An improved *raw* image enhancement algorithm using a statistical model for pixel value error

Henry Dietz

COIMG-151, Noon, January 18, 2022

University of Kentucky
Electrical & Computer Engineering

University of Kentucky

Modeling Raw Pixel Value Error

- 2015: **KARWY** computed ***per-pixel error bounds*** based on details of Sony’s lossy compressed ARW raw format
- 2016: **KREMY** interpolated approximate error bounds from value ranges measured in ***areas automatically identified as evenly shaded***
- 2017: **TIK** computed error bounds by directly ***measuring the value range for pixels across captures*** of the same scene (not raw)

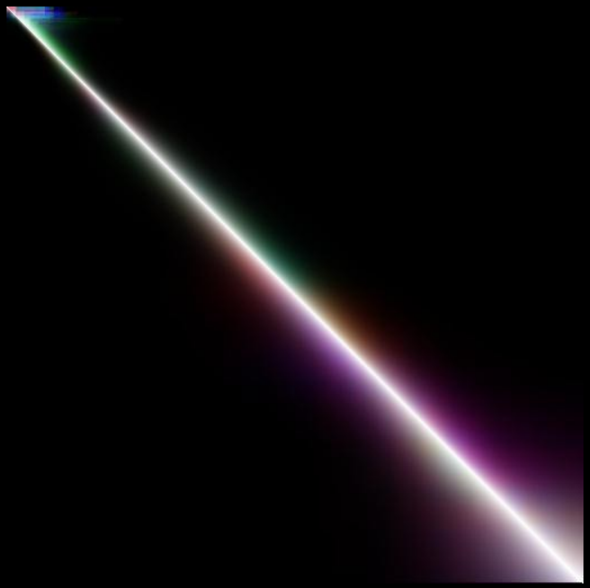
University of Kentucky

The New KREMY Model

- Like TIK, model is represented as an image
  - Easy visualization, manual editing
  - Not used as bounds
- Compute a ***probability density function*** per color channel for value read vs. ideal
  - Most similar neighbor represents same ideal pixel value? No, but close enough
  - If A and B might be the same ideal value, increment counts in square [A..B][A..B]
  - Normalize probabilities per ideal value

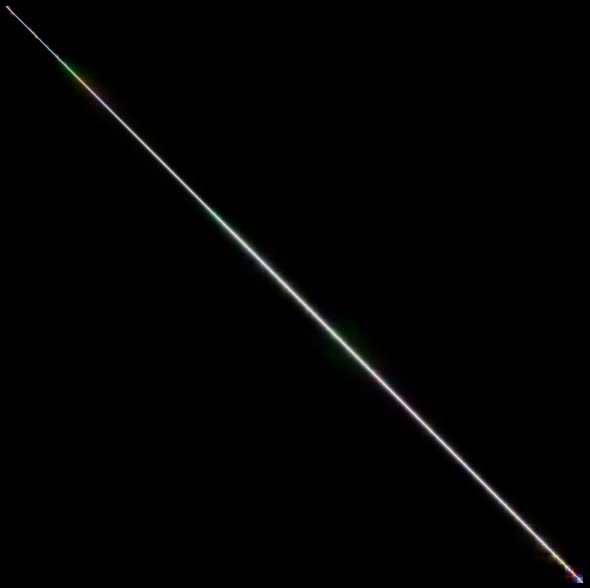
The New KREMY Model

- 2D image per channel:
  - Probability is 0..255
  - X is measured value
  - Y is ideal value
- Generally 4 channels (really 4 CFA patterns)
  - Four PGM images
  - One R,(G1,G2),B PPM image

