Senscape: modeling and presentation of uncertainty in fused sensor data live image streams

Henry Dietz & Paul Eberhart

COIMG-392, 4:10PM, January 30, 2020

University of Kentucky
Electrical & Computer Engineering

University of Kentucky

Fused Sensor Data
Live Image Streams

- Collect data from multiple sensors, but construct and present a unified model
  - Imaging sensors: visible, thermal, etc.
  - Point property sensors: ultrasonic/laser distance meters, accelerometers, etc.
- Model may be complex, but want a visually simple presentation in real time
- Color 2D (or 3D) image streams

University of Kentucky

Fusion Isn't Perfect

- The sensor data "shape" isn't the same:
  - Spatial and tonal resolution/accuracy
  - Temporal resolution and synchronization
  - Dimensionality, from 0 to 3+ dimensions
  - Parallax, occlusion, & other misalignments
- Most systems try to hide imperfections
  - **What you don't know can hurt you**
  - Model the uncertainty and **present it as intrusively as befits the danger**

The Key Points

- Sensor fusion involves alignment & tracking
- Data structure/property contains:
  - Latest raw sensor data and alignment
  - Measured/computed property map with both value and confidence
- Rendering image to display
  - Alignment detection and update
  - Composition and confidence aging rules

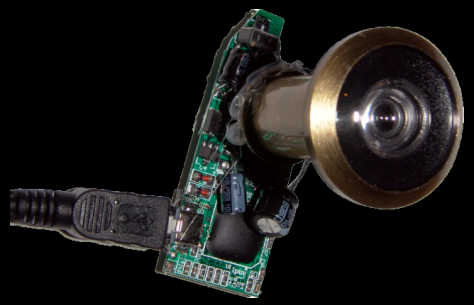
Structure Of This Paper

- Examines 4 systems, two old and two new:

  FireScape: guide firefighters through fires
  NodeScape: supercomputer maintenance
  KVIRP: visible+thermal live video testbed
  Wakam: visible+radar live video testbed

- For each system:
  - Describes system sensors
  - Overviews the processing
  - Provides a fused example

University of Kentucky



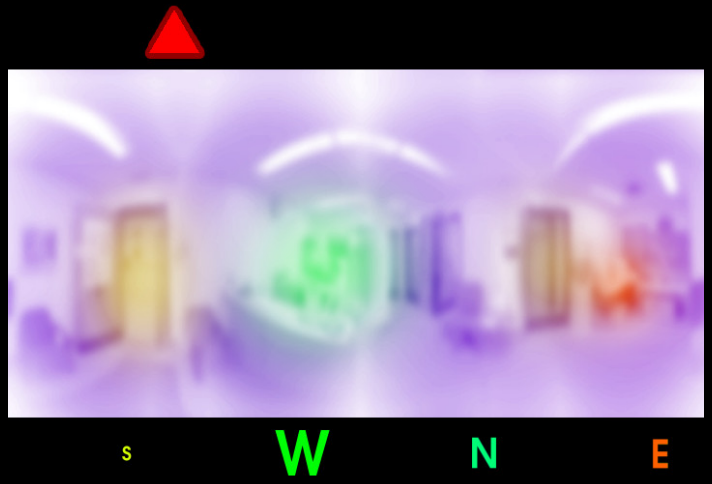
FireScape (2006)

- Bud Meyer, Bill Dieter, and Henry Dietz; hands-free low-cost IR imaging for firefighters
- Concept was sensor fusion and display in a face-mask-mounted display:
  - Three 180° door-peephole fisheyes
  - Diffuse wide-angle point thermal sensors
  - Ultrasonic point distance sensors
  - Compass (magnetometer)
- Never built, subsystems prototyped

University of Kentucky

FireScape

Simulated view →



- Three low-res 180° door-peephole fisheyes, stitched as monochromatic visible+NIR
- Temperature spots painted
  - Blue to red, with green for human bodies
  - Saturation decreases as sample ages
- Compass SWNE size distance to object; Triangle is way out pointer

