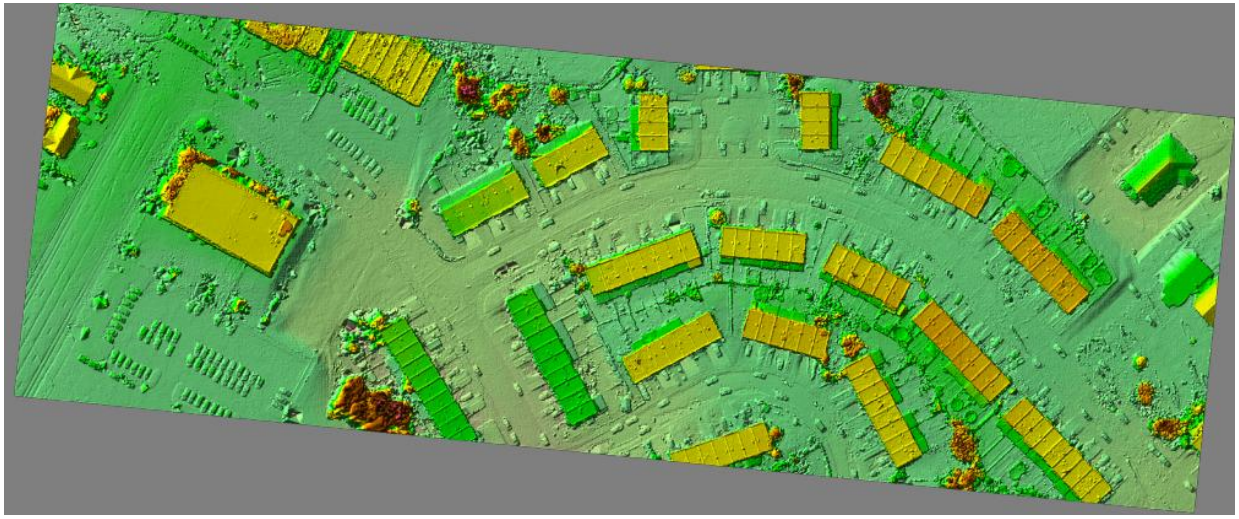
BAE SYSTEMS
INSPIRED WORK

Automatic image object recognition/extraction …2

- In the past three decades, many algorithms have been developed to extract 3-D buildings from a very specific set of digital images
- Until now, there has not been a commercial software package that can reliably do this
- Our approach:
  - Indirectly identify/extract 3-D objects from images
  - Generate 3-D point clouds from stereo images first
  - Extract 3-D objects from generated 3-D point clouds
  - Many years of expertise in 3-D point cloud generation from stereo images
  - 3-D objects in 3-D point clouds have an invariant property of 3-D geometry shape
- Extracting 3-D objects from 3-D point clouds bypasses the complexity of image pixel properties and directly uses the invariant 3-D property of any 3-D object