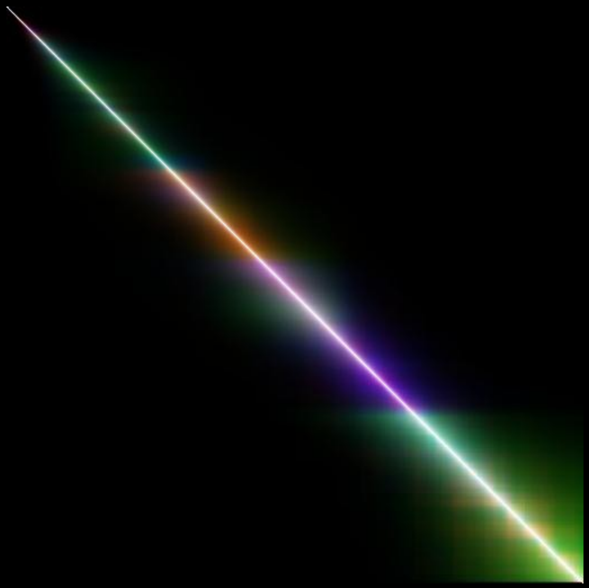


University of Kentucky

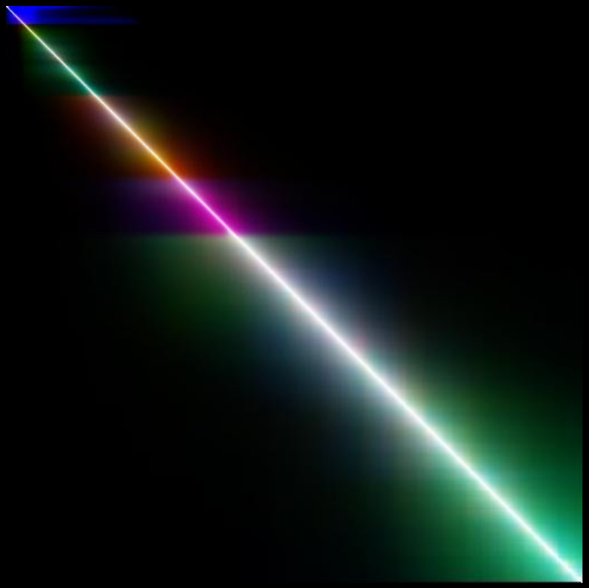
Sample New KREMY Models



Rebel XT ISO 100



Iphone 7 ISO 20



A7C ISO 204800

University of Kentucky

Credible Repair of raw data

- **Texture synthesis** is normally used for *inpainting credible values for missing pixels*
- KARWY and KREMY demonstrated use for ***adjusting*** *pixel values within bounds*
- Here, ***weight textures probabilistically***
  - **Every pixel value is replaced**
  - **Some pixel values can change a lot...**

Texture Synthesis (Simplified)

