# Toward Fast Computation of Dense Image Correspondence on the GPU

**Mark Duchaineau, Jonathan D. Cohen, Sheila Vaidya**

**Lawrence Livermore National Laboratory**

**High Performance Embedded Computing Workshop**

**18-20 September, 2007**

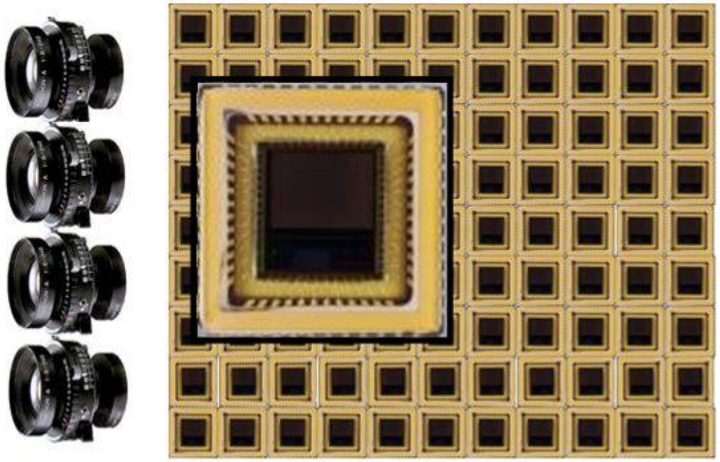
**Auspices Statement**

**This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.**

# Are UAV gigapixel video cameras feasible in the next few years? Yes, but…

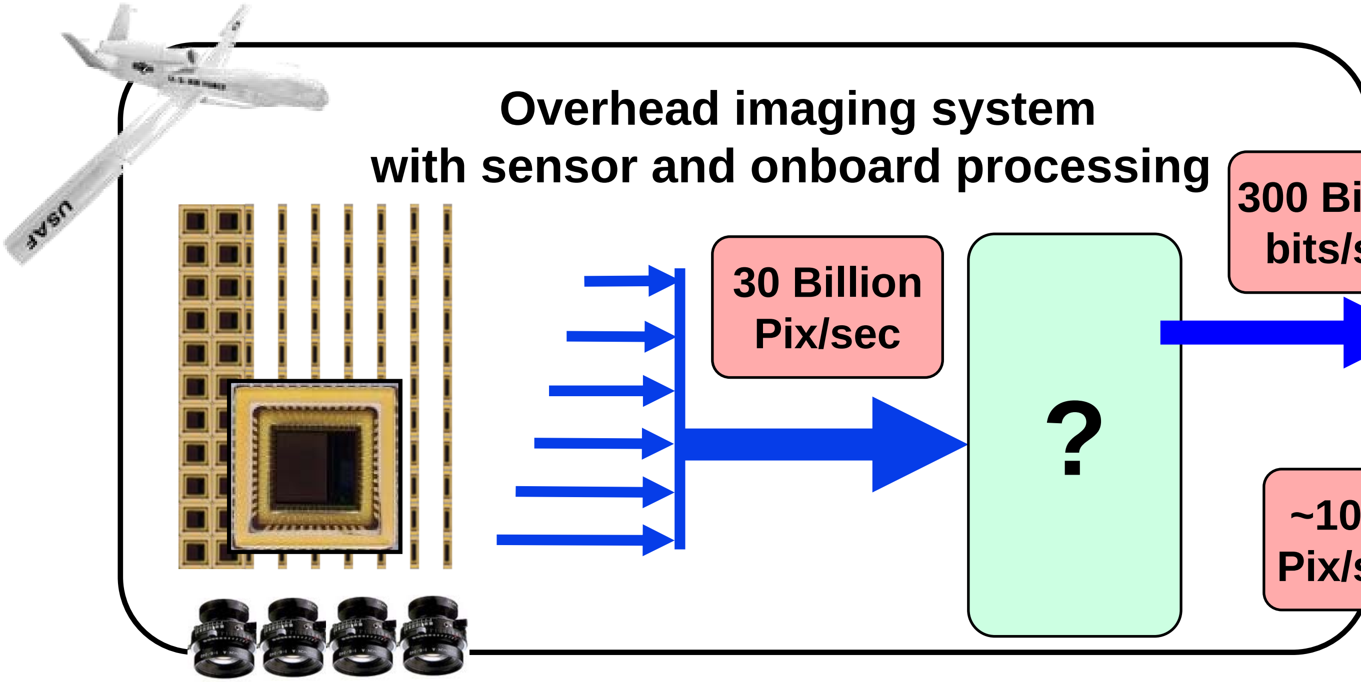
## Recipe for Gigapixel Video?:

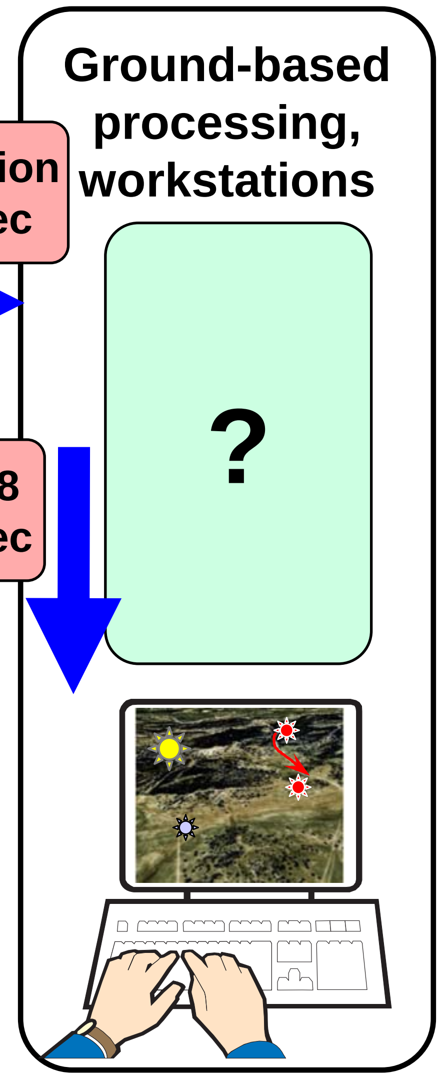
- **320 KAC-3100 CMOS sensors**
  - 2048x1536 2.7um pixels
  - 12Hz full-frame readout, 128mm
  - Noise and packaging issues
- **4 professional camera lenses**
  - Not high enough resolution
  - => custom optics, non-planar focus
- **Major innovations in packaging, assembly and calibration needed**
  - => MIT LL design and automation
- **Massive supercomputer onboard**
  - => *selective refinement* pipeline



