

# Sony ARW2 Compression: Artifacts And Credible Repair

Henry Dietz

*VIPC-227, 3:30PM February 17, 2016*

University of Kentucky  
Electrical & Computer Engineering

# Sony ARW Version 2 “cRAW”

- Raw formats encode *uncooked* sensor data
- **ARW** is Sony's **Alpha RaW** file format
- ARW used to use **packed 12-bit pixel values**
- Since 2007, “cRAW” **lossy compression** in:

**FF E:** NEX-VG900; ILCE-7, 7R, 7S, 7M2, 7RM2, 7SM2

**FF A:** DSLR-A850, A900; SLT-A99

**APS-C E:** NEX-3, 5, 5N, 5R, 6, 7, C3, F3, VG20, VG30;

ILCE-3000, 3500, 5000, 5100, 6000; QX1

**APS-C A:** DSLR-A450, A500, A550, A560, A580, A700; SLT-A33,

A35, A37, A55, A57, A58, A65, A77; ILCA-77M2

**Cyber-shot:** DSC-RX100, RX100M2, RX100M3, RX100M4, RX10,

RX10M2, RX1

# Why Would Sony Do That?

- Compressed raws use **only 8 bits / pixel**
  - **Faster** camera operation, SD card writes
  - Effectively **1.5X memory capacity**
  - Encoding allows random pixel access
- **Computationally cheap encode / decode**
- Introduced as an **option** & *few* complained, so Sony dropped packed 12-bit

# Why Did Sony Stop Doing It?

- Lenses with **higher microcontrast**
- Camera sensor data went from 12→14 bits, so **compression ratio increased**
- Usable sensor **dynamic range increasing**
- **ISO-less exposure** concepts led people to **underexpose by up to 6EV & boost in post**:
  - EI 2015: our **ISO-less?** paper
  - DPReview: **ISO-Invariance** articles
- **Unhappy users**, especially for ILCE-A7RM2

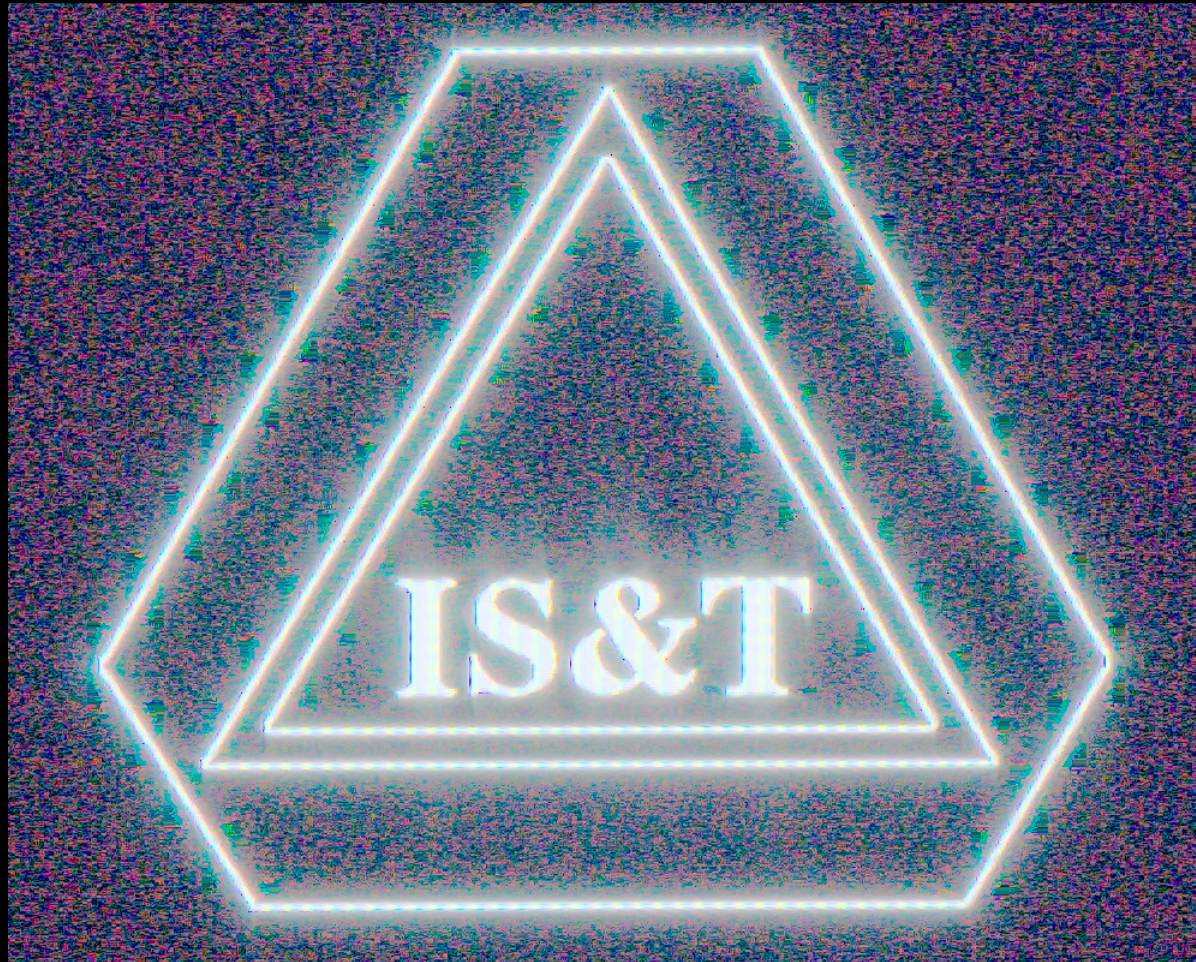
# Why Did Sony Stop Doing It?