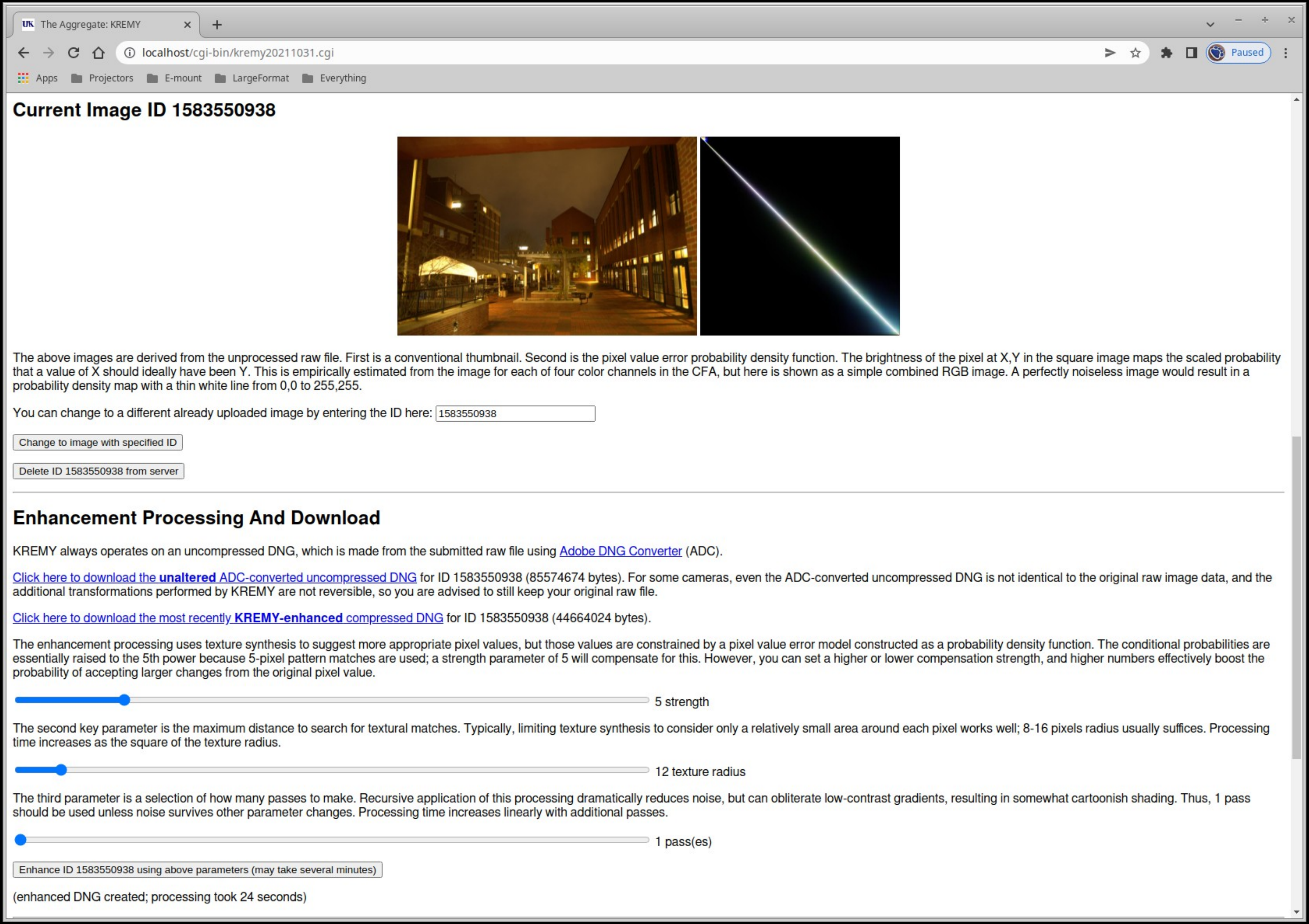
```
for (p is each pixel) {
 v = 0; w = 0;
 for (q is each same CFA context pixel) {
  wprob = patch similarity weighted
   probability p,q have same ideal value;
  v += (*q * wprob); w += wprob;
 }
 *p = (v / w);
}
```

- Probabilities from $\texttt{model}_{\texttt{color(n)}}\texttt{[*q}_{\texttt{n}}\texttt{][*p}_{\texttt{n}}\texttt{]}$
- Weights reduce as distance increases

Implementation of new KREMY

- Use tool (e.g., Adobe DNG Converter) to convert raw to an uncompressed DNG
- Packages improved raw data as a DNG
  - Algorithm is only ~1200 lines C code
  - Uses `raw2dng` or `dcraw` to edit DNG data, but it could be built-into a raw processor
  - Sequential execution takes tens of seconds

University of Kentucky



Current Image ID 1583550938

The above images are derived from the unprocessed raw file. First is a conventional thumbnail. Second is the pixel value error probability density function. The brightness of the pixel at X,Y in the square image maps the scaled probability that a value of X should ideally have been Y. This is empirically estimated from the image for each of four color channels in the CFA, but here is shown as a simple combined RGB image. A perfectly noiseless image would result in a probability density map with a thin white line from 0,0 to 255,255.

You can change to a different already uploaded image by entering the ID here: 1583550938

Change to image with specified ID
Delete ID 1583550938 from server

Enhancement Processing And Download

KREMY always operates on an uncompressed DNG, which is made from the submitted raw file using Adobe DNG Converter (ADC).

Click here to download the **unaltered** ADC-converted uncompressed DNG for ID 1583550938 (85574674 bytes). For some cameras, even the ADC-converted uncompressed DNG is not identical to the original raw image data, and the additional transformations performed by KREMY are not reversible, so you are advised to still keep your original raw file.