- **Pros**
  - 32000x32000 pixels
  - Catches transitory events
  - Many off-the-shelf parts
- **Cons**
  - 1.0 terabyte per minute
  - Massive stream compute must be near the sensor

# Challenge for future wide-area airborne video is handling “1000 HDTV feeds”



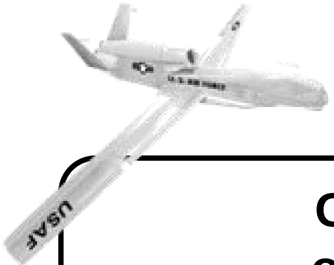

**Increase the number of pixels per second 1000x *within similarly sized overhead vehicle***

**— Sensor and optics for a 3 billion pixels @ 10Hz**

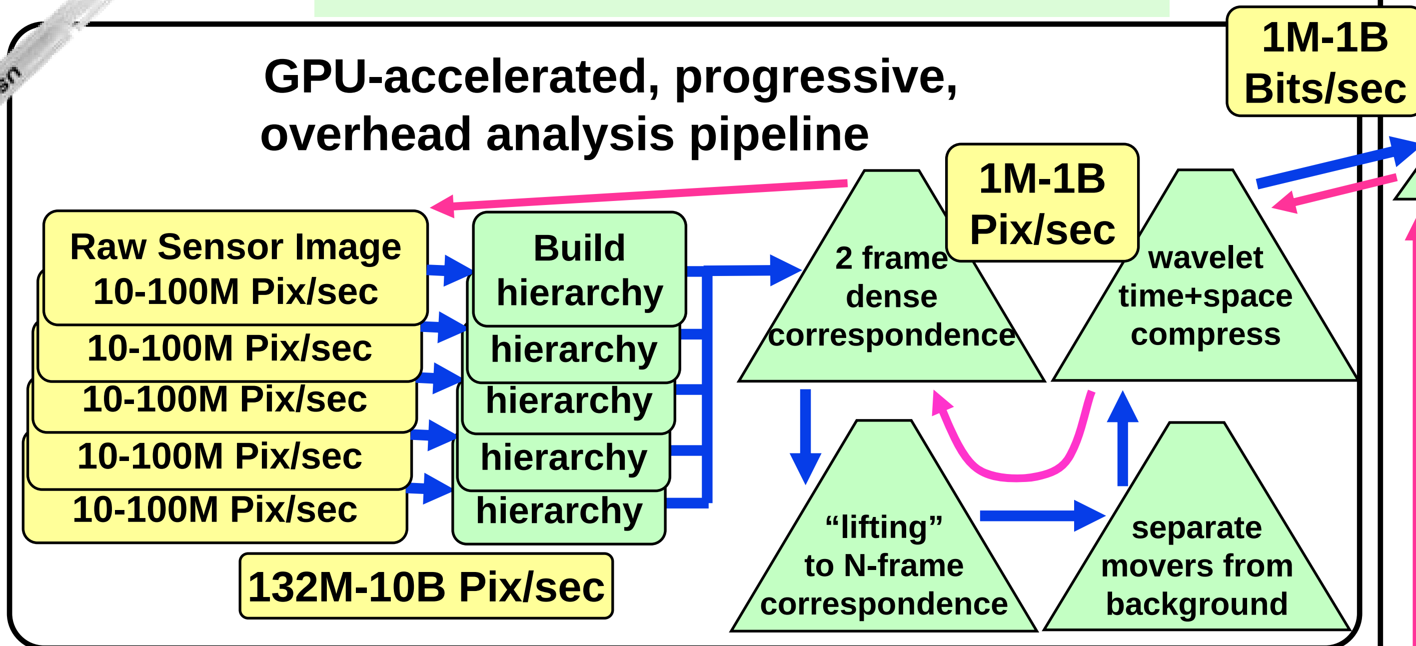
**— Processing and distribution of 1000 HDTV feeds**

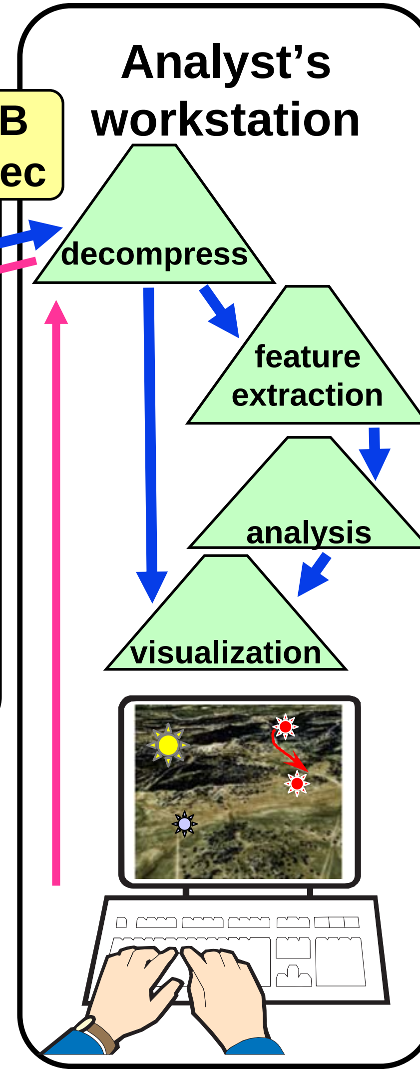
**This effort focuses on solving 3D-enabled onboard analysis, processing scalability**

# LLNL is designing *selective refinement* overhead processing for extreme scalability



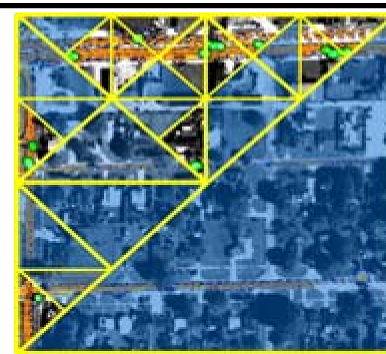

## Real-time Optimally Adapting Meshes for scalable aerial video




**ROAM streams are scalable, matched to available resources**

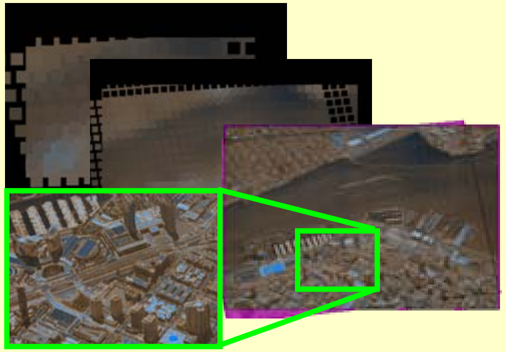
- Real-time analysis
- End-to-end optimality feedback