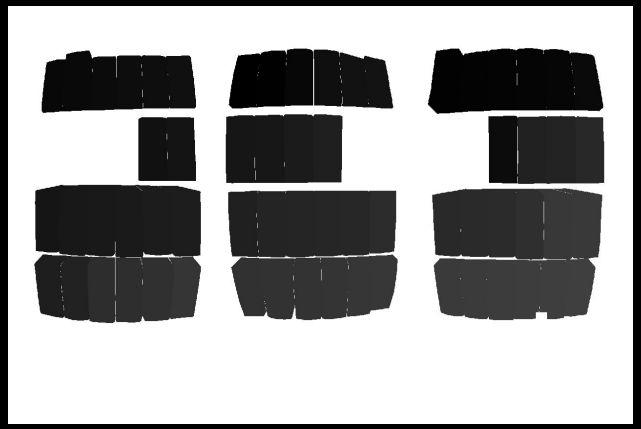
University of Kentucky

NodeScape (Dietz, 2012)

- Cluster supercomputers have many nodes...
  - Which node has what problem?
  - Make problem patterns obvious too
- Daemon process `epacsedon` monitors each node's sensors & sends UDP status updates
  - Temperature sensors, fan RPM, etc.
  - **Synthetic sensors**: e.g., load average
- `nodescape` collects status & creates display

University of Kentucky

NodeScape



- `nodescape` paints data on photo or system
  - Photo of actual cluster nodes
  - Key image IDs which pixels for each node
- ***Tints*** nodes minimum .. maximum value
- As values age, ***dither*** with magenta

University of Kentucky

KVIRP (2019)

- **KVIRP**: Kentucky's Visual / Infra Red Painter, produces 360° visible+thermal video
  - Up to 3008x1504 @ 30 FPS
  - Thermal data 0°C-80°C ±2.5°C
- Open-source design, build for under $150

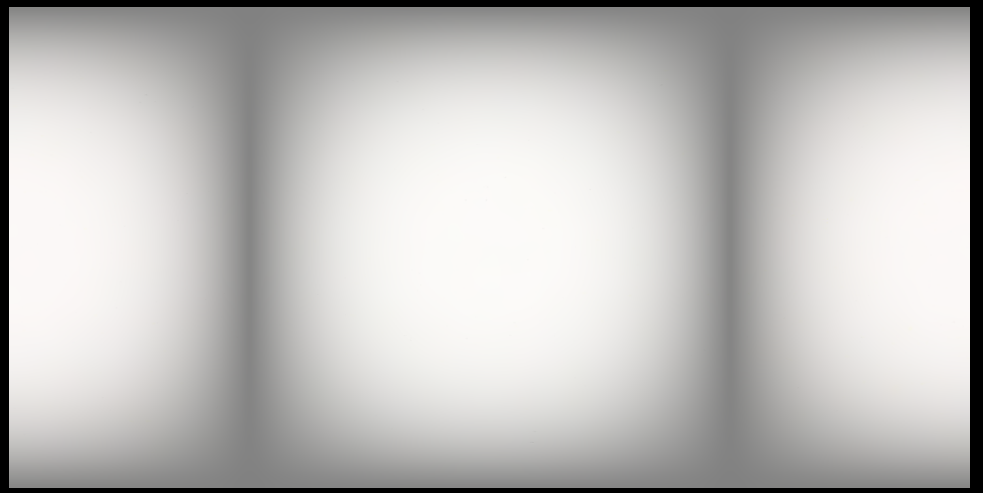
University of Kentucky

KVIRP Insta360 Air

- $100 Insta360 Air dual-fisheye camera
  - Two back-to-back >180° *f*/2.4 fisheyes
  - Sampled together, 3008x1504 @ 30FPS
- Images must be stitched for 360° projection
  - Calibrated stitch can be quite good, but there's about 30mm parallax error (3D-printed chassis hides within that)
  - KVIRP stitching reduces contrast to show image region subject to parallax error

