# Refining raw pixel values using a value error model to drive texture synthesis

Henry Dietz

*IPAS-084, 3:50PM January 31, 2017*

University of Kentucky
Electrical & Computer Engineering

University of Kentucky

# Related Work

- **Adobe DNG Converter** changes pixel values, presumably in applying color / black point
- **RawImageClearer** started as Avisynth plugin **RemoveGrainHD**, uses *smart median filter*
- Dabov, et al, *image denoising using block matching and 3D filtering*
- Many *image stacking* programs

University of Kentucky

# The Idea: "raw" Repair



- "raw" means "uncooked" or "unprocessed" – but if data was corrupted, why not **repair** it?
- Fuji X10 "white orbs" blooming $\Rightarrow$ **DeOrbIt**
- Sony ARW compression artifacts $\Rightarrow$ **KARWY** (KARWY also reduces noise?)

University of Kentucky

# KREMY

- **KREMY** (pronounced "creamy") is KentuckY Raw Error Modeler
- Goal is not improved rendering, but removal of value errors introduced by noise, etc.
- Refines uncompressed DNG data using:
  - Texture synthesis *within error bounds*
  - An empirically-determined error model (KARWY models lossy ARW compression)

University of Kentucky

# Empirical Error Models

- For each pixel site, probablity distribution for possible true values given value recorded
  – *too expensive to construct*
- For each recorded color channel value, probability distribution of possible true values
  – *used in `tik` TDCI software*
- For each recorded color channel value, minimum and maximum viable true values (treat raw as 4 channels)

University of Kentucky

# Image Stacking Error Model

- Model is 4 channels x $2^{16}$ values x (min, max)
- Stack 2 or more DNG raws:
    1. Constant scene, similar exposure settings
    2. Adobe DNG Converter `-u` to make DNGs
    3. Compute (min, max) for each pixel site
    4. Vote for (min, max) values in model
    5. Sum votes to establish reliable (min, max)
    6. Interpolate to smooth, monotonic, model

# KREMY Error Model

- Stacking was impractical – and inconsistent
- Models based on a single image:
  - Change in patch standard deviation
  - Similarity range *in evenly-shaded patches*
- Model is 4 channels x $2^{16}$ values x (min, max), processed to force monotonicity

University of Kentucky

# Enhancement Algorithm

- Re-uses some code from KARWY, but:
  - No smoothing (KARWY does 2 types)
  - Different computation of error model
- Texture synthesis based on:
  - Finding up to 1089 similar pixel sites (sort-of like stacking up to 1089 images)
  - Similarity determined by 3x3 block having all pixels within errors bounds of 3x3 block around this pixel

  ...

University of Kentucky

# Enhancement Algorithm

- – Similarities weighted by distance in spiral search order (which can end early)
- – New pixel value always within error bounds
- Minor adjustments are (optionally) made to final pixel values to approximate
  - – Original average brightness
  - – Original average local contrast
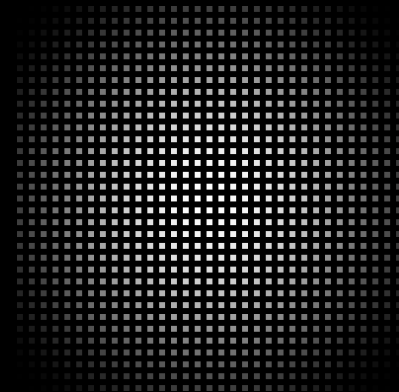
# Overall Impact Of KREMY

- Behaves most like image stacking;
  primary effect is improving SNR
- Does not posterize (unlike median filtering)
- Texture synthesis also enhances textures;
  edges are given more consistent appearance
- All changes are held within error bounds;
  all changes are quite subtle
  (and really hard to see on projected slides)