Click here to download the most recently **KREMY-enhanced** compressed DNG for ID 1583550938 (44664024 bytes).

The enhancement processing uses texture synthesis to suggest more appropriate pixel values, but those values are constrained by a pixel value error model constructed as a probability density function. The conditional probabilities are essentially raised to the 5th power because 5-pixel pattern matches are used; a strength parameter of 5 will compensate for this. However, you can set a higher or lower compensation strength, and higher numbers effectively boost the probability of accepting larger changes from the original pixel value.

5 strength

The second key parameter is the maximum distance to search for textural matches. Typically, limiting texture synthesis to consider only a relatively small area around each pixel works well; 8-16 pixels radius usually suffices. Processing time increases as the square of the texture radius.

12 texture radius

The third parameter is a selection of how many passes to make. Recursive application of this processing dramatically reduces noise, but can obliterate low-contrast gradients, resulting in somewhat cartoonish shading. Thus, 1 pass should be used unless noise survives other parameter changes. Processing time increases linearly with additional passes.

1 pass(es)

Enhance ID 1583550938 using above parameters (may take several minutes)

(enhanced DNG created; processing took 24 seconds)

Testing the new KREMY

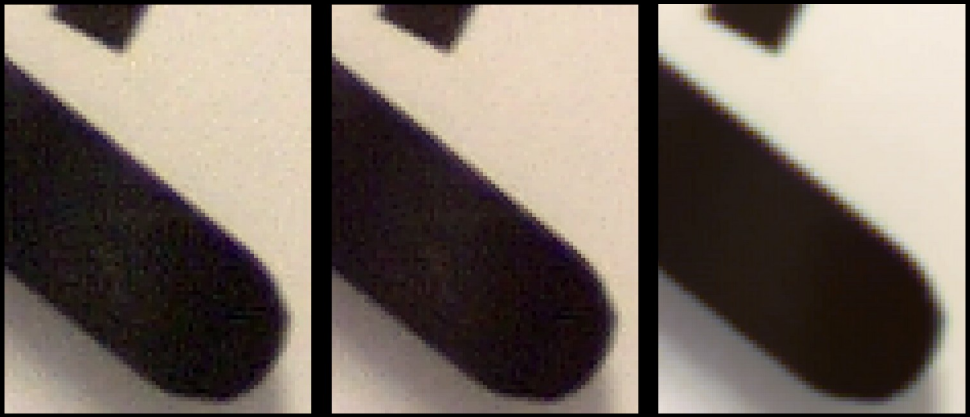
- There is a WWW-interfaced version:

  `http://aggregate.org/DIT/KREMY`

  - Not much user feedback, but positive
  - Some loss of low-contrast details
- Raw processing of images in this paper
  - Used `dcraw` to avoid enhancement
  - Crops scaled 8X without interpolation

University of Kentucky

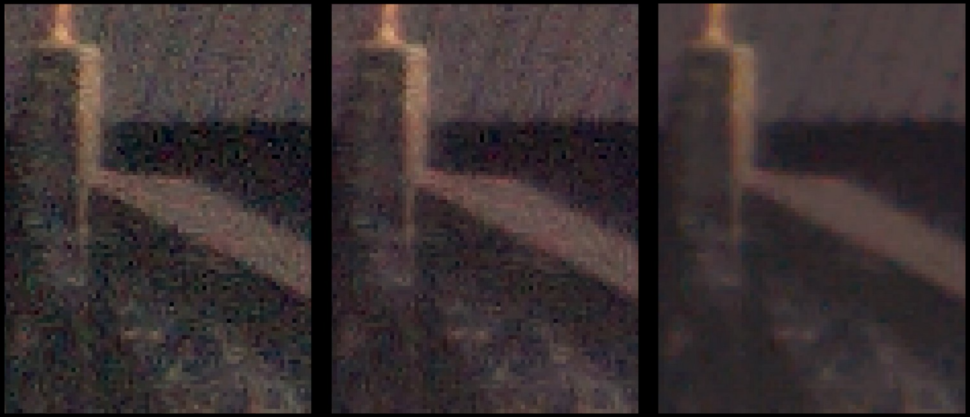
raw | KREMY | new KREMY



Canon Digital Rebel XT

🟥🟩
🟩🟦 ISO 100

new(3, 12)