# Key ideas for wide area aerial video: dense correspondence and selective refinement

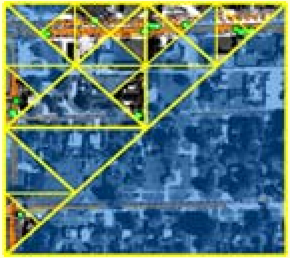
- *Image correspondence* determines where points in a scene move in the pixel arrays over time
  - Enables stabilization of video for human and machine analysis
  - Critical for temporal compression (10-100x more compression than spatial schemes)
  - Dense correspondence allows 3D analysis: eliminates parallax, super-resolution, etc…
- *Selective refinement* works in pieces, just where it is needed
  - Allows optimal exploitation of sparseness (e.g. movers) and temporal coherence
  - Best utilization of limited onboard compute resources given dynamic scenes, priorities

**Real-time dense correspondence**



**Sub-pixel warping enables 3D analysis**

**Selective Refinement Tile Processing**



**Exploiting coherence gives 10-100x speedups**

# Dense correspondence algorithms stabilize aerial imagery more than image registration

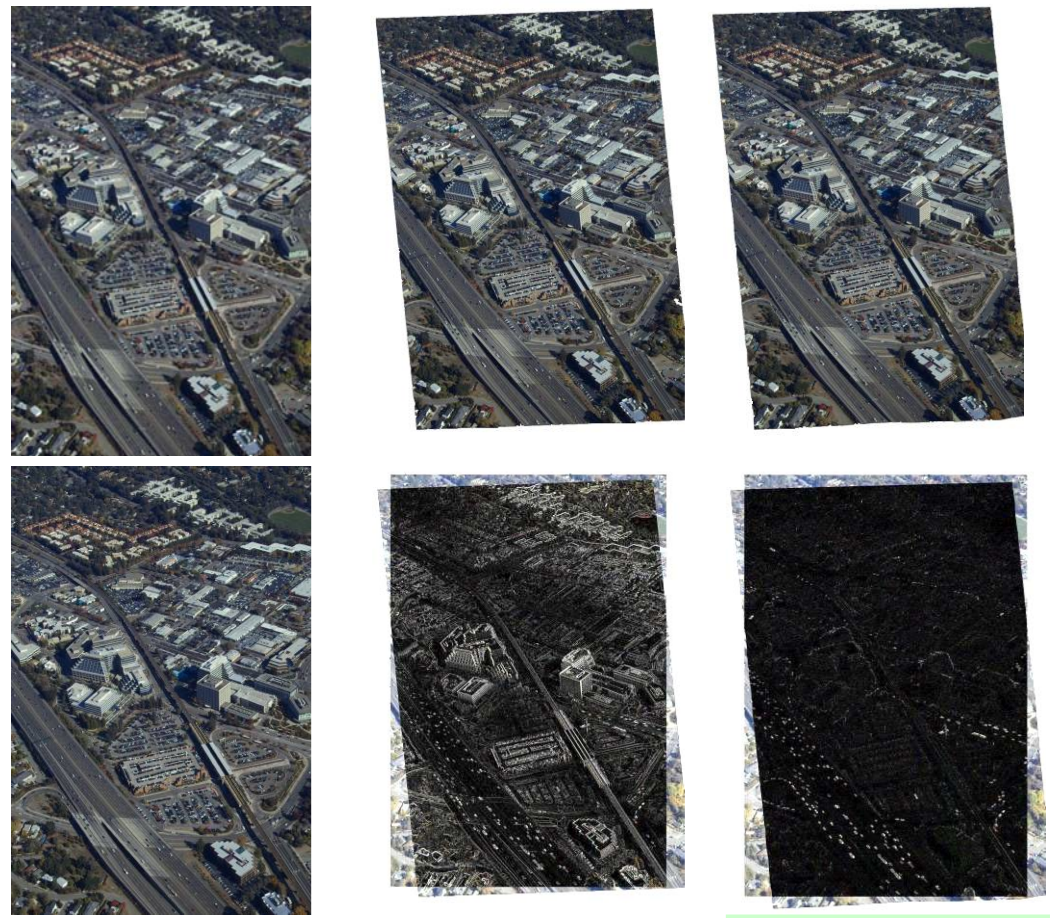
- Dense correspondence provides millions of degrees of freedom, compared to 3-48 degrees of freedom for global image registration
- This gives better mover/background analysis, compression and additional advanced features, e.g. resolution enhancement, 3D
- Real-time dense correspondence is fast on new multi-core chips

Unstabilized images | Global registration | Dense correspondence

Images courtesy Brett Wayne

# Dense correspondence eliminates “false” change detections due to parallax

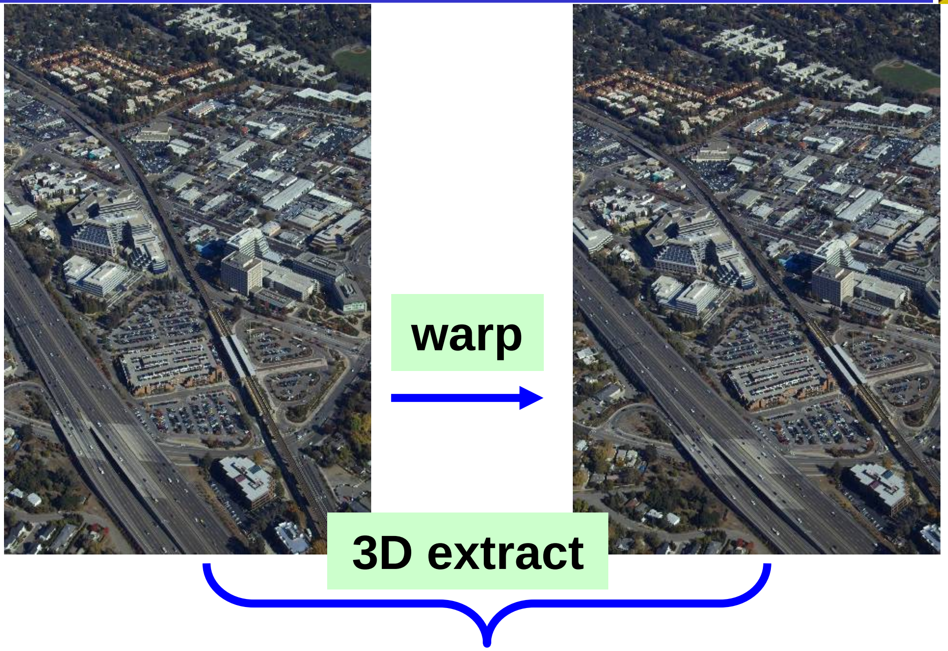
- Affine warp versus dense correspondence
  - Left: source and target images
  - Middle: Affine warp of source with difference image
    - Major difference from parallax
  - Right: Dense correspondence warp with difference image
    - Main differences are movers