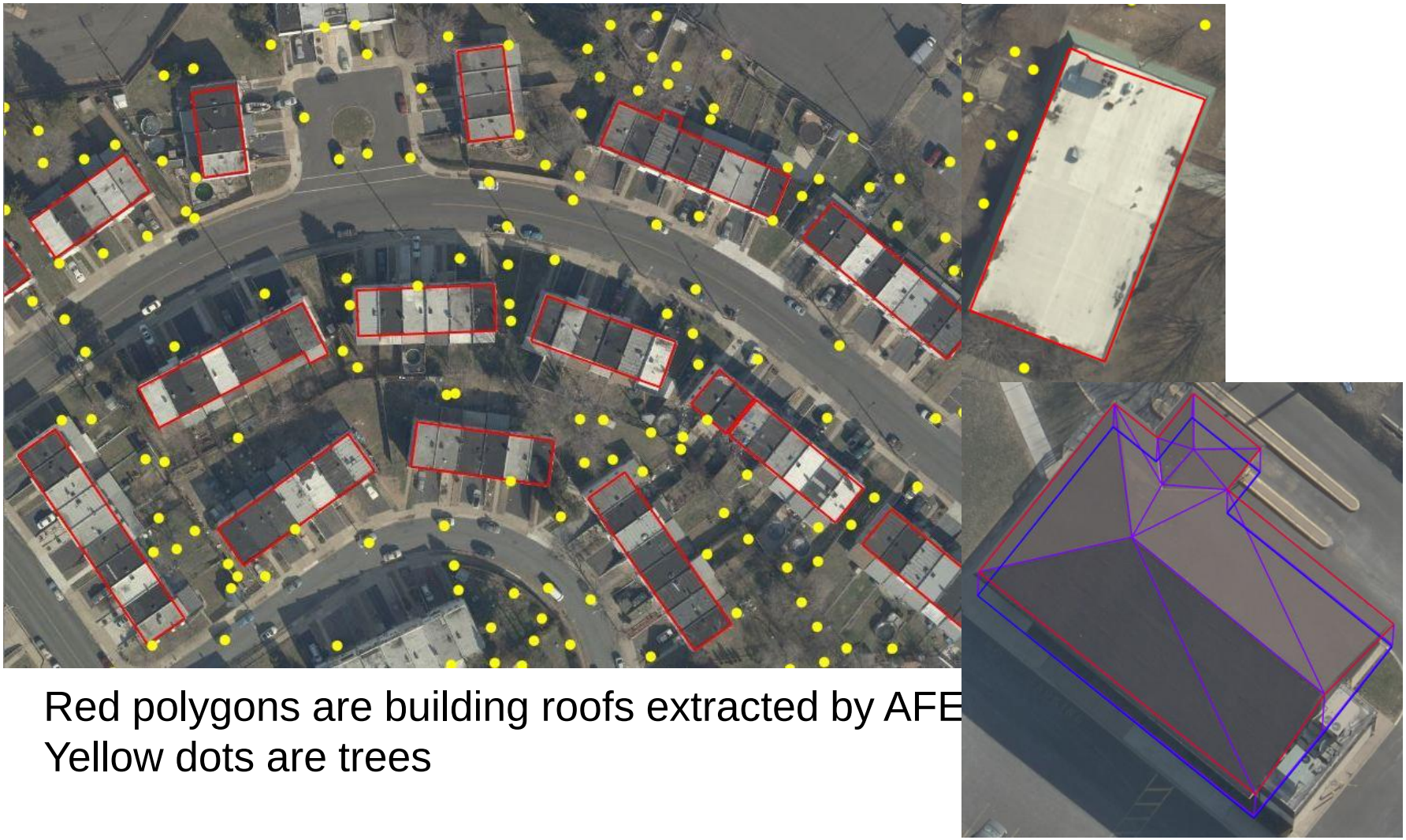
Radiometry vs. 3-D shapes

- Six different building colors and patterns
- Very difficult to extract buildings based on radiometric properties only



- Terrain Shaded Relief (TSR) of Digital Surface Model (DSM) generated by ASM
- The locations and approximate shapes of the buildings are obvious

Automatic image object recognition/extraction …3



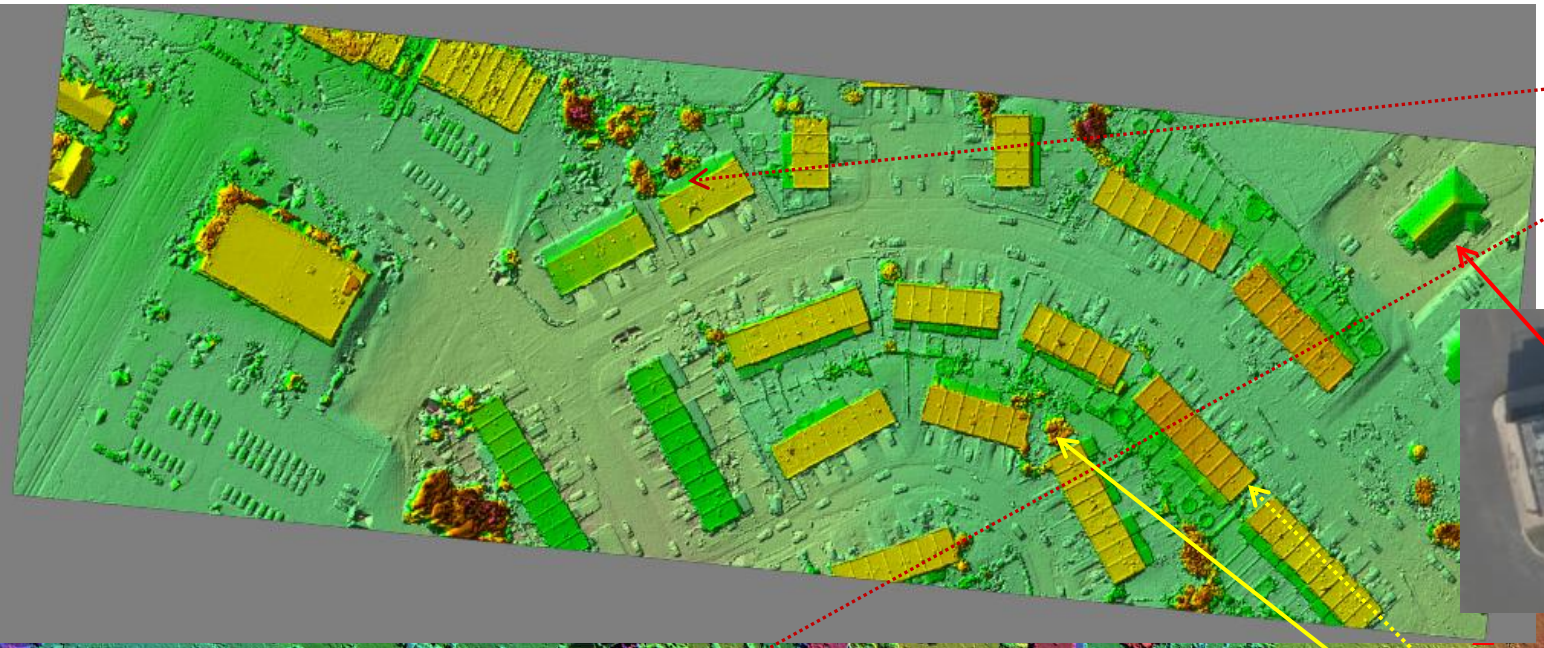
Red polygons are building roofs extracted by AFE
Yellow dots are trees