University of Kentucky

# Base ISO APS-C Bayer DSLR
## Canon EOS Digital Rebel XT @ ISO 100
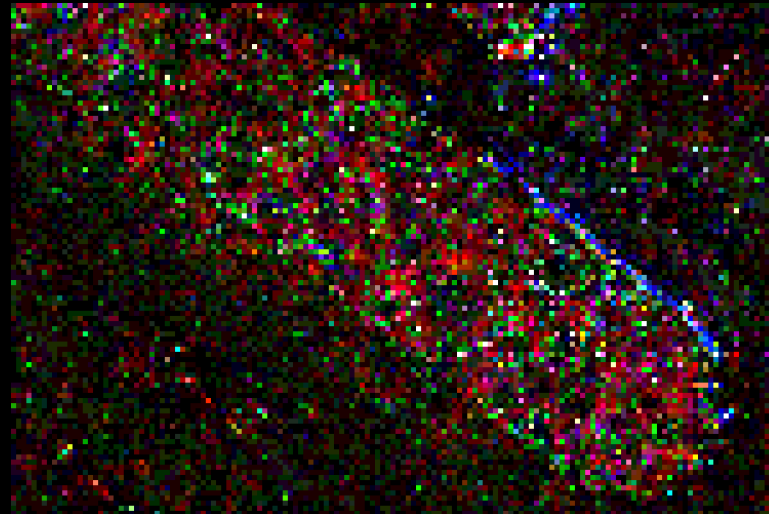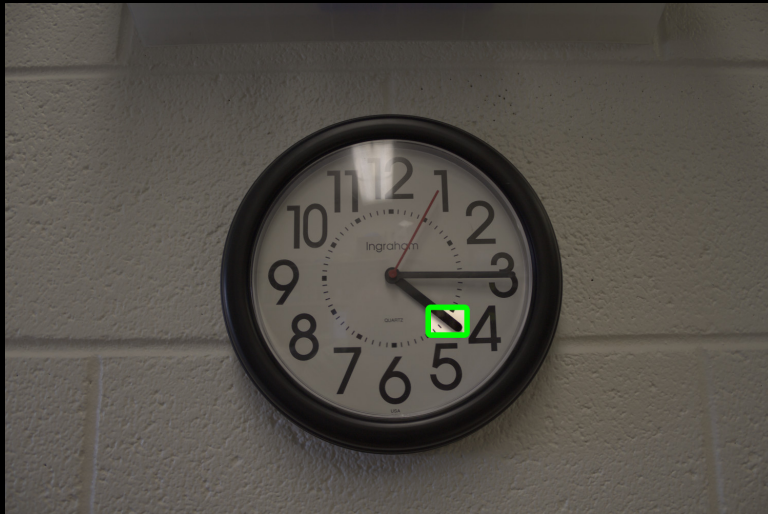
# Base ISO Compact Bayer
## Canon PowerShot S70 @ ISO 50