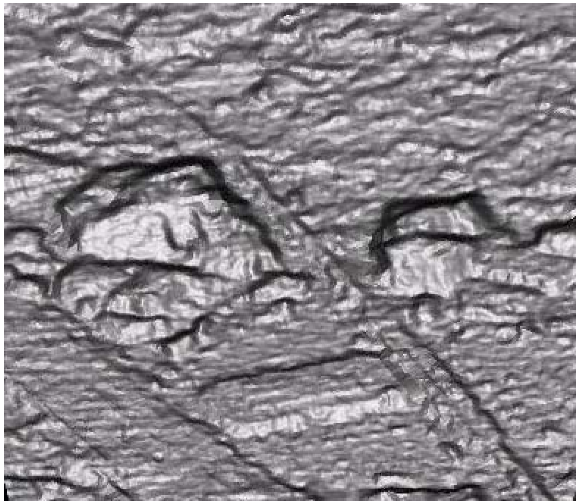


inputs | global registration | dense correspondence

# Dense correspondence is the main computational effort in 3D extraction

- 3D points can be extracted using a conventional depth-from-stereo algorithm pipeline:
  - Perform dense correspondence
  - Extract camera pose
  - Triangulate to obtain 3D samples



Collaboration with Michael J. Goldman, Mauricio Hess-Flores and Prof. Kenneth I. Joy of U.C. Davis

# Dense correspondence, selective refinement define second generation systems

- **Generation 1:**
  - GPS/INS and flat earth model ➔ simple projection, but not stable for 3D scenes
  - Global image registration, compose mappings for stabilization
  - Background subtraction for mover detection
  - Compression: background once per minute, movers as sprites (LLNL) or raw back-ground subtraction (MITLL)
  - Operations on whole images, many resamplings
- **Generation 2:**
  - 3D parallax tracking directly from images ➔ expensive solver, stable everywhere
  - Register to sample image per minute --or-- to 3D model extracted in real-time
  - Fewer false positives in mover detection
  - Super-resolution and signal-to-noise enhancement
  - Parameterized time for background and movers likely will allow 1000x compression
  - Operates on sparse subsets, one resampling at end of pipe

# Processing includes pre-flight calibration, onboard streaming, ground-station analysis

## Calibration processing

1) Bias/gain maps per pixel
    - Fine-scale pixel variations
    - Broad-scale vignetting effects
2) Focal-plane stitching geometry
3) Ray direction map per pixel
4) Noise modeling
5) Temperature adjustment model
6) Optimize wavelets and layout

## Per-frame onboard stream processing

- Image conditioning
- Derived values
- Hierarchy building
- Dense correspondence
- Mover/background segmentation
- Multi-frame layout and fitting
- Wavelet transform in time, space
- Quantization, encoding, packing

## (Semi-)automated ground-based processing

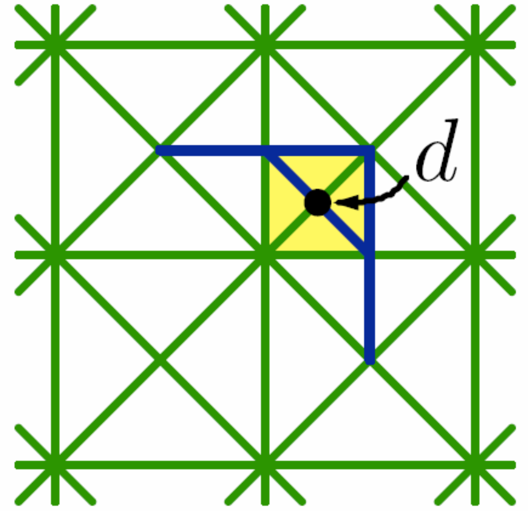
- General feature extraction
- Track generation/network analysis
- Road and facility modeling
- Multi-mission integration
- Database generation

## Interactive ground-based analysis

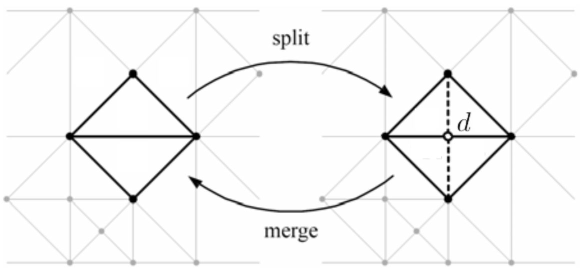
- Rapid pan/zoom in space and time
- Manual track generation
- Facility monitoring
- Analyst integration of additional data layers
- Database query