BAE SYSTEMS
INSPIRED WORK

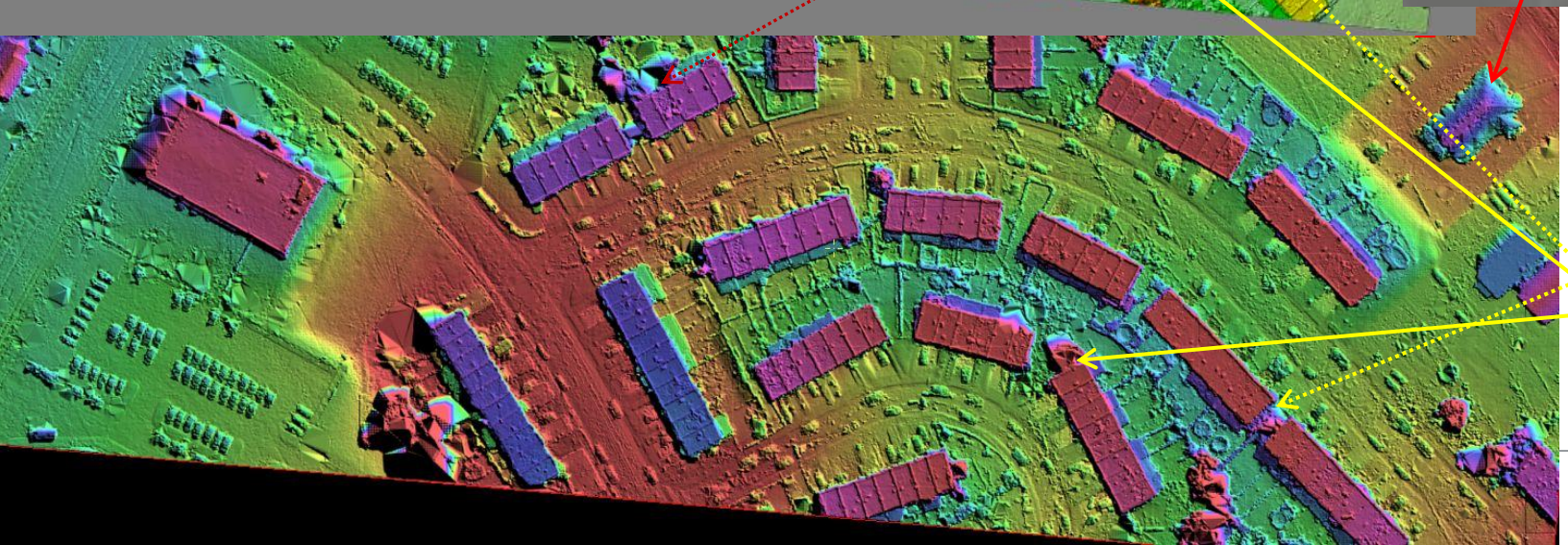
Stereo image matching algorithms

- Global Least Square Correlation (GLSC) and array algebra in the 1990s from Rauhala
- NGATE released in 2006
- Publication of Semi-Global Matching (SGM) by Hirschmuller in 2007
- Open Source Computer Vision Library (OpenCV)
- The significant contribution from SGM is the smoothness constraint, well published with specifics on its implementation
  - Allows smaller matching window
  - Increases robustness and reliability even with smaller matching window
- ASM is a hybrid of NGATE, SGM smoothness constraint, GPU, computer vision, and several new algorithms