University of Kentucky

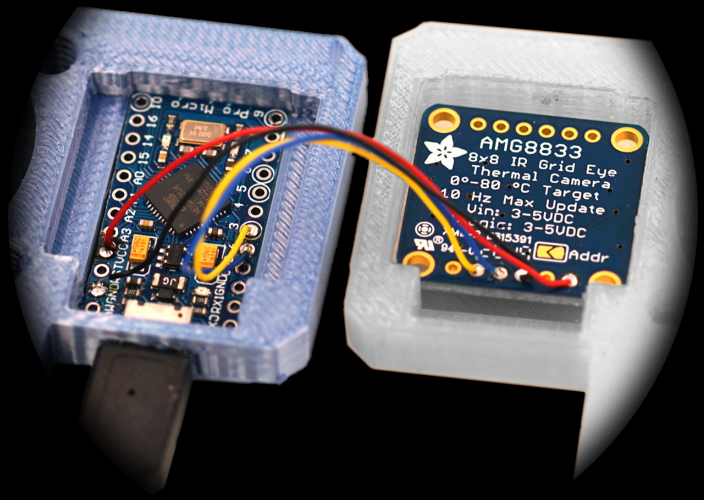
Insta360 Air Stitching



- With calibration and no near objects, could stitch essentially perfectly… but let’s not
- Confidence in the stitched image spatially varies, especially for fisheye views