# Real-time Optimally Adapting Meshes/ROAM is our selective-refinement backbone

**Uses 4-8 meshes with diamond data structure**



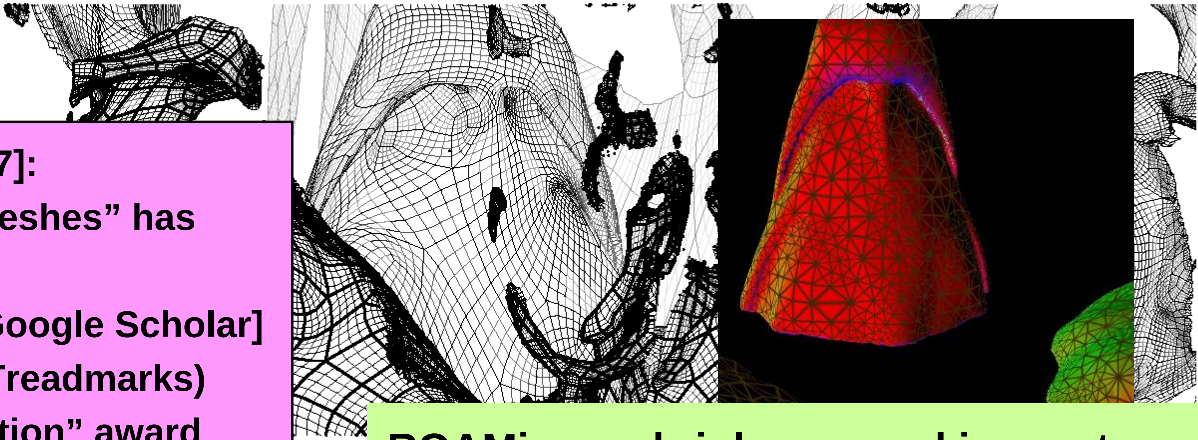

**Dual-queue incremental updates exploit frame-to-frame coherence**




**Flexible framework – many optimizations, extensions**

- frustum cull
- defer priority compute
- triangle patches
- texture tiles
- line-of-sight
- 3D diamonds
- fast dense correspondence

**Claims to fame [As of Aug 29, 2007]:**
- **"Real-time Optimally Adapting Meshes" has 11,900 Google hits**
- **'97 paper has 427 citations [via Google Scholar]**
- **Used in several game titles (e.g. Treadmarks)**
- **Los Alamos "Outstanding Innovation" award**
- **Vis '04 "Best Paper" (Hwa/Duchaineau/Joy)**



**ROAMing a shrink-wrapped isocontour from an 8B elem Gordon-Bell Prize run**

# A selective refinement ROAM tile hierarchy starts with image pyramid construction

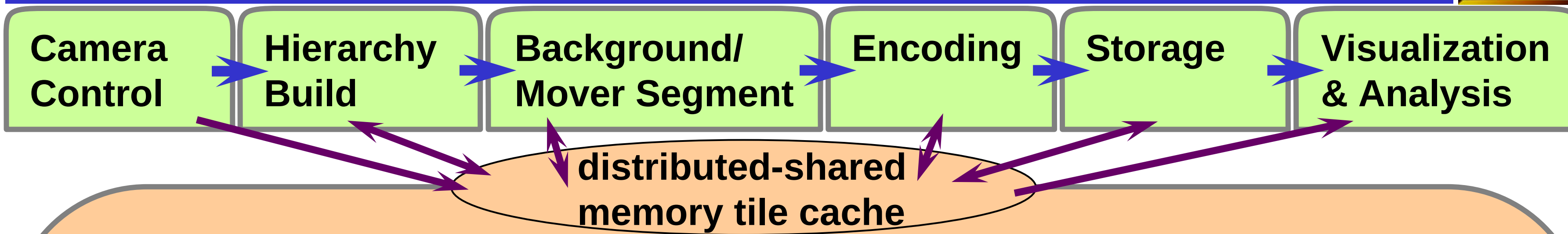
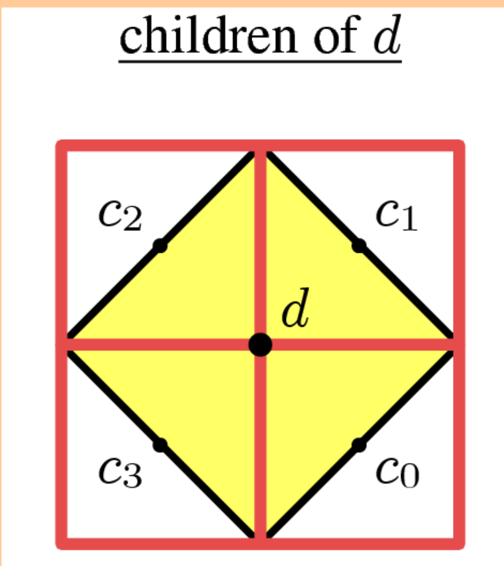
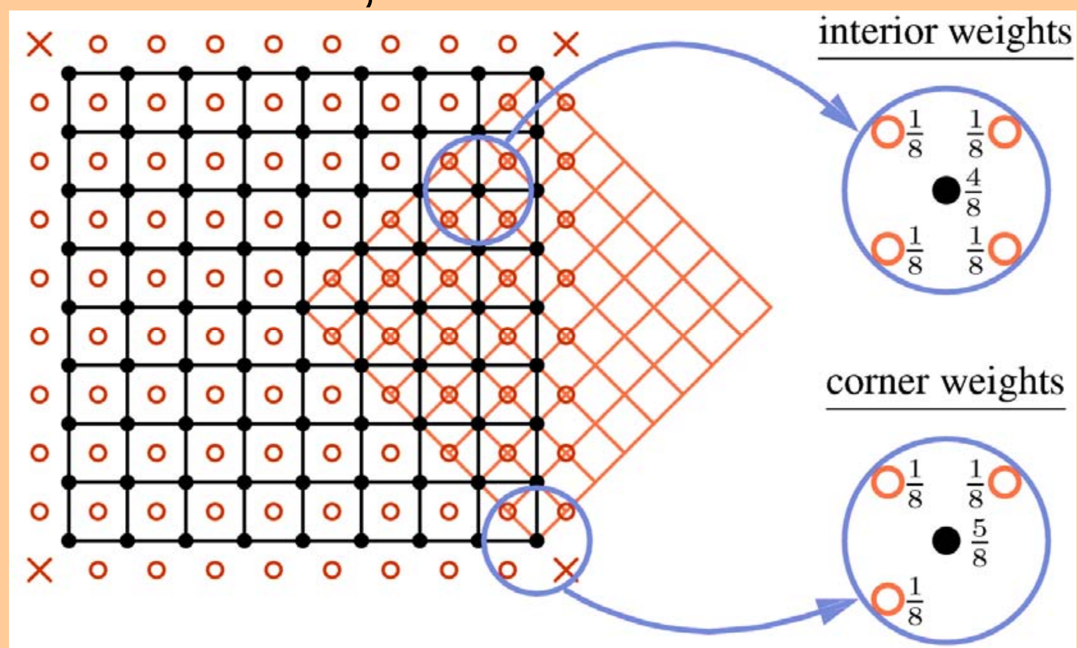



Image hierarchy processing by example: a low-pass wavelet filter reads four fine tiles, writes one coarse tile