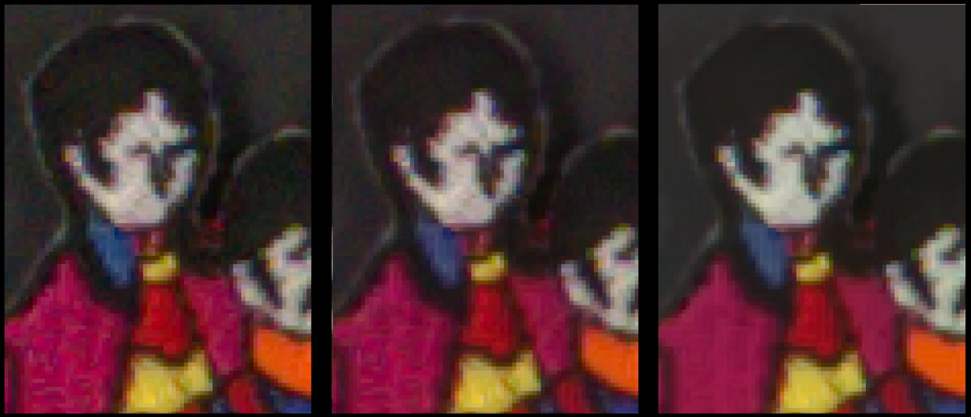
Canon PowerShot S70

🟥🟩
🟩🟦 ISO 50

new(3, 12)

University of Kentucky

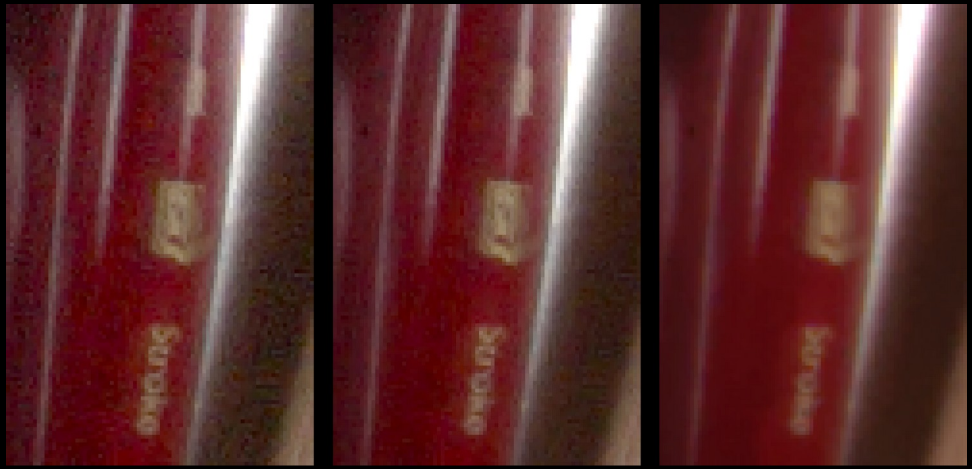
raw | KREMY | new KREMY



Olympus E-M1 Mark II

🟥🟩
🟩🟦 ISO 400

new(3, 12)