University of Kentucky

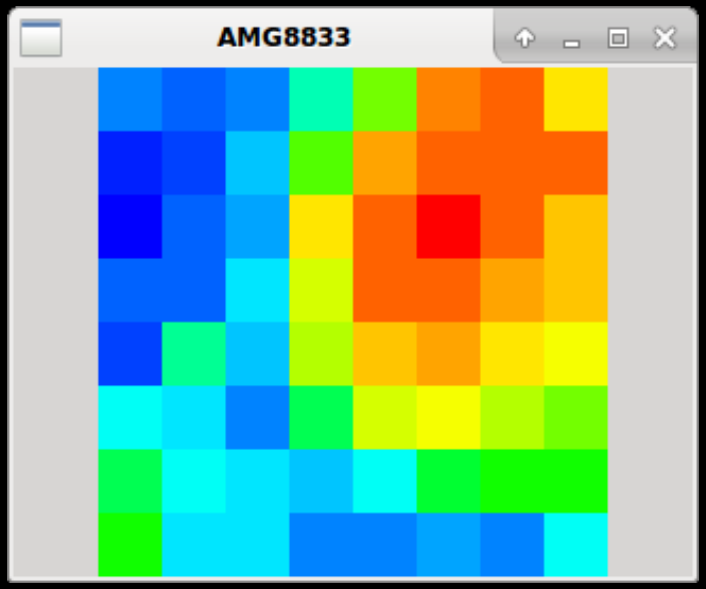
KVIRP AMG8833

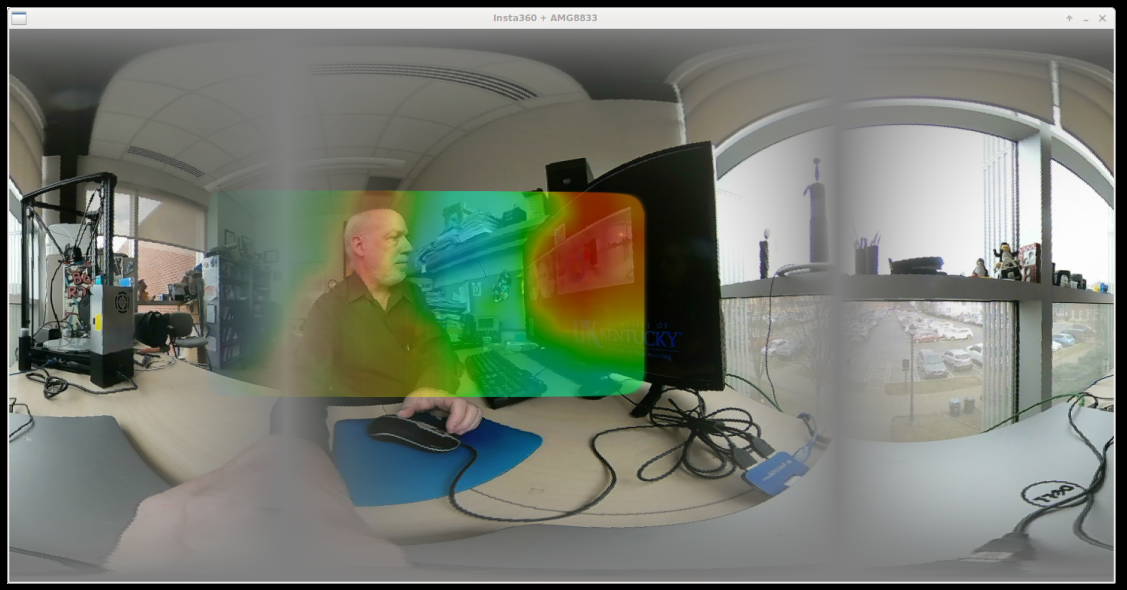


- $40 **Adafruit AMG8833 Thermal Camera**
  - 0°C-80°C in 0.25° steps, accurate ±2.5°C
  - Maximum sample rate of 10Hz
  - 8x8 pixels, approx. 60° view angle
- $3 **ATmega32U4 Pro Micro** used for control
- Paint blue .. red temperatures on 360° image
  - Painted temperatures move with image
  - Confidence shown by saturation

University of Kentucky

KVIRP Fusion Example





- Motion tracking is done by computing feature alignment with previous image
- Poor matches or big movements destroy confidence of painted attributes

University of Kentucky

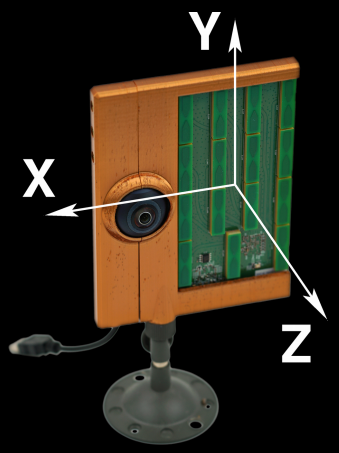
Wakam (2019)

- **Wakam**: WAlabot Kentucky cAMera produces 360° visible+radar video
  - Like KVIRP, up to 3008x1504 @ 30 FPS
  - Radar target angle, depth, signal strength
- Open-source design, build for about $150

University of Kentucky

Wakam Radar Sensor

- $50 **Walabot Creator**
  - 15 linearly-polarized antennas, 3.3-10GHz
  - Low-level API & SDK for Python and C++
  - USB interface (raw, via API only)
- Creator is held in the reference orientation and for each “tracked” object reports:



Amplitude: signal strength
X: R x sin(θ)
Y: R x cos(θ) x sin(Φ)
Z: R x cos(θ) x cos(Φ)