# High ISO APS-C Bayer EVIL
## Sony NEX-7 @ ISO 1600
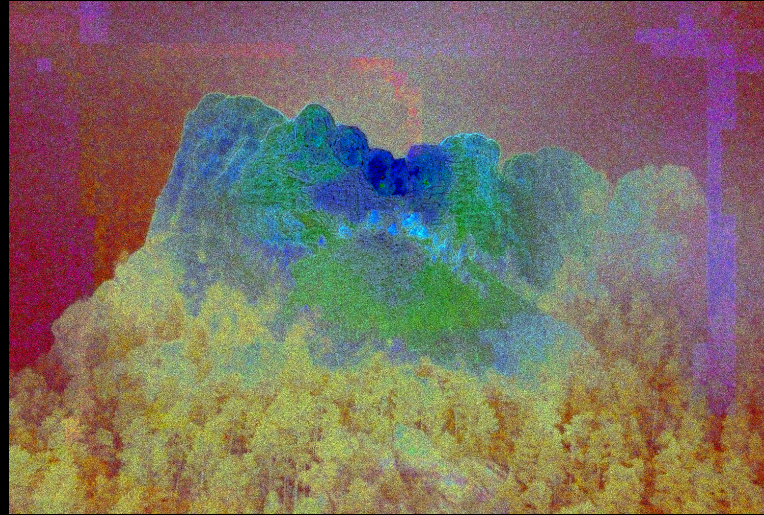
# High ISO APS-C Bayer EVIL
## Sony NEX-7 @ ISO 1600 (badly underexposed)
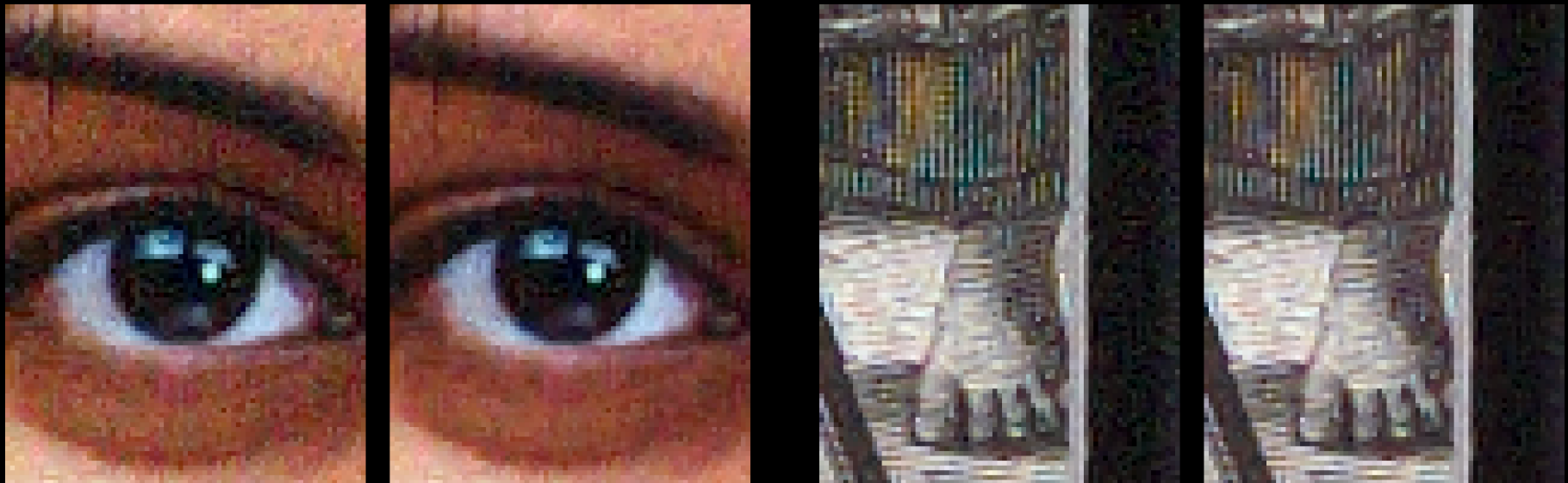
University of Kentucky
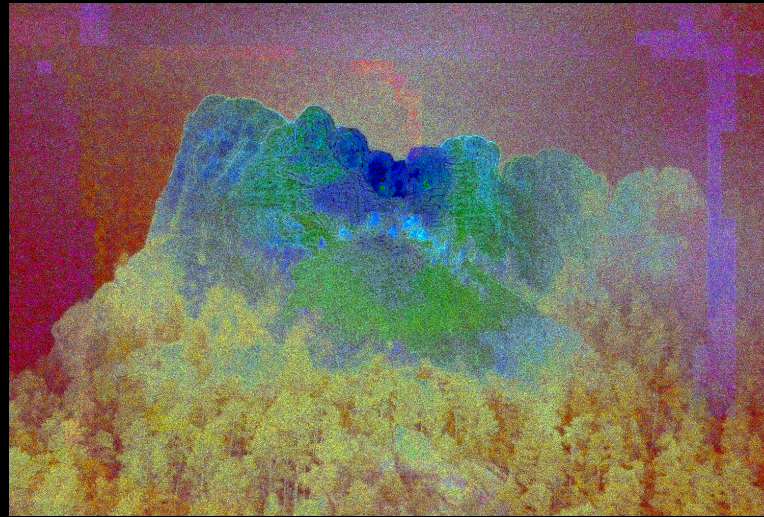
# Medium ISO MFT Bayer
## Olympus E-M1 Mark II @ ISO 400