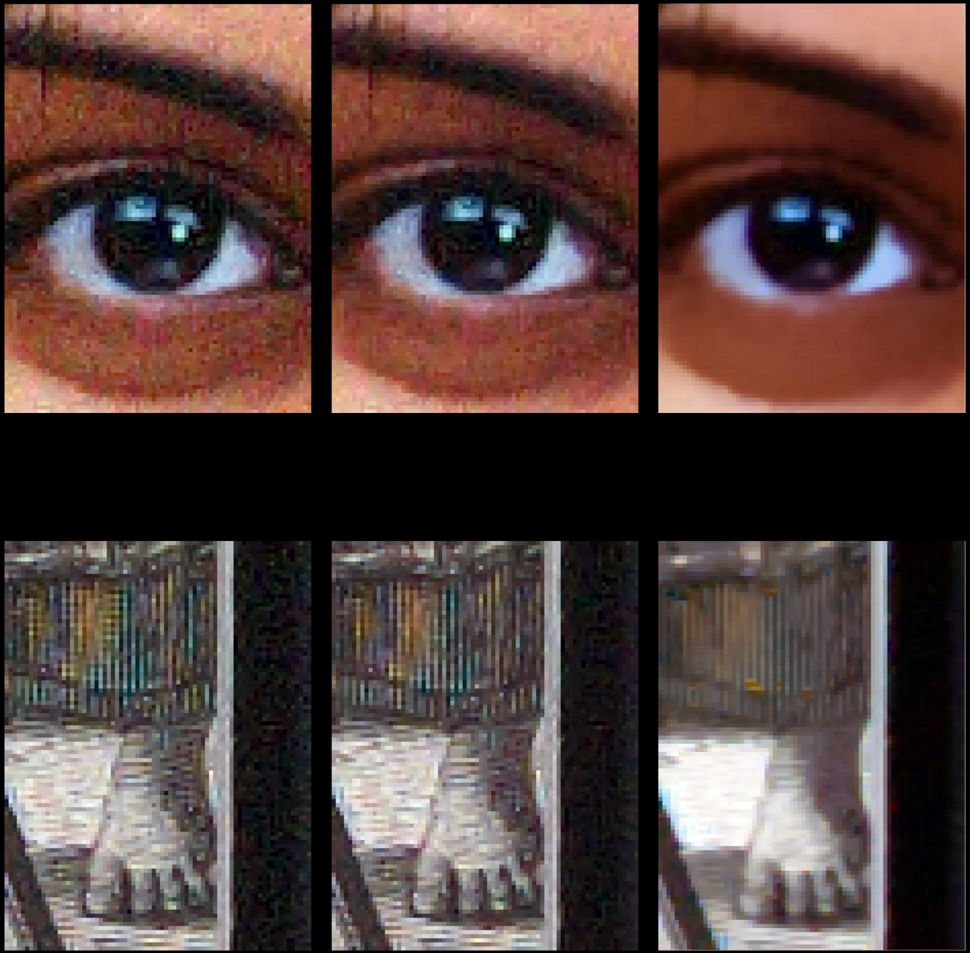
Apple iPhone 7

🟥🟩
🟩🟦 ISO 20

new(3, 12)

University of Kentucky

raw | KREMY | new KREMY



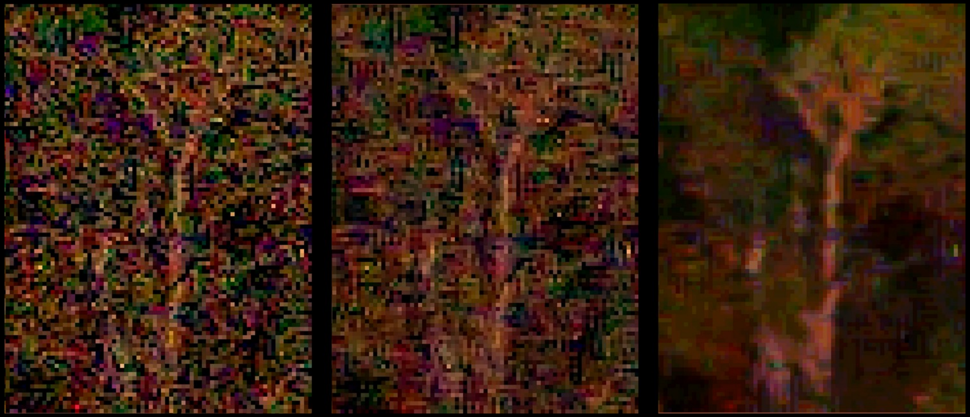
Nikon D810

🟥🟩
🟩🟦 ISO 1600

new(3, 12)