- Lenses with **higher microcontrast**
- Camera sensor data went from 12→14 bits, so **compression ratio increased**
- Usable sensor **dynamic range increasing**
- **ISO-less exposure** concepts led people to **underexpose by up to 6EV & boost in post**:
  - EI 2015: our **ISO-less?** paper
  - DPRReview: **ISO-Invariance** articles
- **Unhappy users**, especially for ILCE-A7RM2
- **Sony didn't stop, but added 16-bit raw**

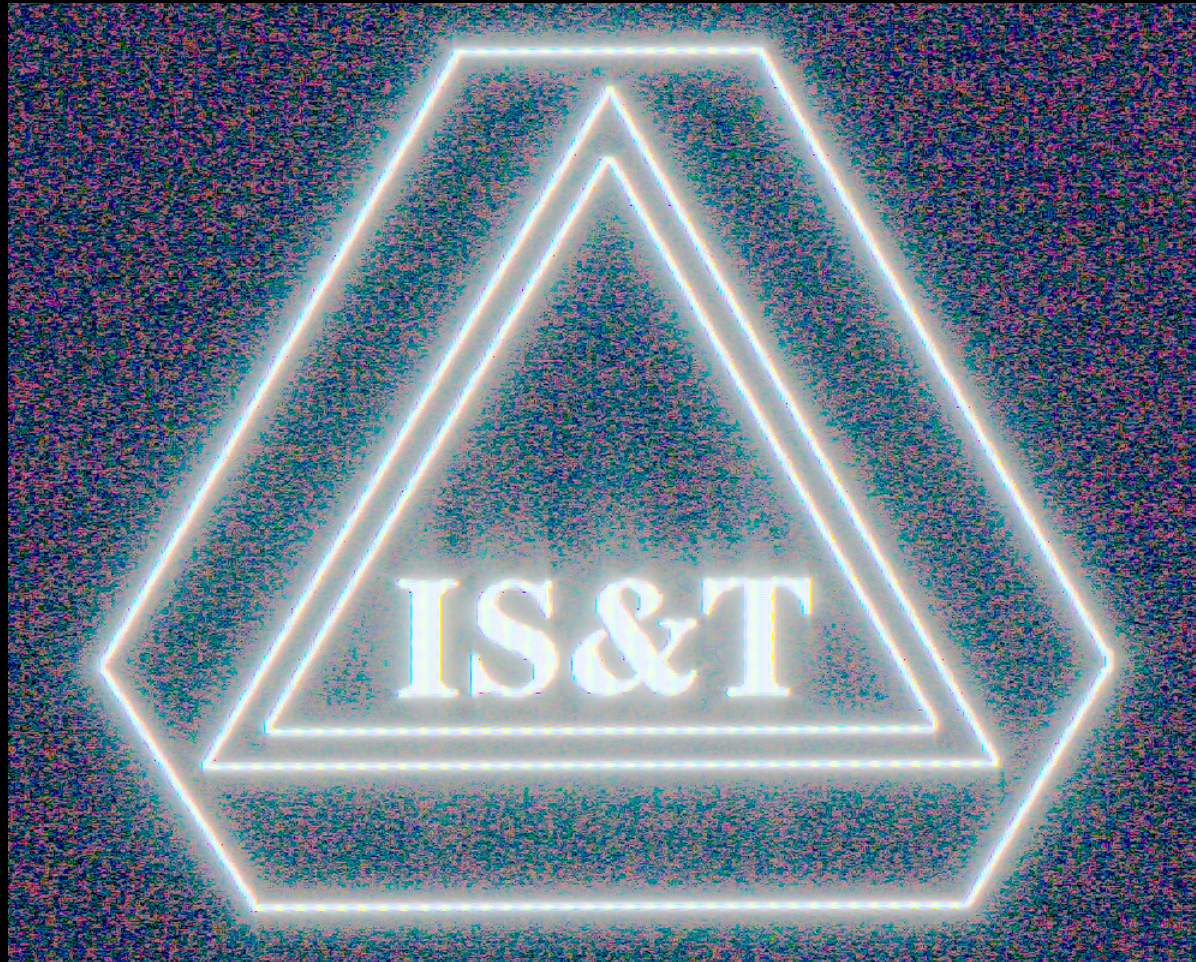
# The Problem: Boosting This



# Reveals This



# Rather Than This (16-bit ARW)





# Lossy Compression, 1<sup>st</sup> Step

- Start with 14-bit raw data
- Tone map to approximate log encoding:
  - Add fixed black offset (e.g., 512)
  - Reduce value to 11-bit by 5-segment linear mapping specified by when value step goes from 1→2, 2→4, 4→8, 8→16, & 16→32
  - Step change thresholds recorded in EXIF