# High ISO FF Bayer DSLR
## Nikon D810 @ ISO 1600



- Particularly effective – even reduces moiré

University of Kentucky

# Base ISO Cell Phone
## Apple iPhone 7 @ ISO 20

# Base ISO Non-Bayer (CMYG)
## Canon PowerShot G1 @ ISO 50



University of Kentucky

# Base ISO Non-Bayer (RGBE)
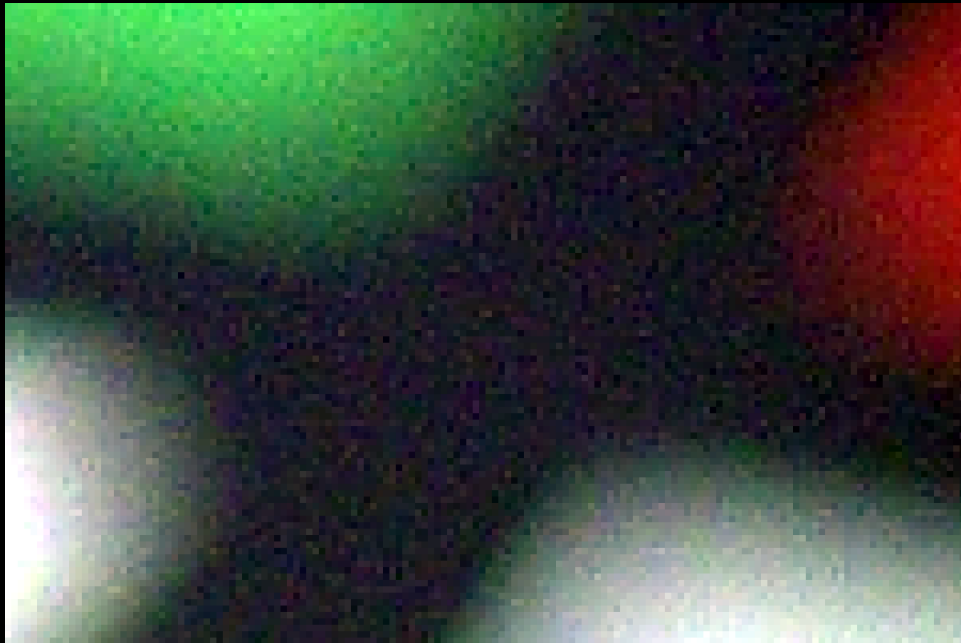## Sony DSC-F828 @ ISO 64



University of Kentucky
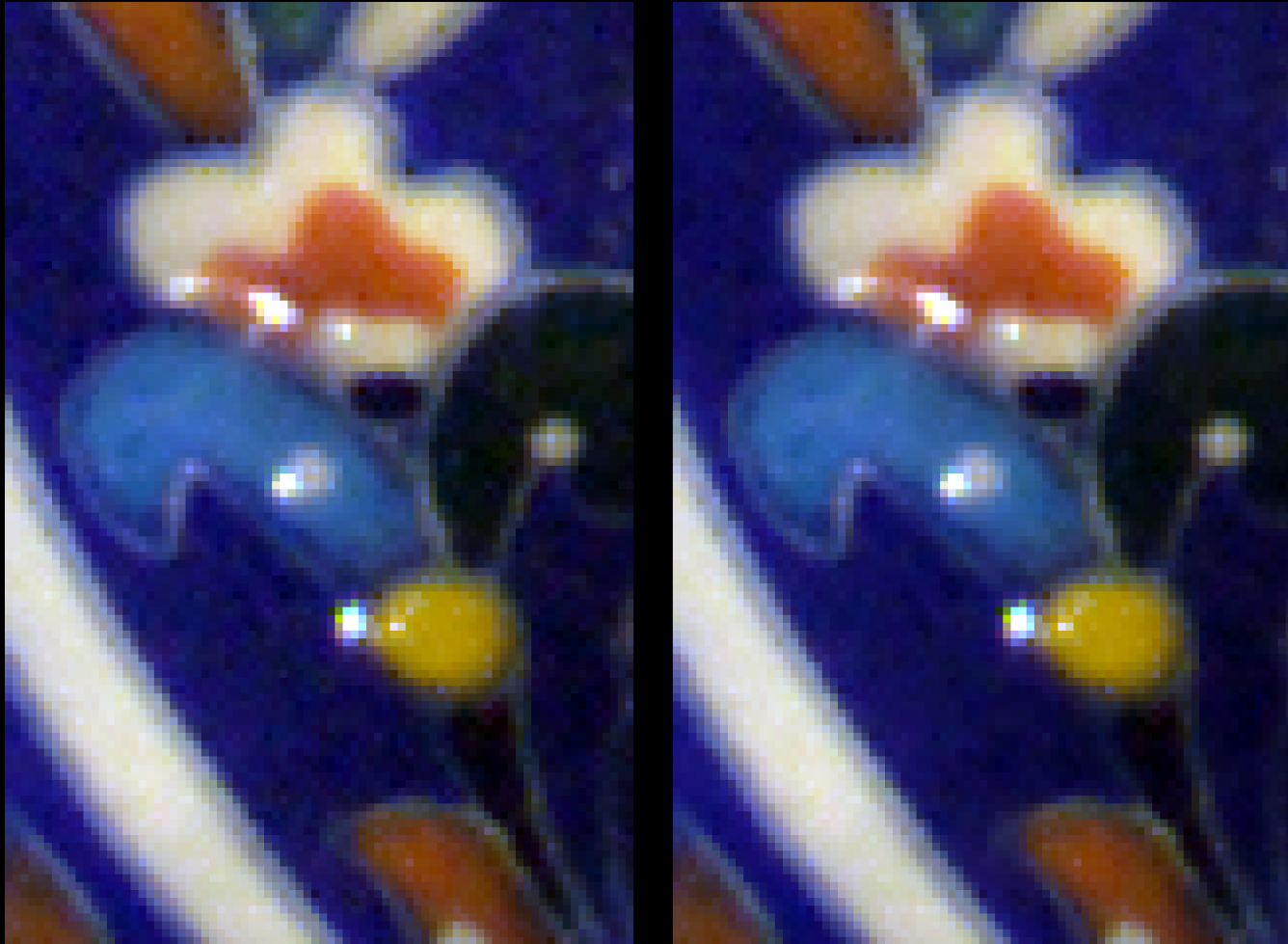
# Low ISO FF Bayer DSLR
## Canon 5D Mark IV @ ISO 250