Accuracy comparison between ASM and NGATE



ASM generates sharper building edges



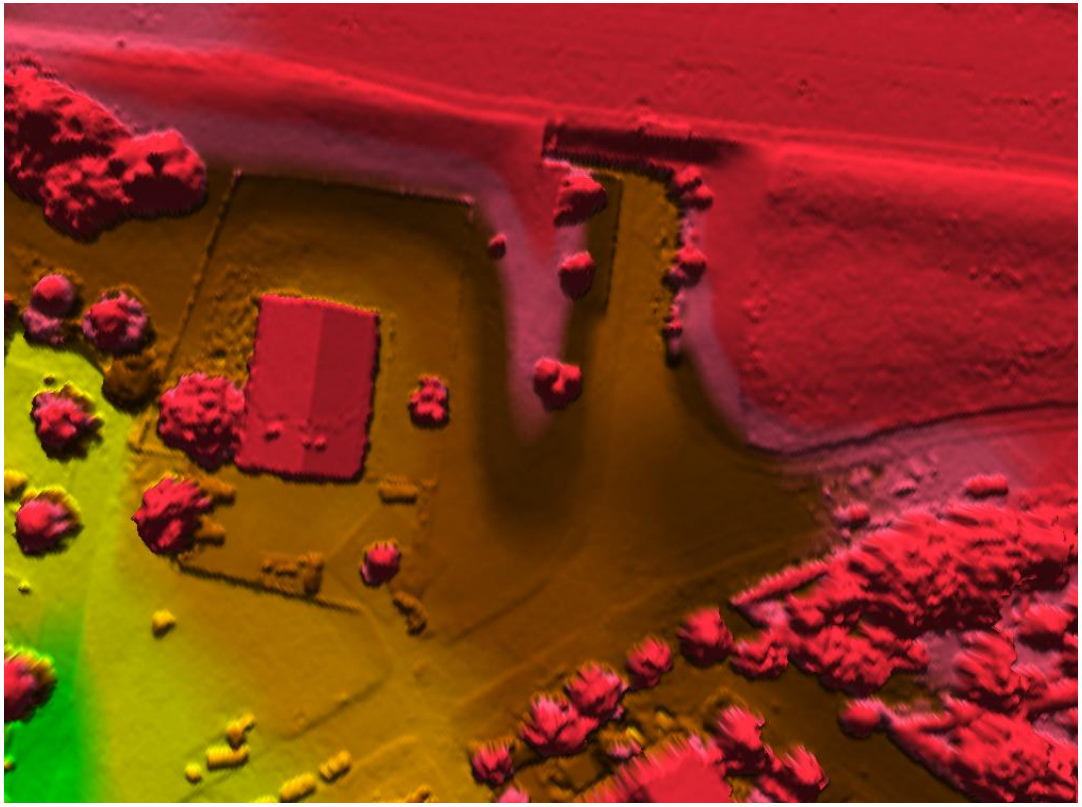
ASM generates sharper building edges

UAV images

UAV images …2

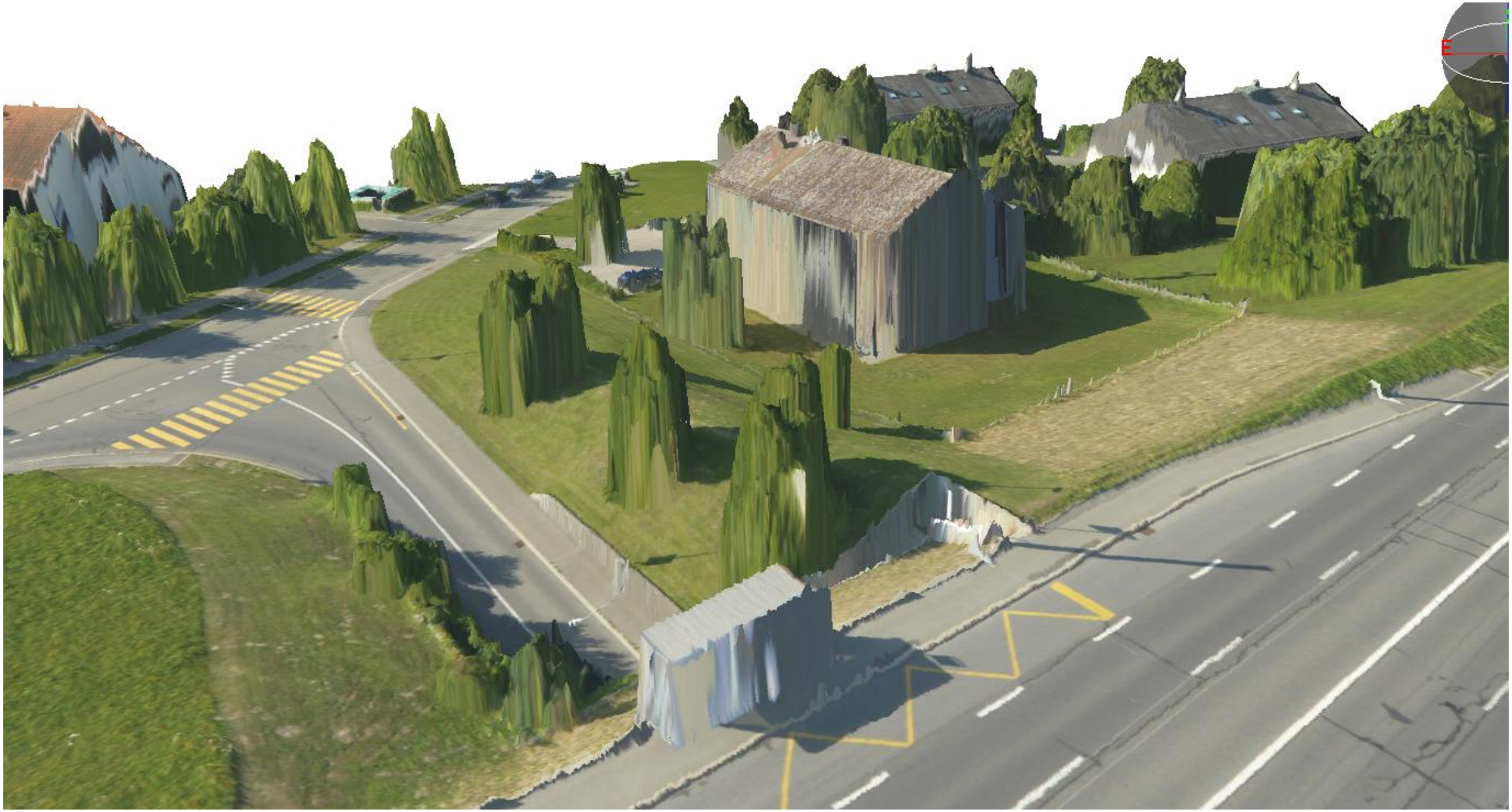
- Small format:
  - Lines: 6132
  - Samples: 8176
- Extremely high resolution:
  - GSD: 0.038 m
- Non-metric frame camera (Hasselblad color digital camera)
  - Affordable (much cheaper than metric aerial camera)
  - Significant lens distortion (up to 20 pixels)
  - Lightweight (can mount on a small UAV)
- UAV from 3-D Robotics costs approximately $1000