University of Kentucky

raw | KREMY | new KREMY

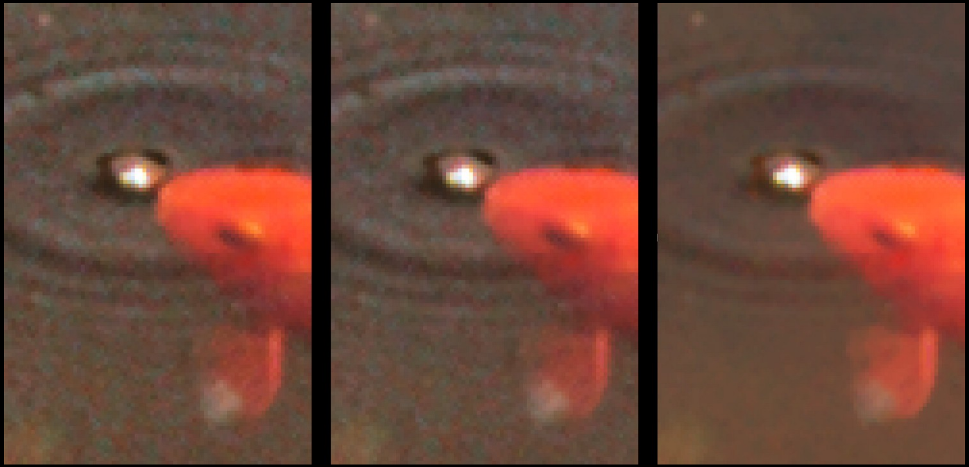


Sony NEX-7

🟥🟩
🟩🟦 ISO 1600

new(5, 12)

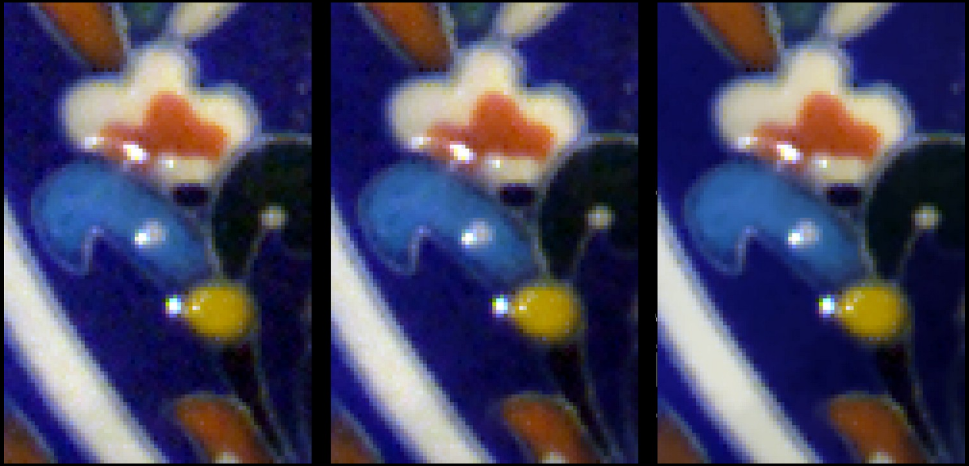
raw | KREMY | new KREMY



Canon PowerShot G1

🟩🟪
🟨🟦 ISO 100

new(3, 12)



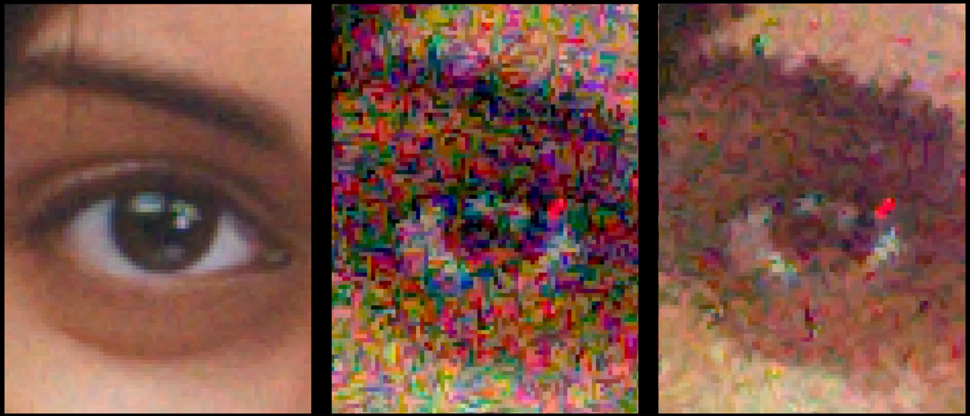
Sony DSC F828

🟥🟦
🟩🟦 ISO 64

new(3, 12)