- Quantify improvement by replacing LSBs with random noise, compare original vs. KREMY

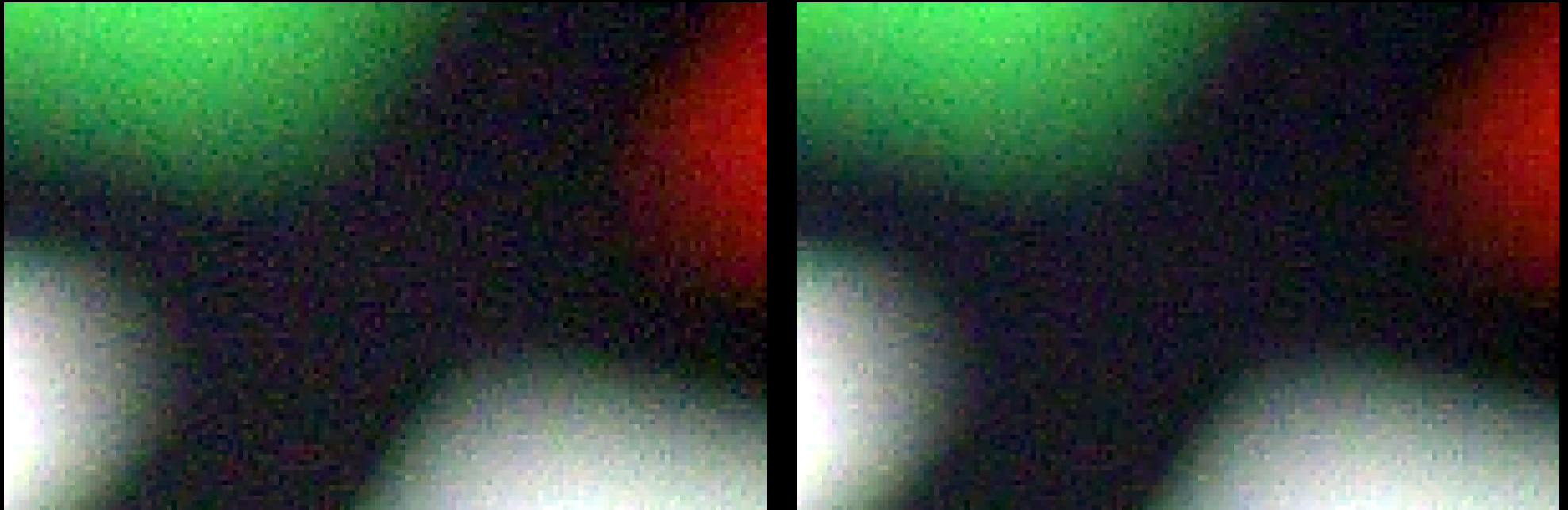University of Kentucky

# Low ISO FF Bayer DSLR
## Canon 5D Mark IV @ ISO 250
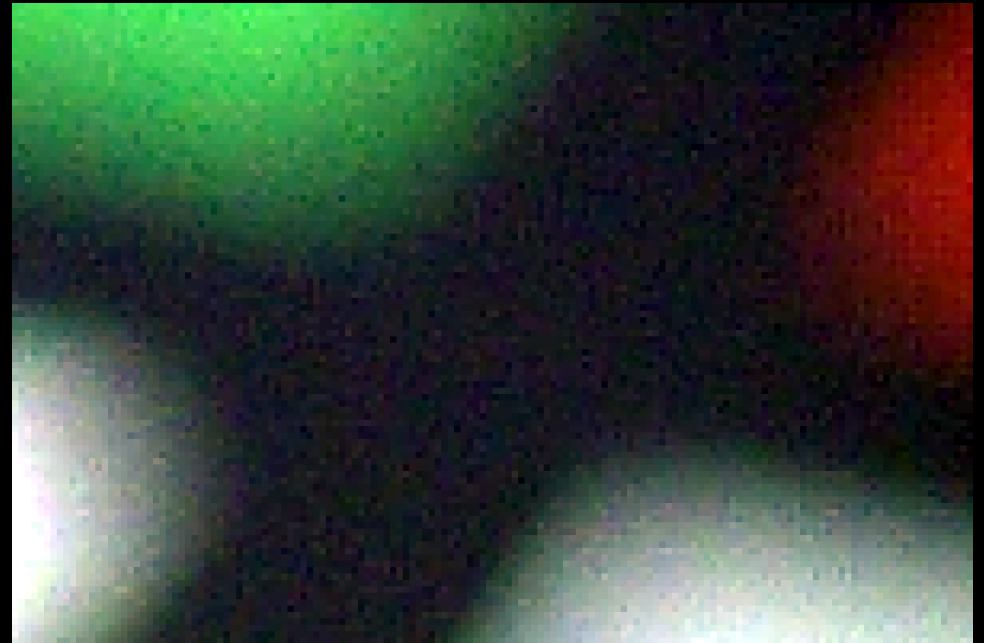


- All raw bits intact