UAV images …3



- Automatically identify and extract small 3-D objects, such as cars, from 4cm UAV images
- What if we have 1cm UAV images?
- The potential for applications outside 3-D mapping is much larger

3-D view of UAV images plus DSM



UAV image accuracy assessment master Digital Terrain Model (DTM)

Quality statistics

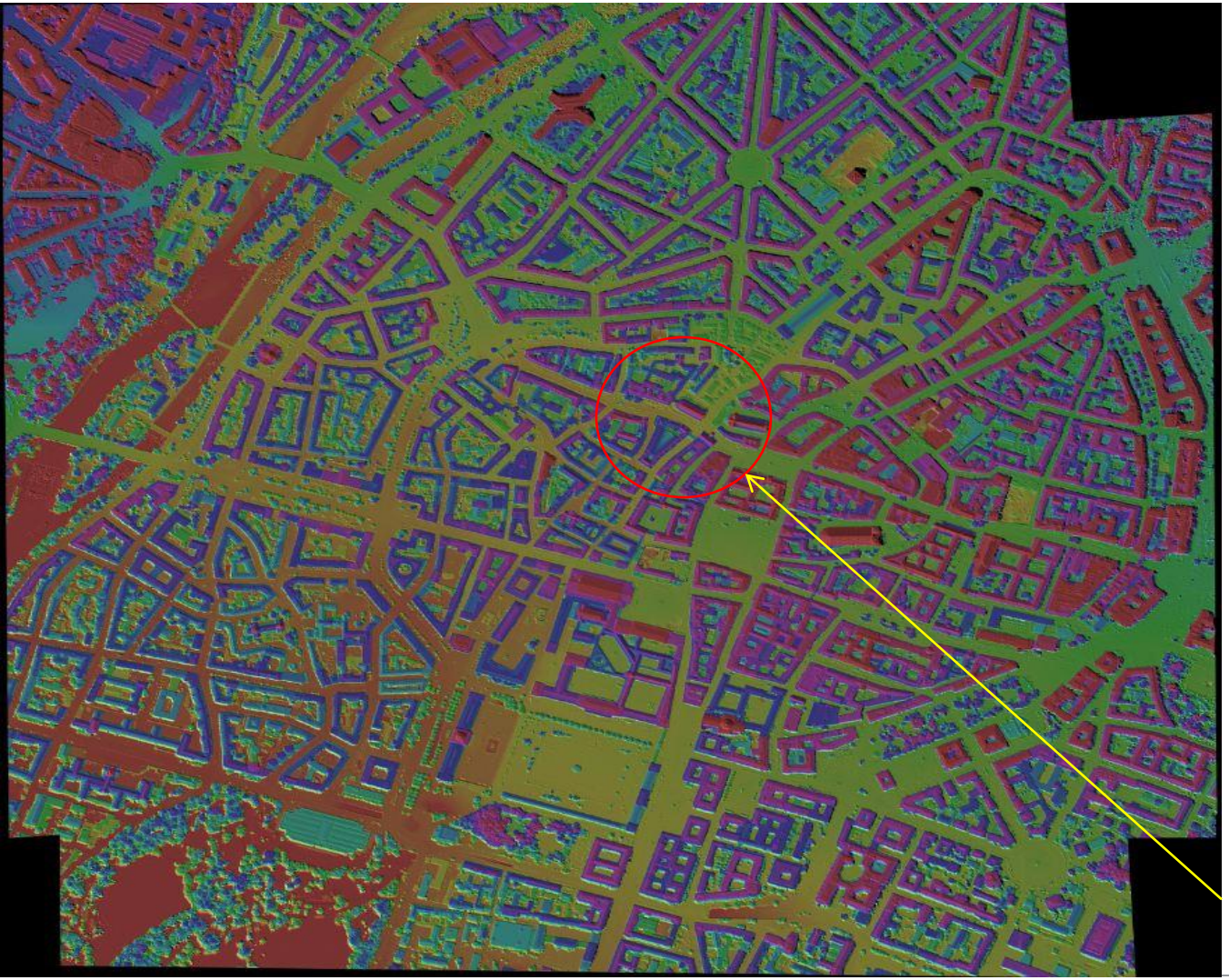
- Comparison of two DTM files
- Master DTM File: EDIT_MASTER.dth
- Slave DTM File: horizontal_strip_6_pairs.dth
- num_pts = 124, rms = .0532, std = .0533, bias = -.0038
- percent blunders = .0000
- After two outlier removals
- num_pts = 121, rms = .0366, std = .0367, bias = .0022
- percent blunders = 2.4793
- After three outlier removals
- num_pts = 120, rms = .0352, std = .0352, bias = .0032
- percent blunders = 2.5000
- Note: bias = (Z from Master) - (Z interpolated from Slave)