University of Kentucky

raw lo | raw hi | new KREMY



raw @ ISO 1600

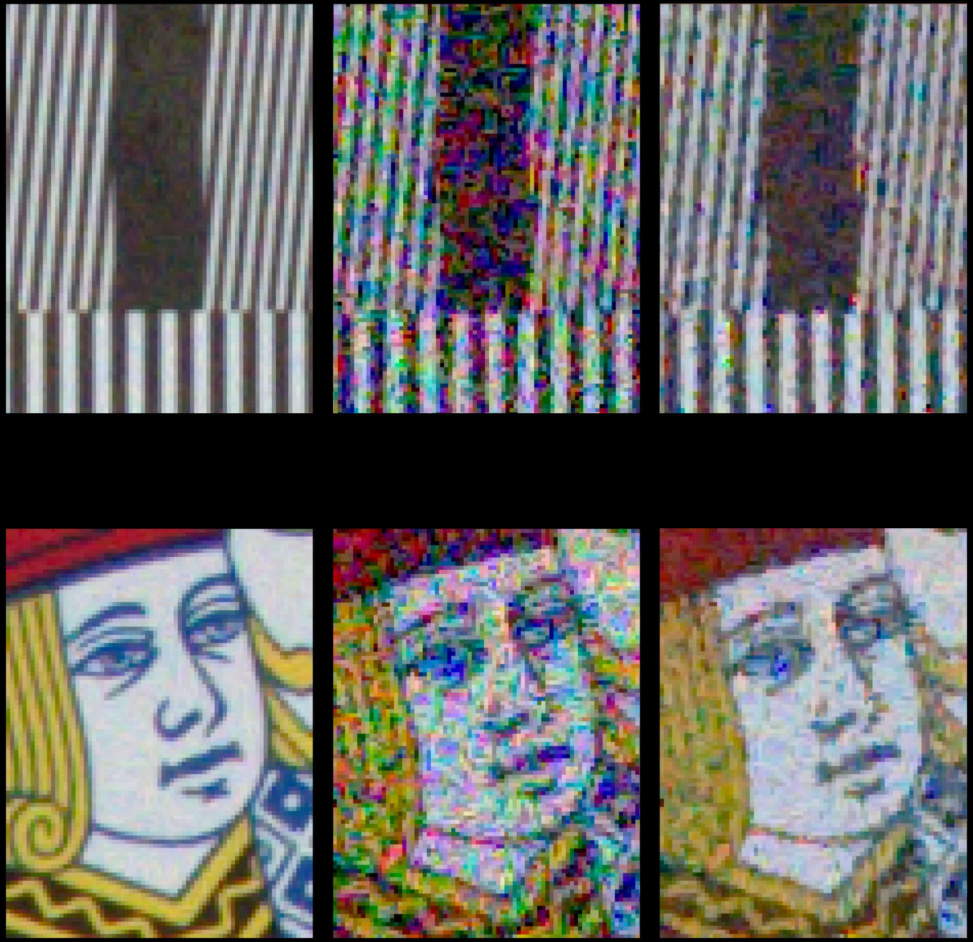
raw @ ISO 204800

Sony A7C

🟥🟩
🟩🟦 ISO 204800

new(5, 24)

raw lo | raw hi | new KREMY



Canon R5

🟥🟩
🟩🟦 ISO 102400

new(5, 48)

Conclusions

- Probability-based pixel value model works
  - Can compute from a single capture
  - Models combined effect of all noise
  - **Probability density image** is efficient to use, editable, and visually meaningful
  - Probabilities model more extreme noise
- Texture synthesis in new KREMY
  - Simple, untrained, deterministic, algorithm
  - Very effective ***credibly improving raws***