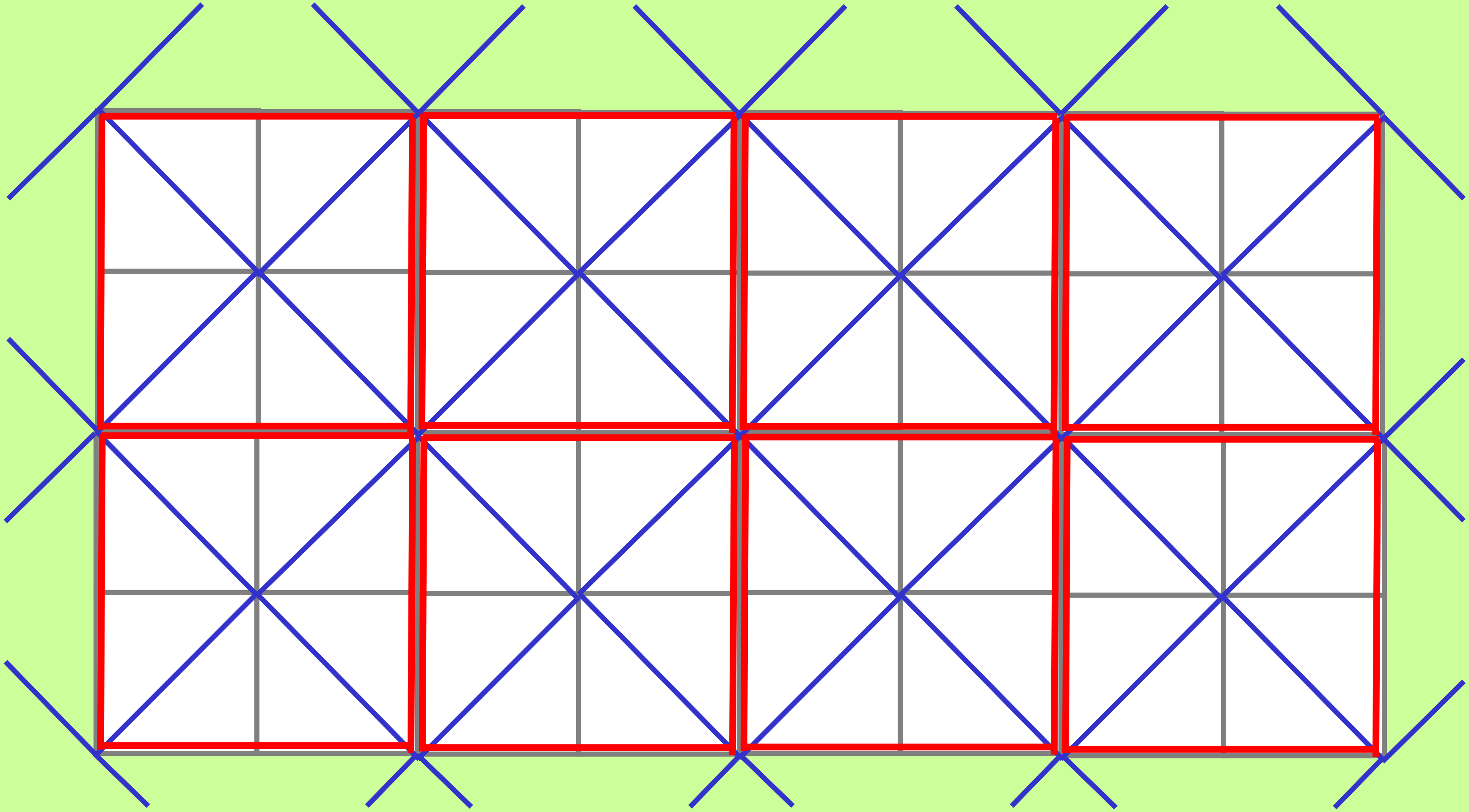



Hollow dots : cell-centered data
Solid dots: vertex-centered data

# Multi-resolution tiles are built through local 4-8 wavelet transform filters

Camera Control → **Hierarchy Build** → Background/ Mover Segment → Encoding → Storage → Visualization & Analysis