Aerial frame images

15 Intergraph DMC frame images with 10 cm GSD, 16 bits per pixel, panchromatic, 15,552 x 14,144, München, Germany

Make sure images are properly triangulated: American photogrammetry and German photogrammetry may be different

Aerial frame images …2



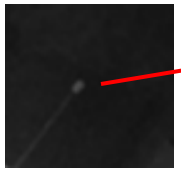
452 million posts at 10cm spacing DSM

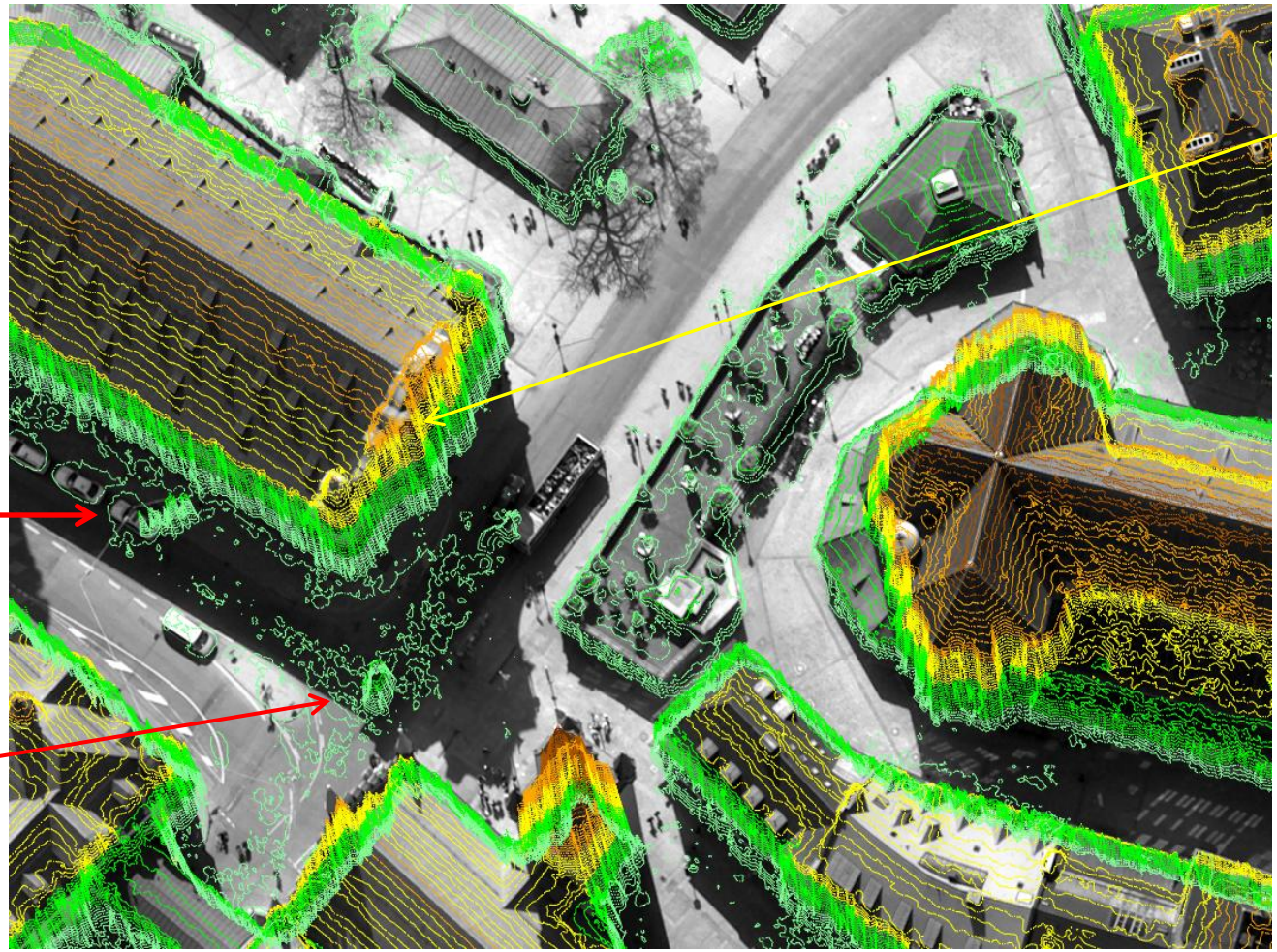
Intel Xeon CPU 3.2 GHz (4 processors), 24 GB, GeForce® GTX Titan