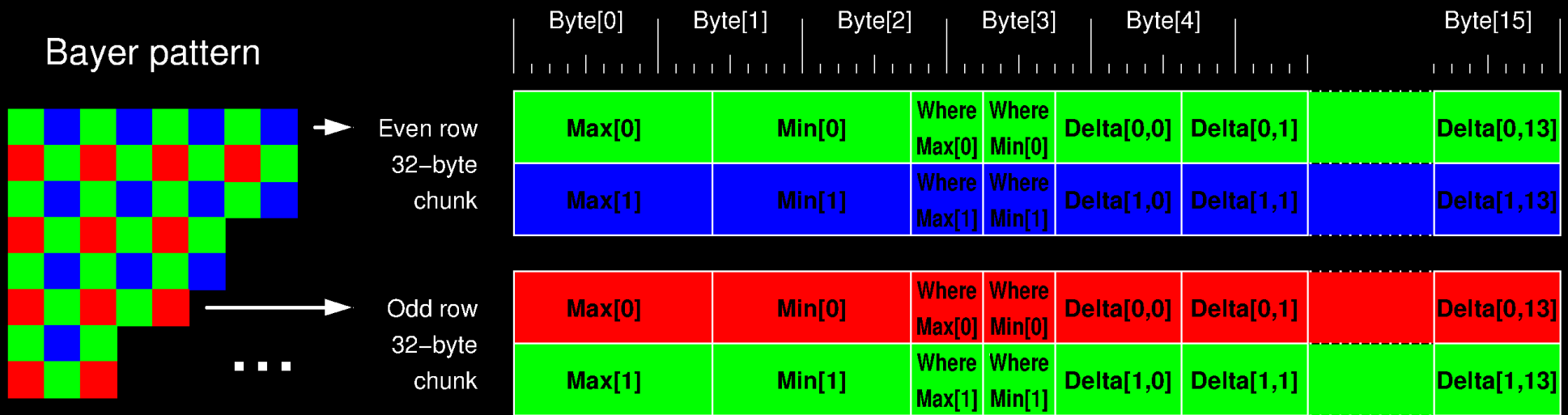
*Really not very different from other cameras...*

# Lossy Compression, 2<sup>nd</sup> Step

- Work on a 32-pixel horizontal strip of pixels
- Break strip into interleaved 16-pixel strips
- For each interleaved 16-pixel strip, record:  
Max, Min, MaxPos, MinPos, 14 Deltas...
- **Delta is scaled to fit 7 bits by dropping least significant bits of actual delta value**

*Very different from most other cameras!*

# Lossy Compression, 2<sup>nd</sup> Step



- The big problem is the **7-bit Deltas**...  
Suppose Min..Max is 1000..1518; mapping 518→127 forces **3-bits dropped**, to step of 8
- Note that **Max  $\neq$  Min+8\*k** for any integer *k*