- Left: original raw; Right: KREMY raw

# Low ISO FF Bayer DSLR
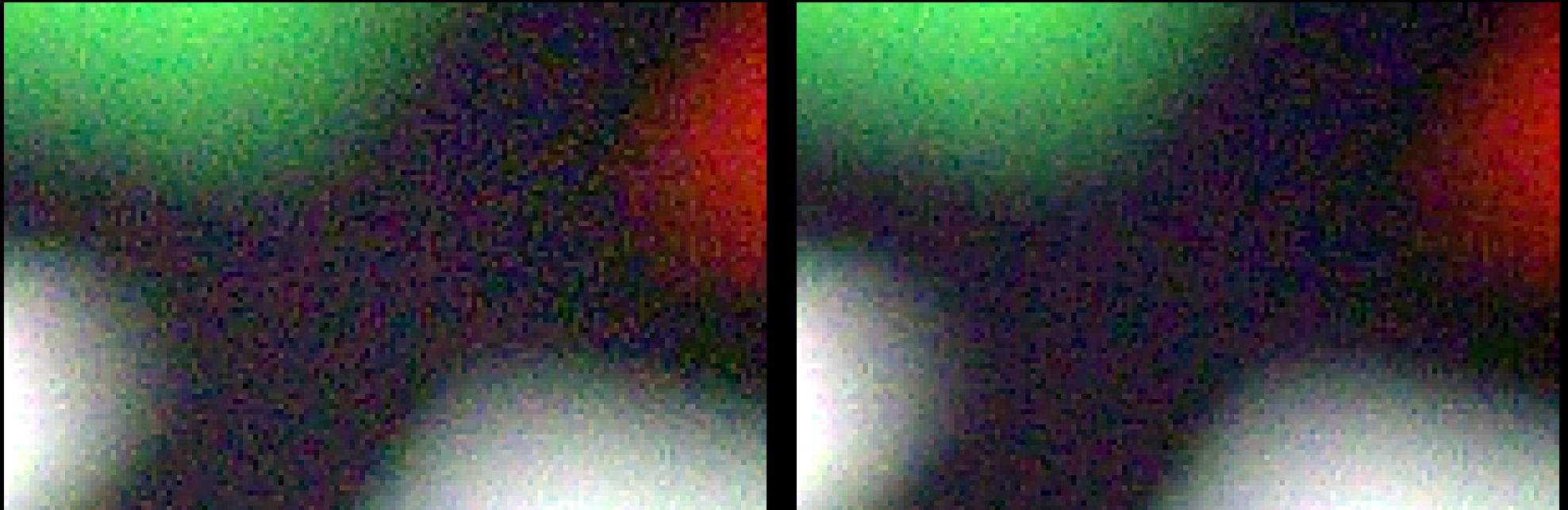## Canon 5D Mark IV @ ISO 250



- **4** LSBs replaced with random data
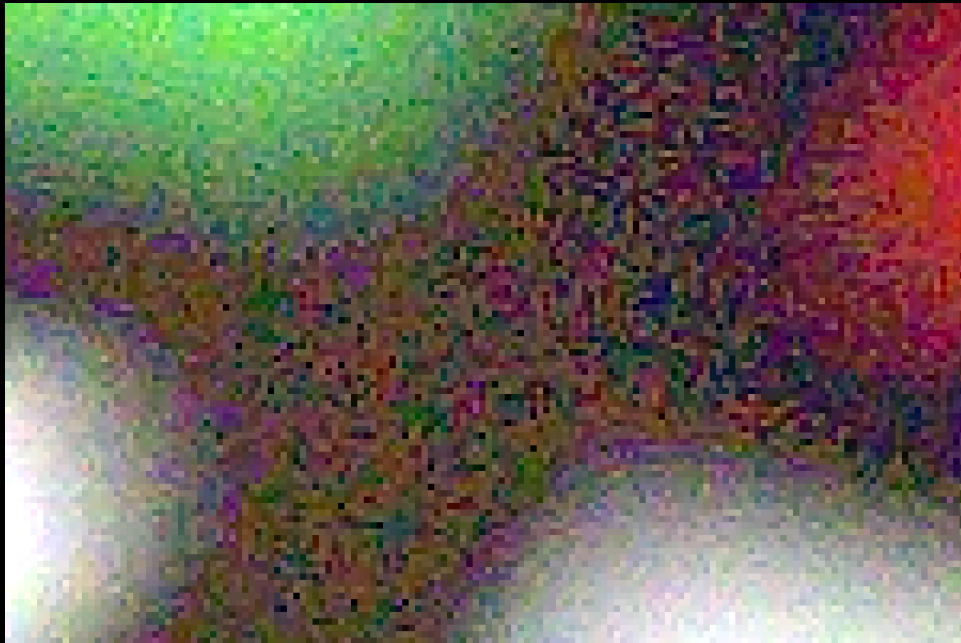- Left: original raw; Right: KREMY raw