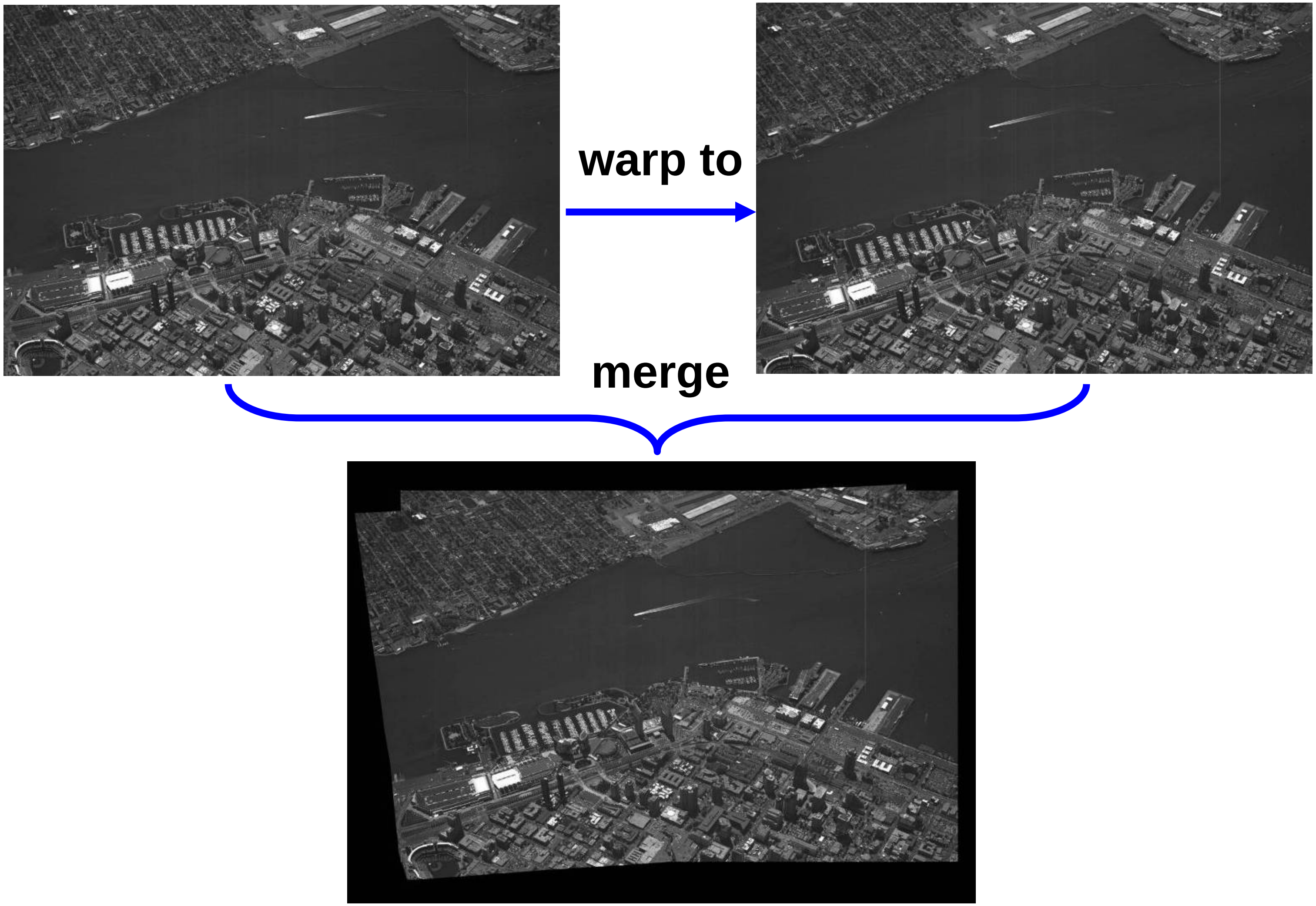


# We are developing new, ROAM-based dense image correspondence algorithms

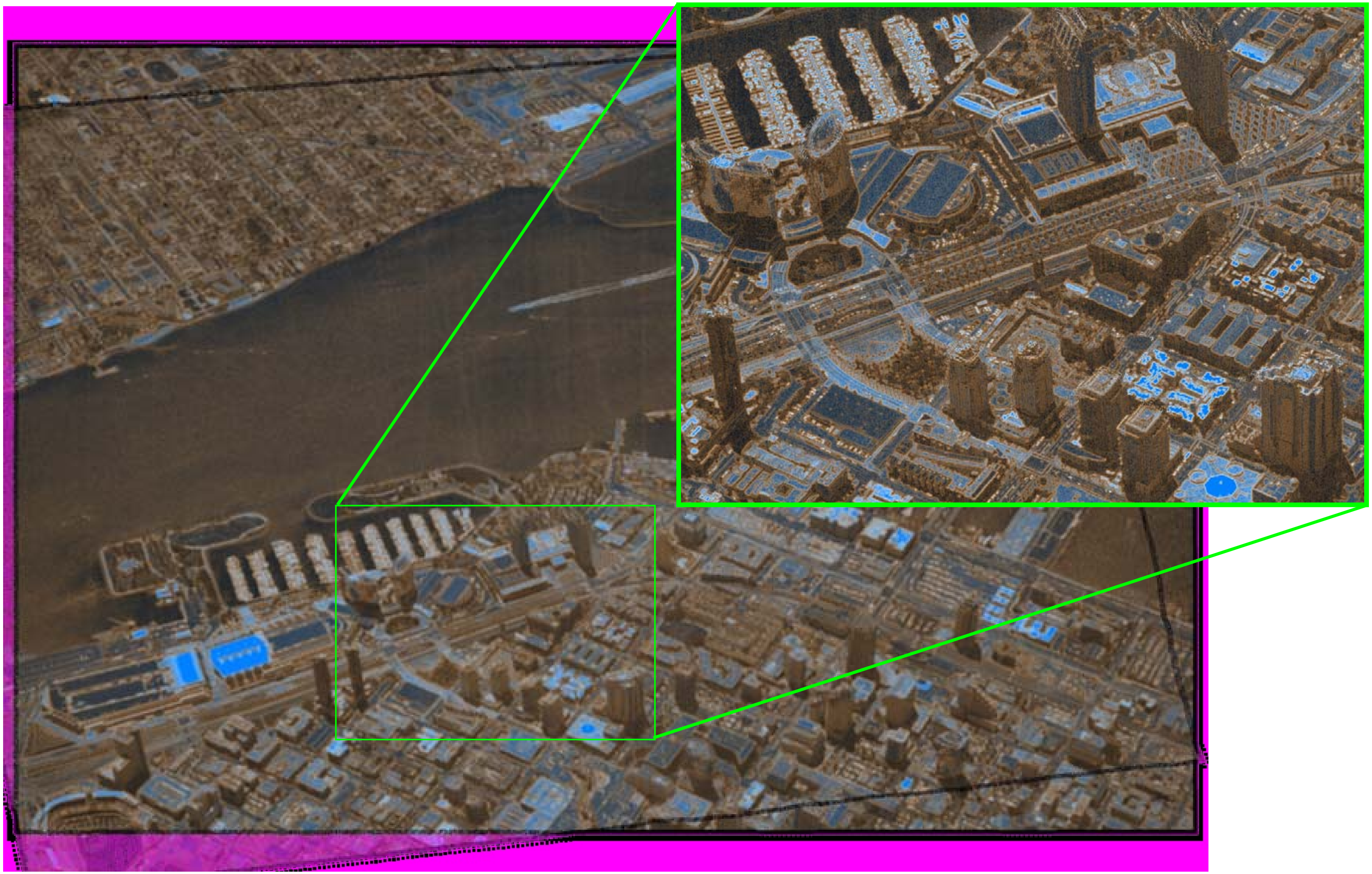
- **Goal is to unify: 1) image registration, 2) stereo dense correspondence, and 3) optical flow—*but make algorithm GPU friendly and progressive!***
- **Take any two images (not consecutive) and warp first to match second**
- **Provide subpixel accurate position per fine-resolution pixel**
- **Work despite camera motion, noise and artifacts, water shimmer, dense urban clutter, moving and shiny objects, parallax from moving view, etc.**

# Correspondence is computed for pairs of images from the video sequence

- Frames 10 and 31 from aerial video sequence:



# Selectively refining correspondence is robust, maps well to GPU/Cell acceleration

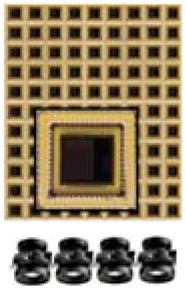


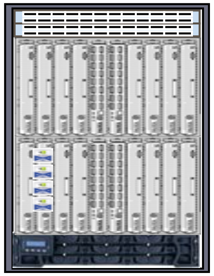
# Algorithm proceeds coarse-to-fine, matching and straightening for ~10 iterations per level

```
For level = 0 to LMAX {   // A variant of "multi-resolution optical flow"

    if level == 0: initialize to identity mapping
    otherwise: perform 4-8 mesh refinement from level-1 to level

    For iteration = 1 to 10 {

        // Matching motion per pixel:
        Move each source pixel to closest matching isoline in target
        -- clamp motion to fraction of a pixel, skip small gradients

        // Warp mesh straightening:
        Least-squares fit a local affine map per 5x5 neighborhood
        Update each pixel map position to the average of its
                positions in all the nearby affine maps

    }

}
```