3h 18m to complete: 4 sections, 9 stereo pairs per post, asm_urban.strategy

Dense matching benchmark site

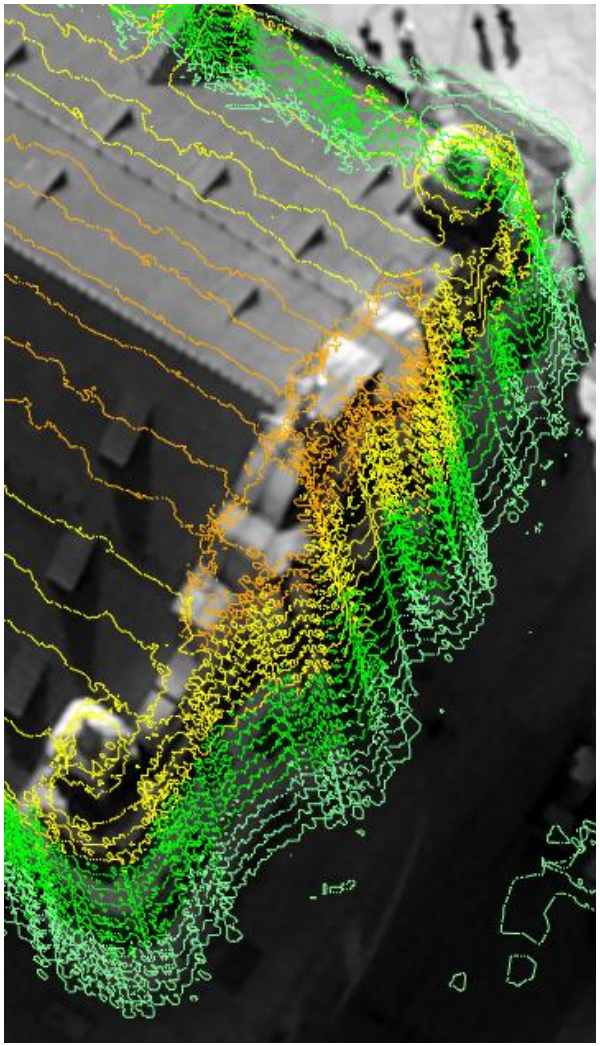
Aerial frame images …3

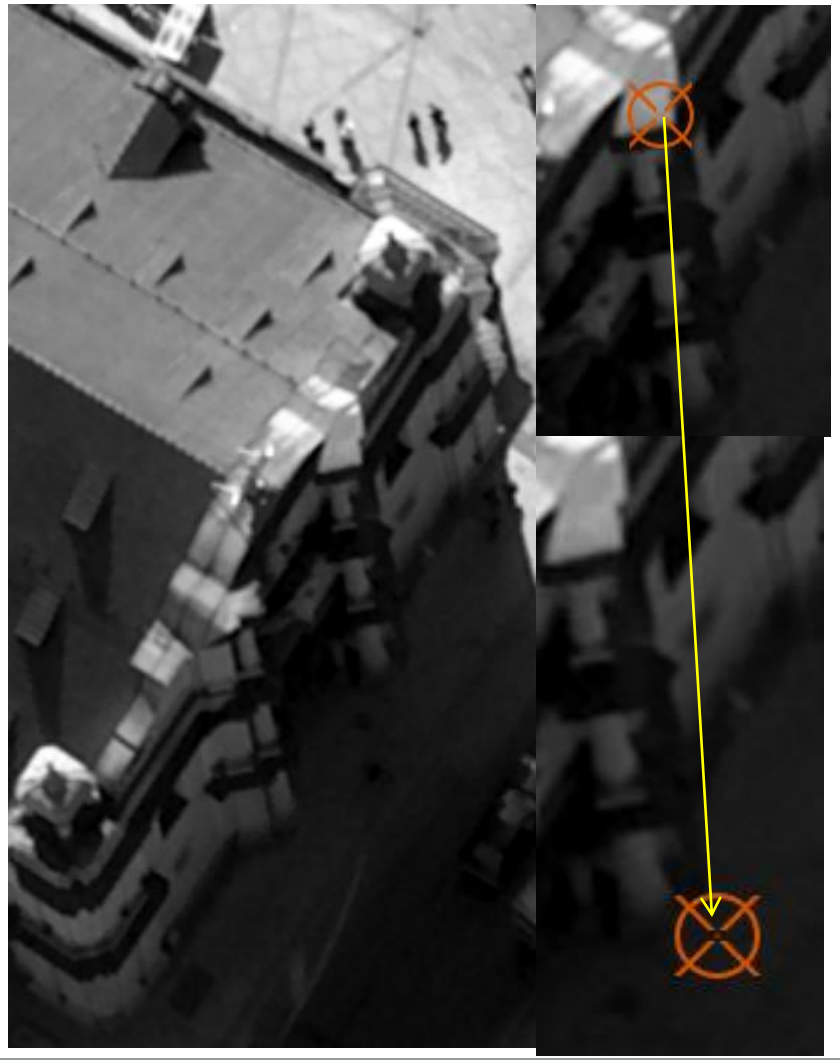




University of Stuttgart dense matching benchmark site

Aerial frame images …4



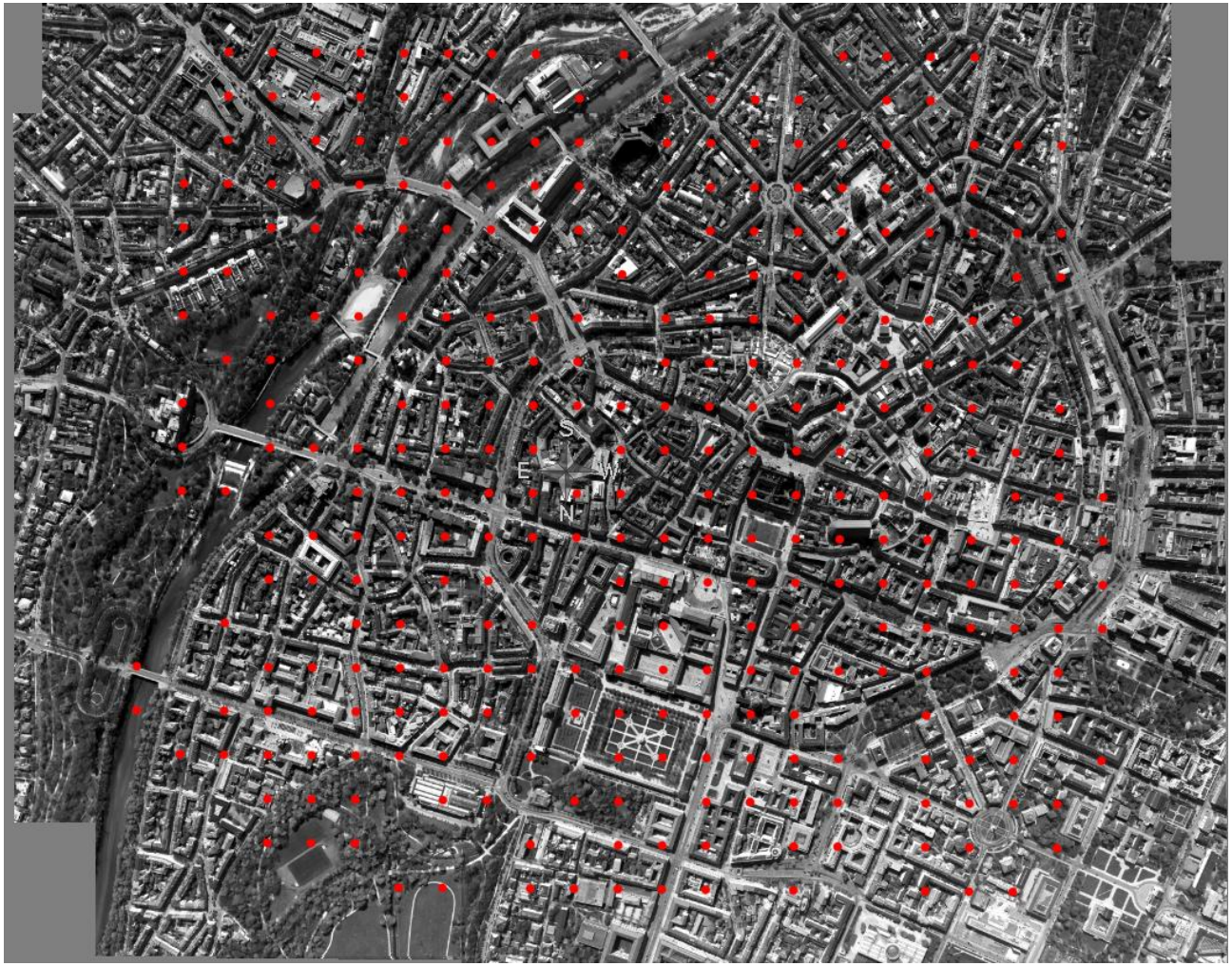


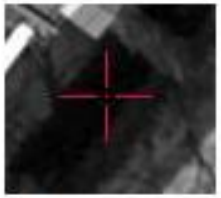
X=25.13
Y=25.88
Z=98.33

DSM has 89° slope at building edge

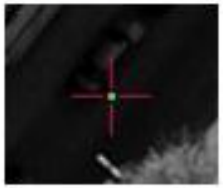
X=24.58
Y=25.91
Z=65.51

Ground truth points



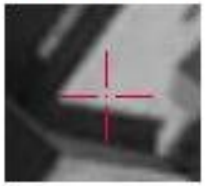


-2.34 meters



-1.01 meters

-0.88 meters



-0.66 meters

Quality statistics

- Comparison of two DTM files
- Master DTM File: EDIT_MASTER2.dth
- Slave DTM File: ASM_10cm_12pairs_2sections_3.dth
- num_pts = 316, rms = .2308, std = .2248, bias = -.0539
- percent blunders = .0000
- After two outlier removals
- num_pts = 313, rms = .1745, std = .1699, bias = -.0409
- percent blunders = .9585
- After three outlier removals
- num_pts = 312, rms = .1707, std = .1665, bias = -.0389
- percent blunders = .9615
- Note: bias = (Z from Master) - (Z interpolated from Slave)

BAE SYSTEMS
INSPIRED WORK

Summary

- ASM can work well with satellite images, ADS40 images, frame images and UAV images
- ASM is accurate and fast
- Capability of GPU card is still doubling every 1.5 years?
  - ASM can take advantage of that for free
- Parallel programming is a new trend in software engineering
  - Utilizing GPU
  - Utilizing multi-core CPU
  - C++ AMP can use both GPU and multi-core CPU
  - C++ AMP lowers the entry point to parallel programming
  - C++ AMP needs to be as reliable as the rest of Visual C++
- GPU programming is much more expensive than CPU programming

BAE SYSTEMS
INSPIRED WORK

Thank you!

Special thanks to Joseph Spann, Michael Schwarzlose, Raddoslav Gaidadjiev, Dr. Stewart Walker, William Smith, and Mark VanDierendonck, Microsoft AMP team

bingcai.zhang@baesystems.com