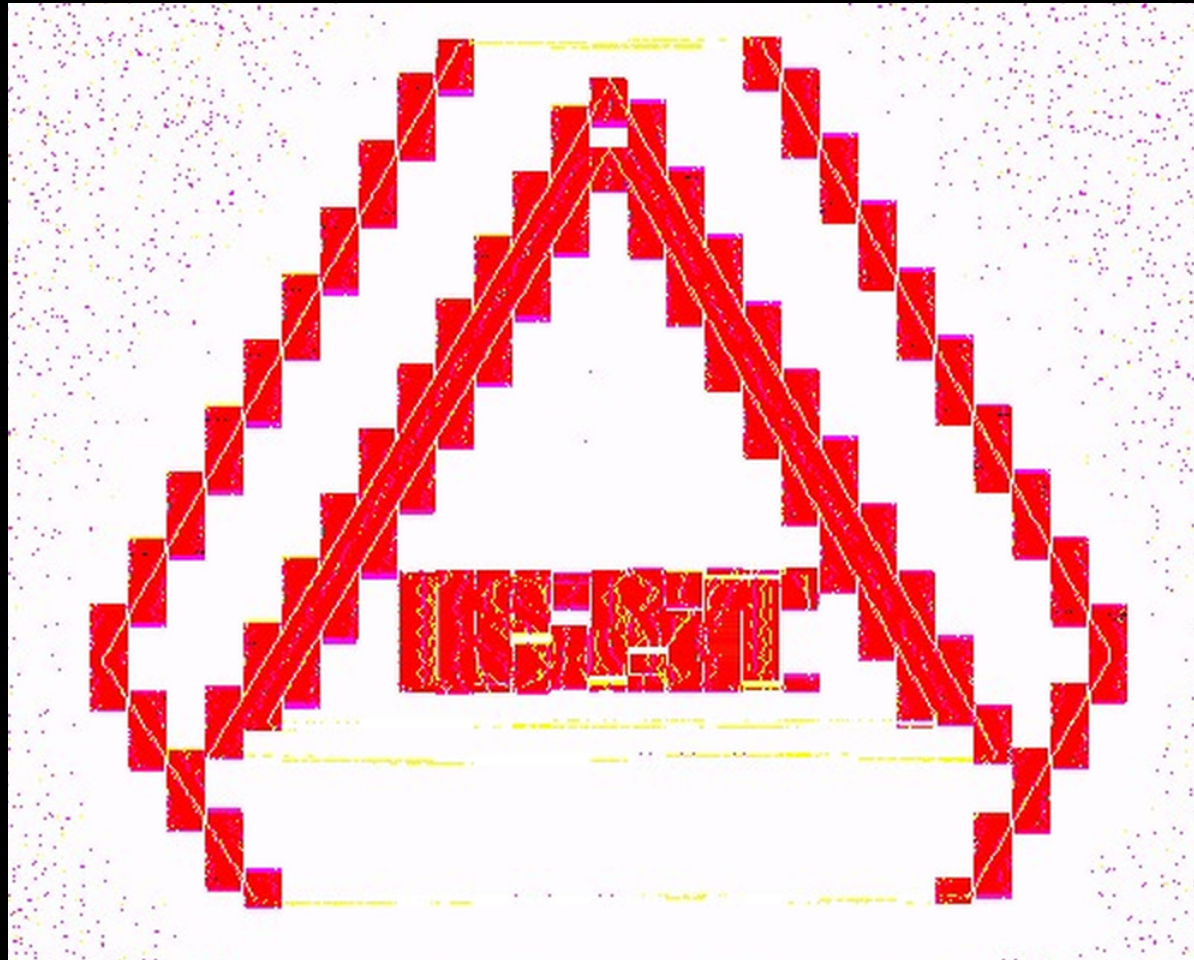
# Credible Repair With KARWY

- **KARWY**: U. of **KY ARW** repair raw *wrapper*
- Tried over 75 different repair algorithms
- Algorithm implemented behind WWW form:
  1. Construct an error model for each pixel
  2. **Adobe DNG Converter** packages as DNG
  3. Smooth initial pixel value estimates
  4. Texture synthesis to find value in range
  5. Final pixel value adjustment (*add* noise)

# KARWY Error Model

- Done using decode logic from dcraw
- Compute 11-bit value range:
  - Min, Max give precise 11-bit values
  - $\Delta < 128$  reconstructs precise 11-bit
  - Range of others based on delta truncation
- Map 11-bit value into range of 14-bit values
- Extend range slightly to allow for noise

# KARWY Bits Valid Map

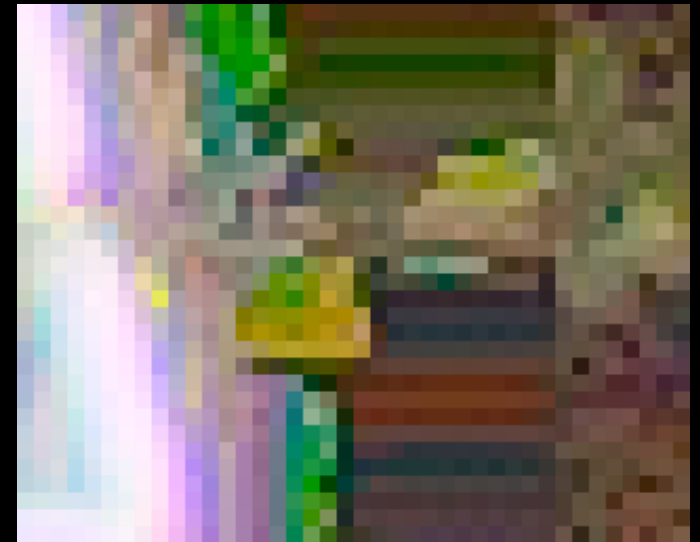


# Why Does KARWY Use Adobe DNG Converter?

- At writing of KARWY, ARW2 only provided a lossy compressed format... so wrap as DNG
- **Generating DNG directly gave inconsistent interpretations using various DNG editors**
- Adobe DNG Converter used to make wrapper:
  - **Builds all the magic fields Adobe wants**
  - **Does NOT preserve pixel data:  
both values and image size often wrong!**