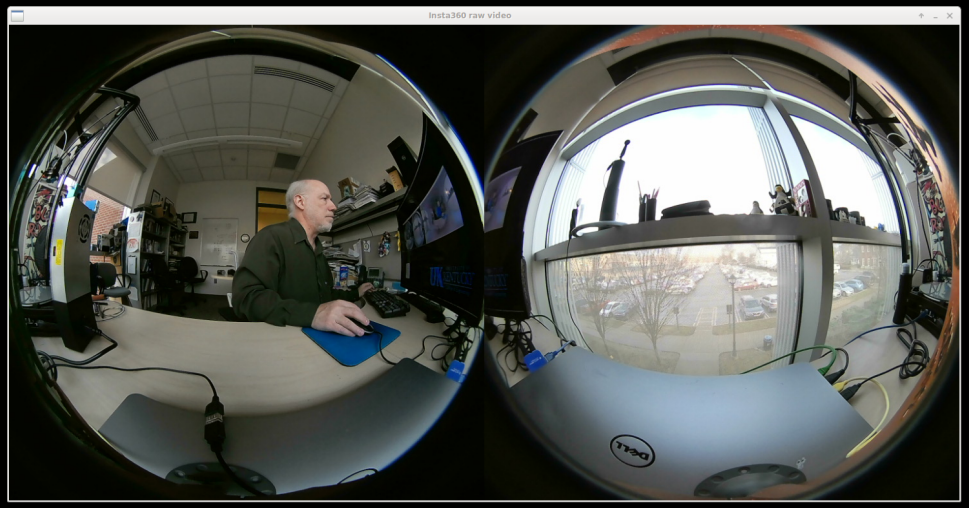
University of Kentucky

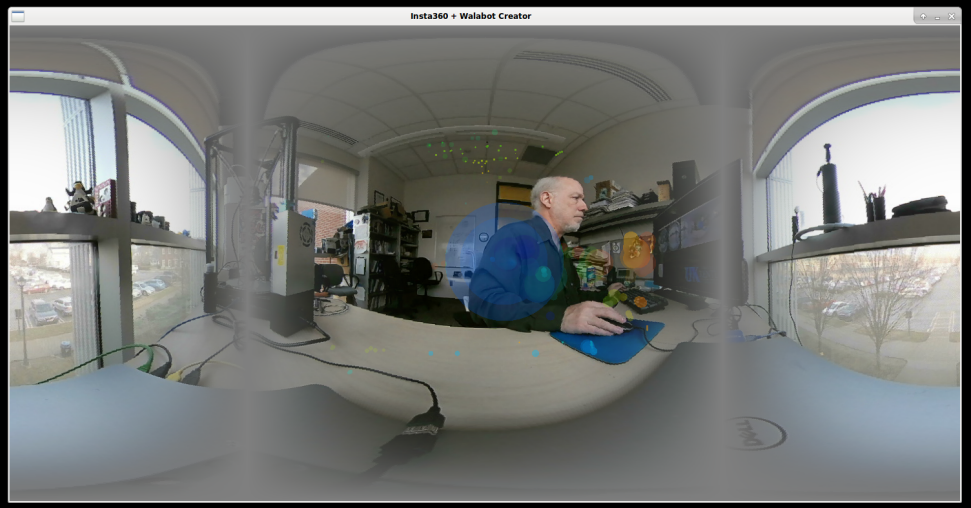
Wakam Processing

- Insta360 Air stitched image is used as base
  - Color image with saturation confidence
  - Also used to track motion of wakam
- Each radar target painted as spherical tint:
  - Aligned position derived from X, Y, Z
  - Color tint is Z: near..far is blue .. red
  - Diameter based on amplitude
  - Confidence initially scaled by amplitude, but can add and decays with motion; displays as saturation of tint

University of Kentucky

Wakam Fusion Example





- The Walabot Creator tracking is not very consistent; lots of stray targets
  - High signal-strength targets more reliable
  - Good correlation on Z depth
- Walabot API/SDK is the weak link here...

University of Kentucky

Open Source KVIRP & Wakam

- Hardware design, software, & 3D models
- Similar C++ code for KVIRP & Wakam
  - `kvirp.cpp` ~15kB, `wakam.cpp` ~18kB
  - **OpenCV**, version 4.2.0-dev
  - Wakam also uses **Walabot Linux SDK**
  - Custom alignment, defishing, painting
  - Runs as 2 processes, with shared buffer, to manage async sensor data updates

University of Kentucky

Summary:
What Makes A Senscape?

Presentation of **fused sensor data** as a **live image stream** in which:

1. There is a base image for alignment
2. Each property has an aligned history
3. Each property maps values, confidence
4. Rules govern confidence update, display

Make confidence as obvious as it is important.

University of Kentucky

Conclusions

- Painting with tracking & aging works
- Uncertainty of a property value ***is*** a property.
- Two open-source testbeds developed:
  - KVIRP, for visible + thermal imaging
  - Wakam, for visible + radar point targets

University of Kentucky