University of Kentucky

# Low ISO FF Bayer DSLR
## Canon 5D Mark IV @ ISO 250



- **5** LSBs replaced with random data
- Left: original raw; Right: KREMY raw

University of Kentucky

# Low ISO FF Bayer DSLR
## Canon 5D Mark IV @ ISO 250



- **6** LSBs replaced with random data
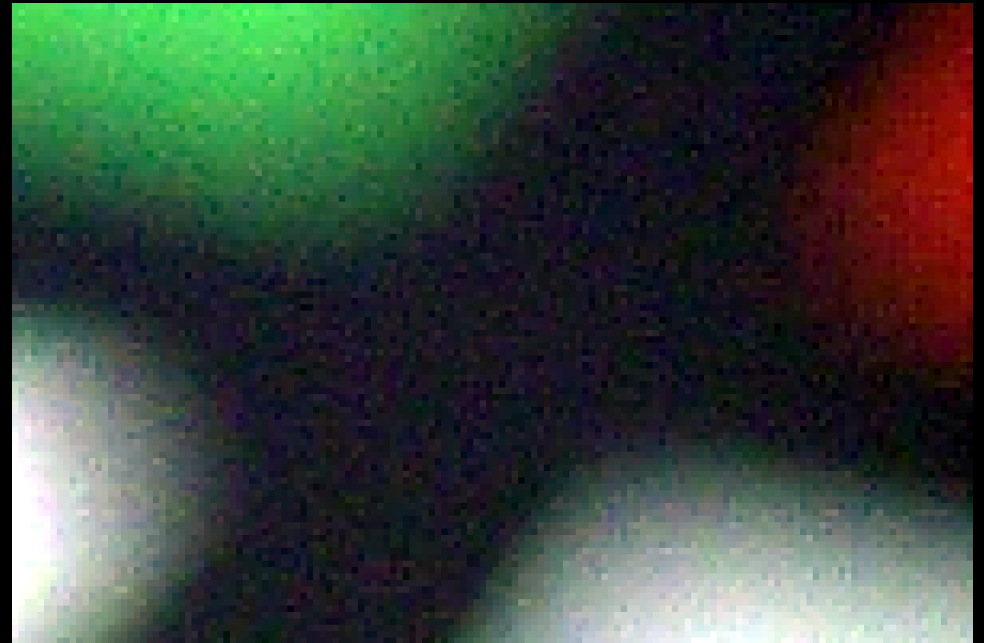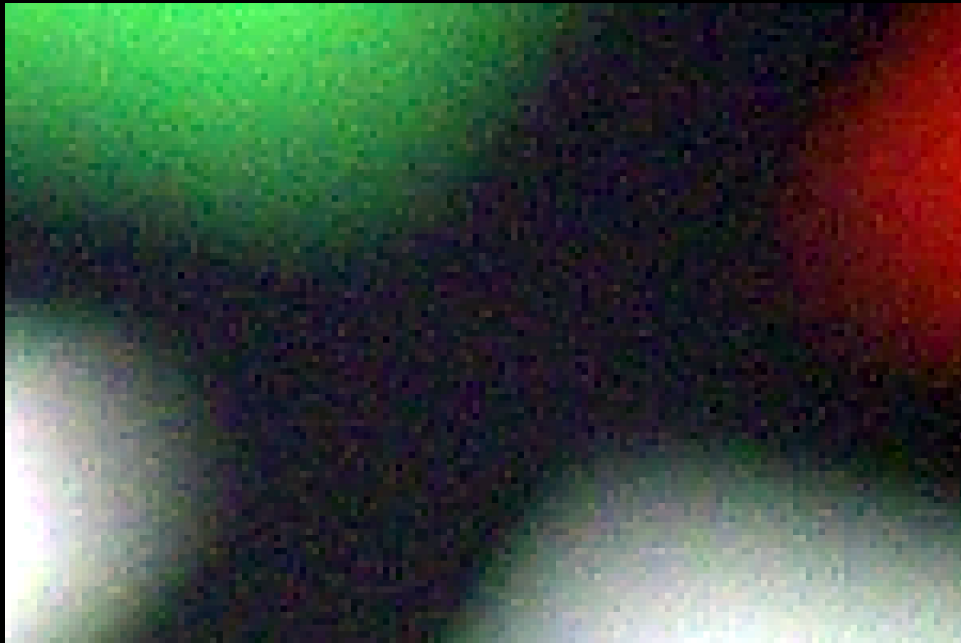- Left: original raw; Right: KREMY raw

University of Kentucky

# Low ISO FF Bayer DSLR
## Canon 5D Mark IV @ ISO 250



- Left: all raw bits intact
- Right: KREMY raw from **4** LSBs replaced

University of Kentucky

# Conclusions

- Empirical construction of value error model
- Texture synthesis within error bounds is a subtle improvement, but like ≥4 extra bits
- Works for all 2x2 CFA pattern DNGs
- Future work: speed-up algorithm, add controls