# KARWY Smoothing

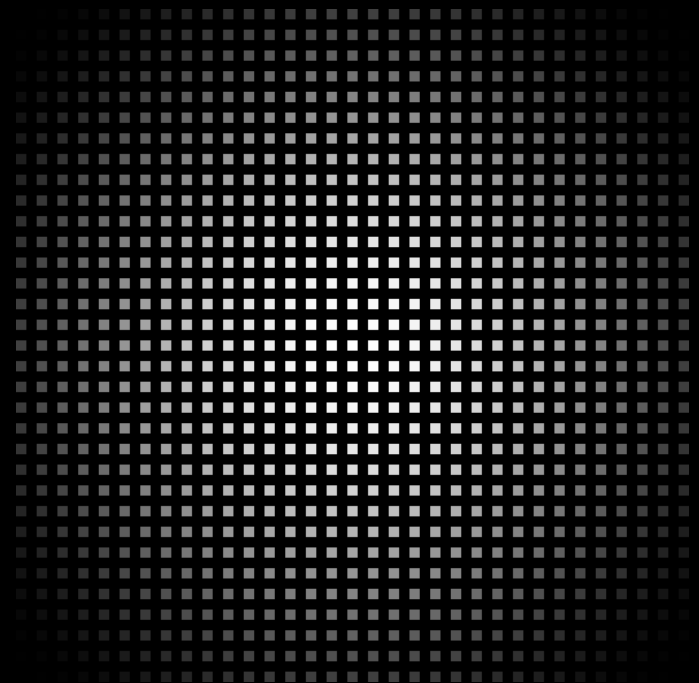
- Not really smoothing pixel values...  
*smoothing initial pixel value estimates*
- Optional % bad, near bad, other
- Removal of “Blondie”  
(parallel lines)  
artifacts:



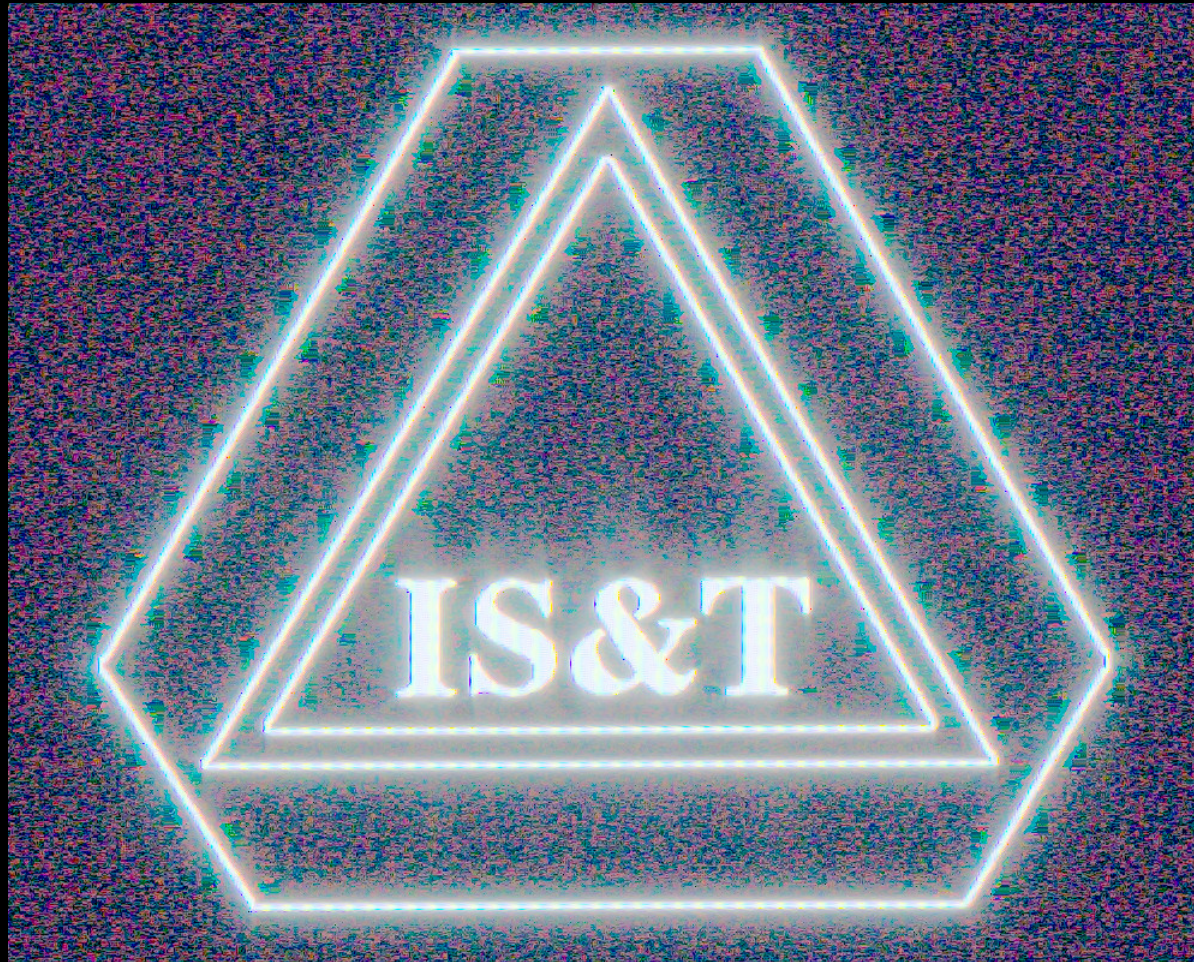


# KARWY Texture Synthesis

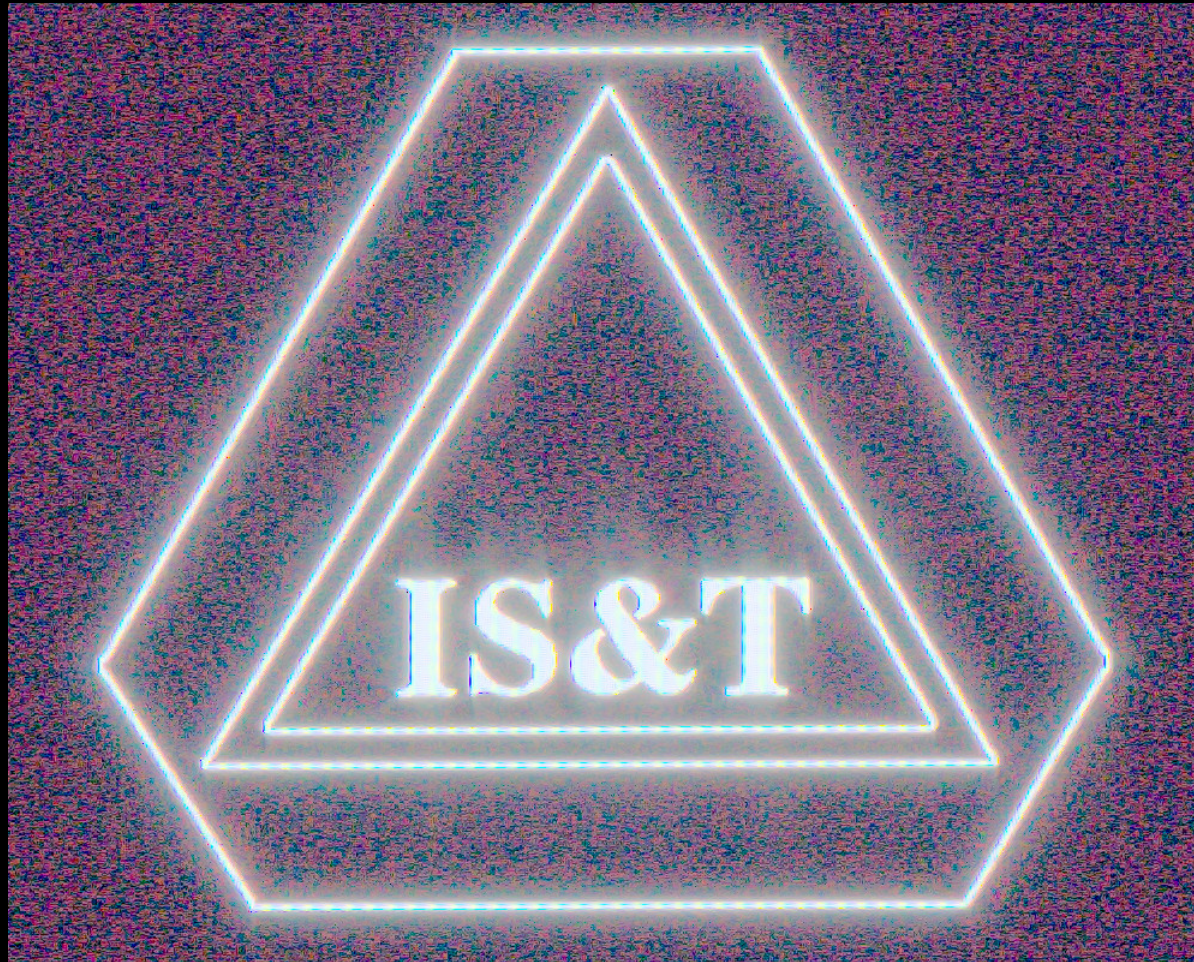
- Search for similar pixel environment in 1089 same-color positions
- Weighted sum:
  - Bits valid
  - Value range overlap
  - Distance weighting
- Constrained to range