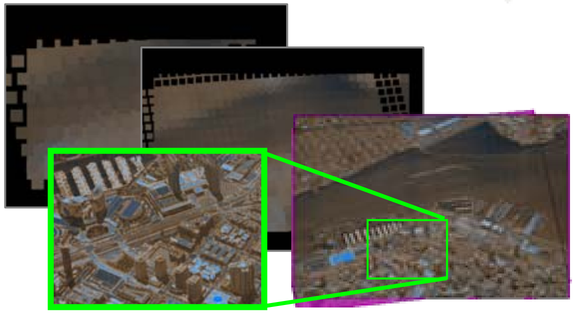
# The kernel benchmarks indicate 22 million pixels per second corresponded per GPU

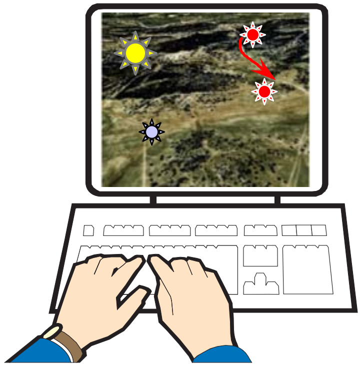
- **Proof-of-concept dense correspondence on aerial video**
- **Current acceleration: 21.9 Mpix/sec registered per GPU**
- **Details:**
  - **219 iterations per second on 1 Mpix**
  - **10 iterations total needed**
  - **Test performed on:**
    - **Single Nvida 8800GTX-based card**
    - **AGP 16x interface**
    - **Single threaded application**
    - **2.66 GHz quad-core Intel processor**
    - **4GB CPU RAM, 768MB GPU RAM**
    - **128 GPU cores, accessed via OpenGL ARB fragment programs**





**>4TFlop, ~5kw for 16 GPUs**



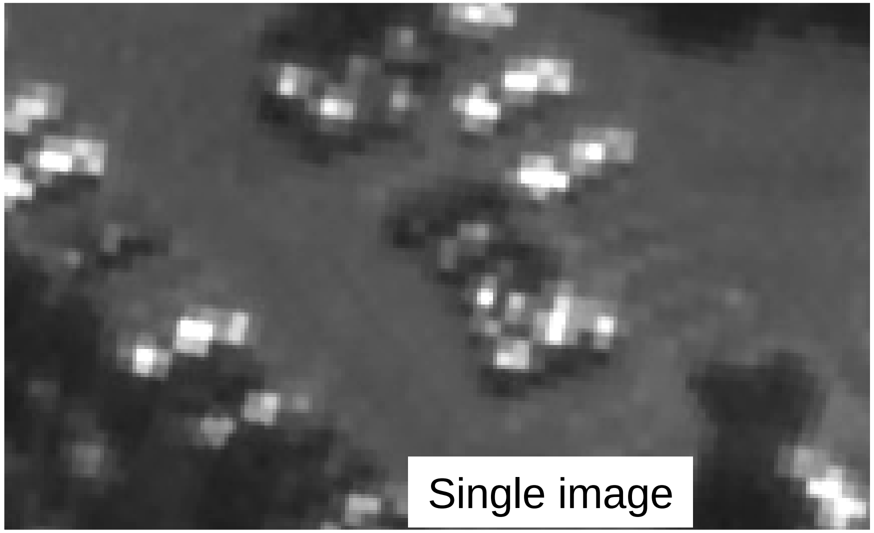


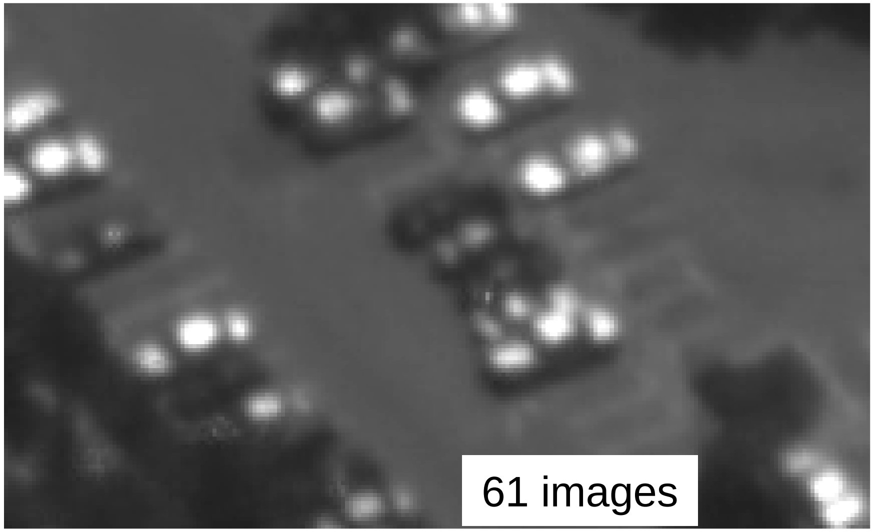
# The dense correspondence algorithm successfully stabilizes the imagery

- **Robustness is good overall**
  - **Image borders**
  - **water**
- **Issues with tall buildings, trees**
  - **Foreground/ background splitting**

# Dense correspondence enables resolution improvements

**Resolution can be improved by combining frames**
**→ proves subpixel accuracy of correspondence**



# Lessons learned and future directions

- **The dense correspondence algorithm is observed to:**
  - **Eliminate parallax and “false positives”**
  - **Work robustly for many aerial videos**
  - **Support resolution enhancement, 3D extraction, and mover detection**
  - **Run fast on a GPU implementation**
    - **Real-time 22 Mpix/sec per GPU**
  - **Fails for tall, thin structures and extreme viewpoint changes**
- **Future directions are to:**
  - **Improve correspondence algorithm to detect and honor depth discontinuities**
    - **Important for cluttered urban environments**
  - **Fully integrate GPU correspondence into ROAM tile infrastructure**
  - **Compare GPU to IBM Cell, FPGA and other high-performance hardware options**
  - **Demonstrate real-time 3D extraction, tracking**