# Artifacted Example



# Repaired By KARWY



# Rishi Sanyal's DPReview ILCE-7M2 ISO Invariance Test



# ISO Invariance KARWY

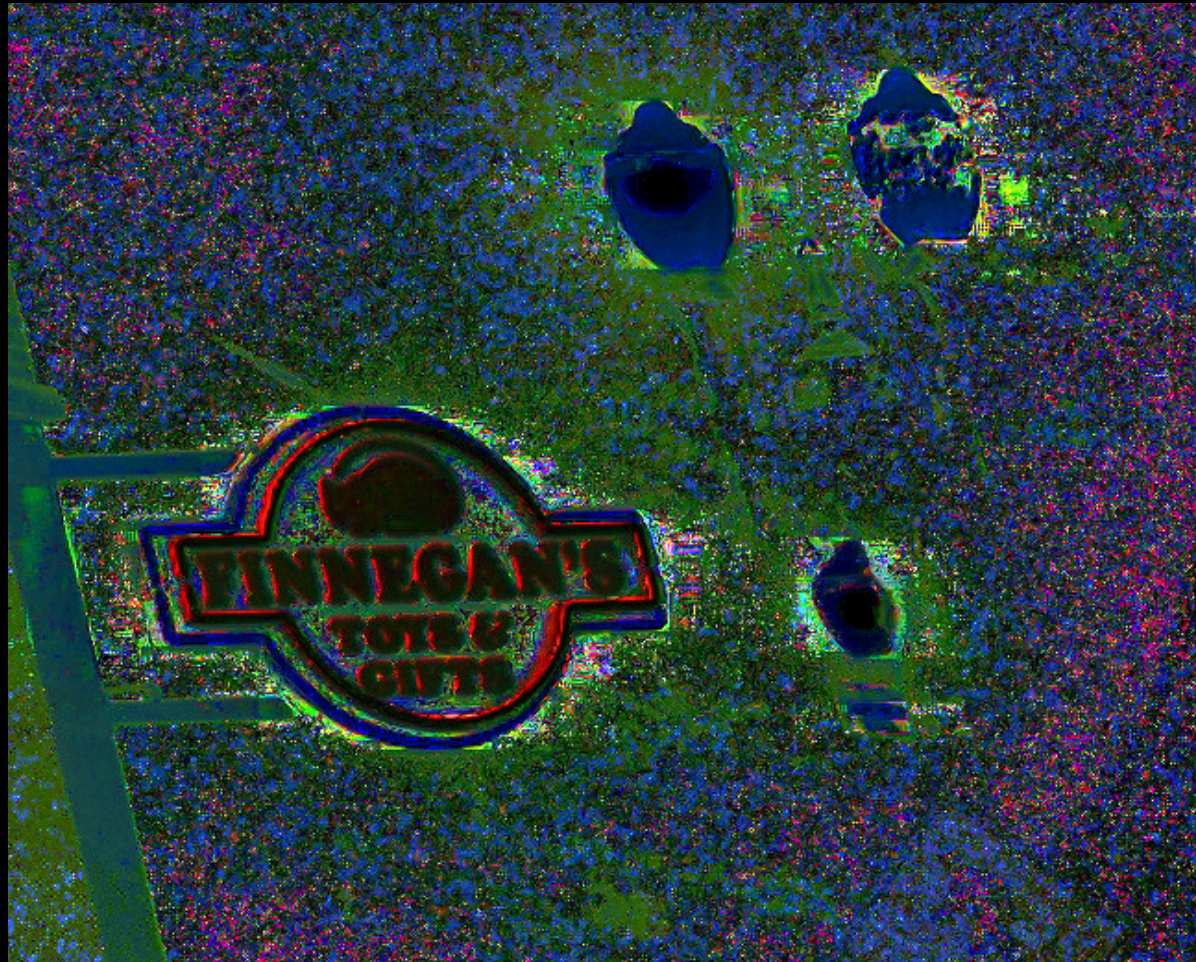


# ISO Invariance KARWY Smooth



# ISO Invariance

## Enhanced Repair Difference



# Matti Koski's ILCE-7 Star Trail From RawDigger Article





# Star Trail KARWY

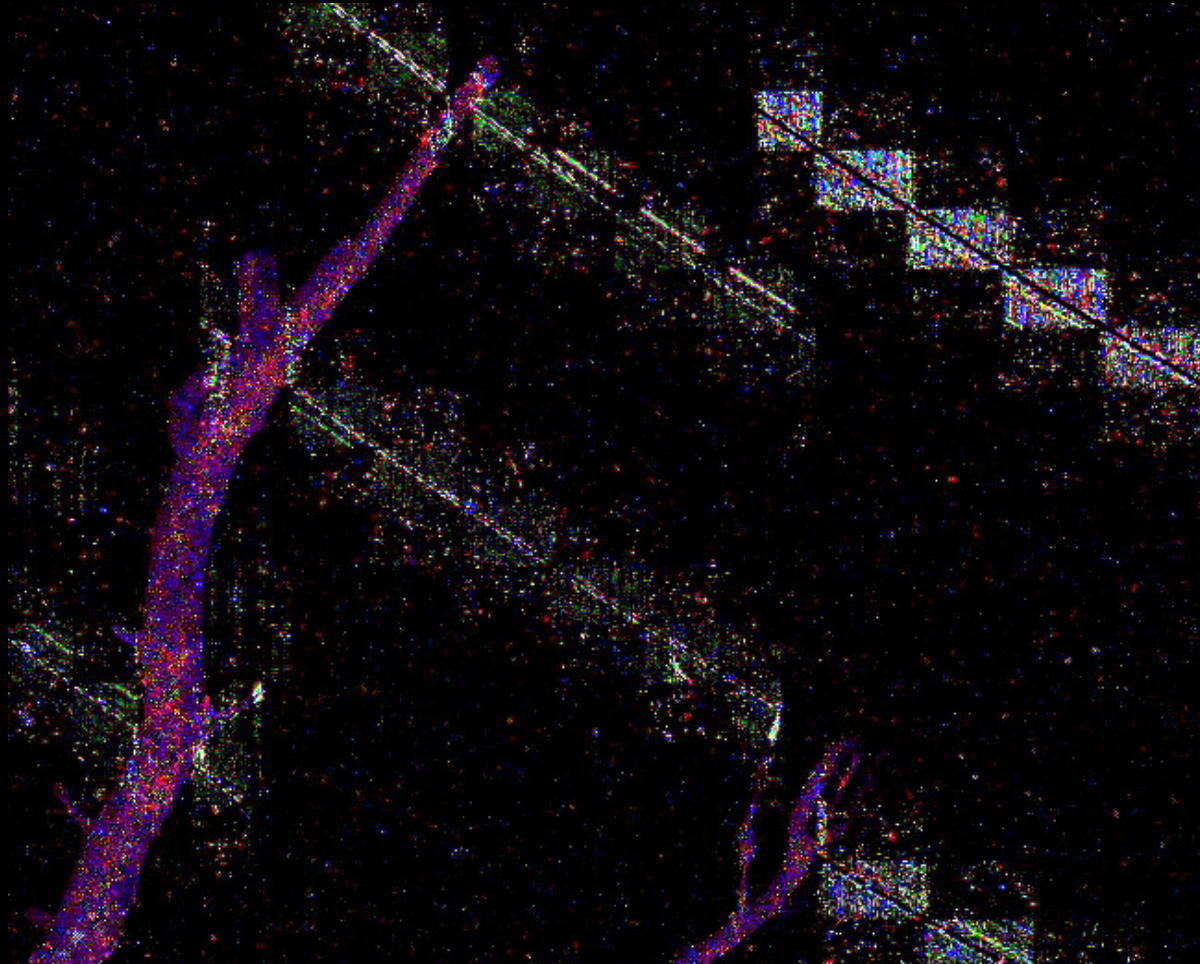


# Star Trail KARWY Smooth



# Star Trail

## Enhanced Repair Difference

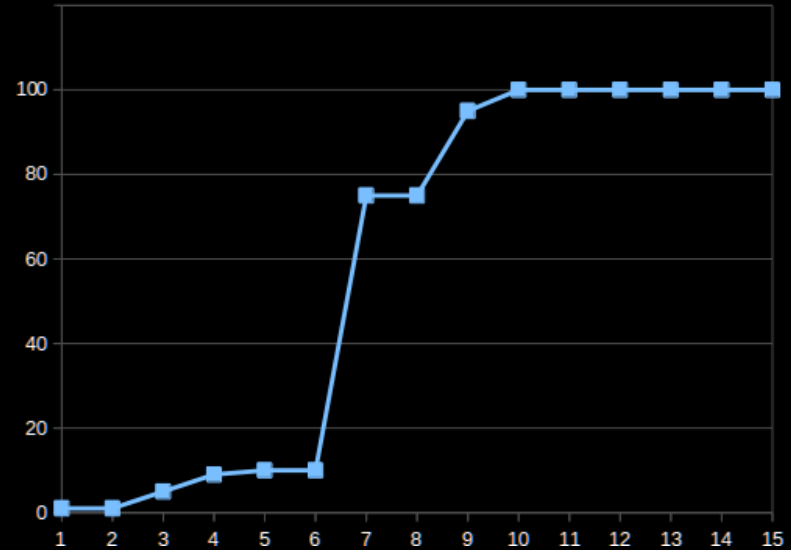


# Repair Quality Evaluation

- It's pretty good, right?

# Repair Quality Evaluation

- It's pretty good, right?
- KARWY WWW form allowed voting; >200 users, only 15 voted:
  - 6/15 gave perfect scores
  - 6/15 gave scores below 20%
- Most low scores when *no artifacts to repair*, one due to Adobe DNG Converter problem



# Conclusions

- Adobe DNG Converter changes raw data!
- Credible ARW repair by new algorithm:
  - Construct pixel value range error model
  - Use texture synthesis to refine values
- Artifacts well understood, poorly recognized
- Might be possible to improve dynamic range of *any* raw using this repair algorithm...

