The Error Diffusion Halftoning Algorithm: Some Stability Results and Applications Beyond Halftoning

Chai Wah Wu, IBM Research Division

April 11-15, 2005

Collaborators

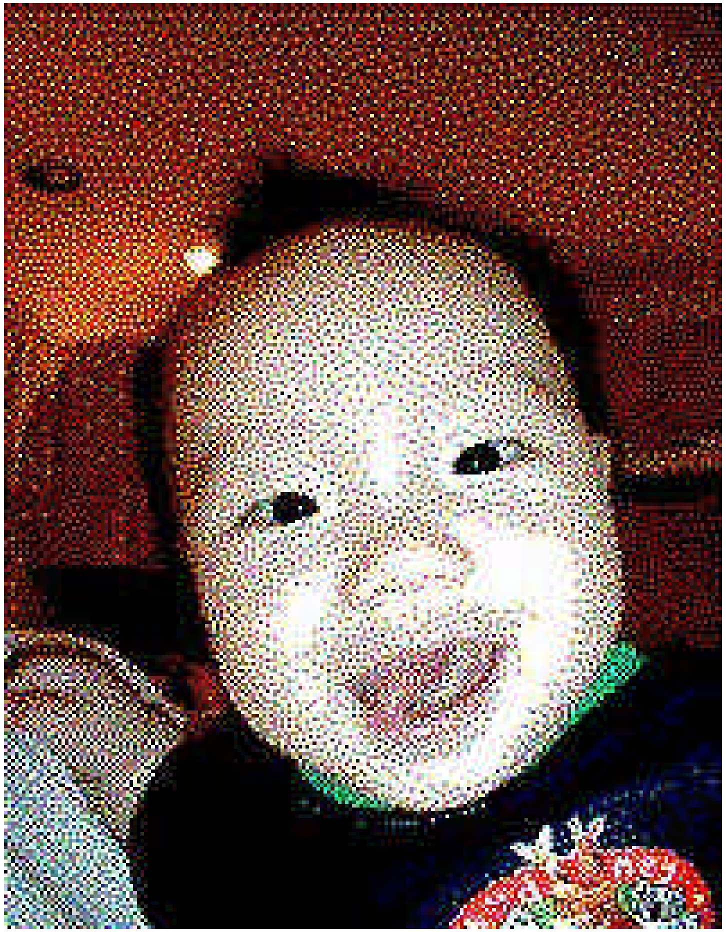
- Roy Adler
- Don Coppersmith
- Bruce Kitchens
- Marco Martens
- Arnaldo Nogueira
- Tomasz Nowicki
- Giuseppe Paleologo
- Mike Shub
- Mikel Stanich
- Gerry Thompson
- Charles Tresser
- Steve Wright

Digital halftoning

- The art of using a few (output) colors to produce a picture such that when viewed at the right distance appears to have many (input) colors.
- Necessary in printers (both analog and digital) and sometimes displays as they use only a handful of colors.
- Possible because of
  1. Trichromacy of the human visual system (mixing 3 colors is enough to get any color).
  2. Blurring behavior of the human visual system (Low-pass filter)
- Trichromacy is due to the human eye having three types of color receptors (cones).

Original

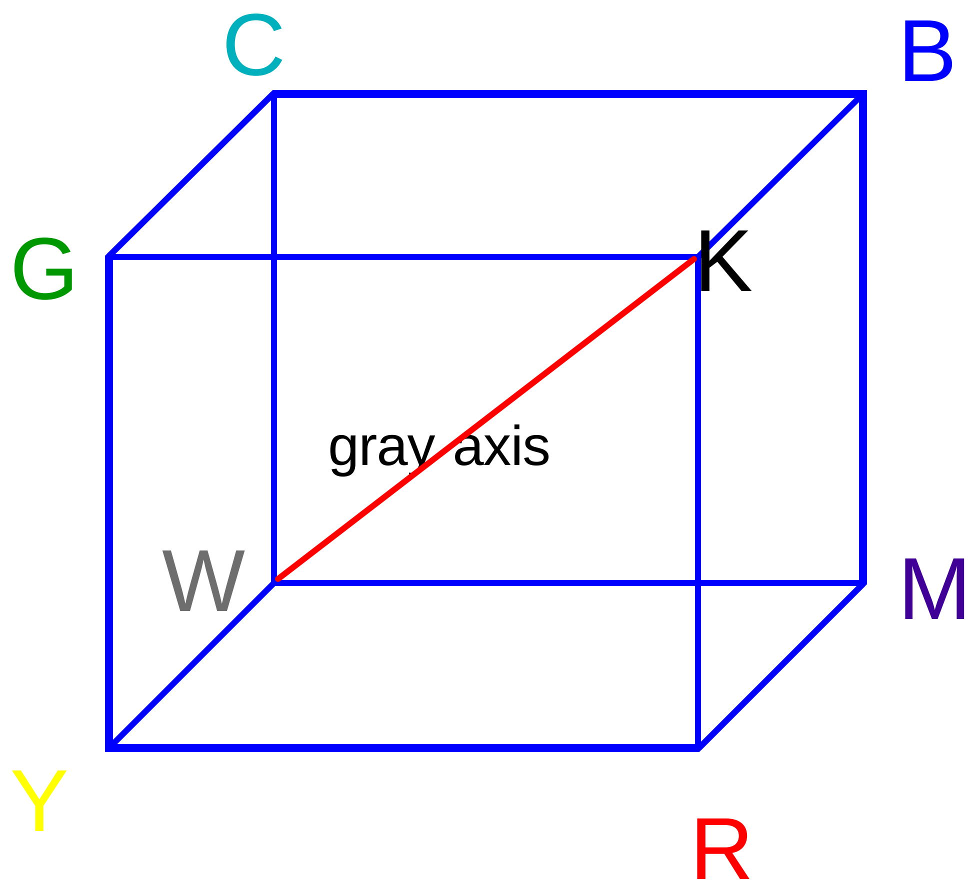
Halftoned



Digital halftoning problem

- Given an input image *I*, generate an output image *O* using a restricted set of output colors such that *O* is *close* to *I*.
- Input image *I* consists of elements from the set of input colors (or color gamut) S,
- Output image *O* consists of elements from the set of output colors V.
- S and V lie in some color space, such as RGB, CIEXYZ, or CIELab.
- Grayscale images: one-dimensional color space. Represented by unit interval: 0 is white, 1 is black with gray in between.
- We are interested in the convex hull of V (I will explain why later).

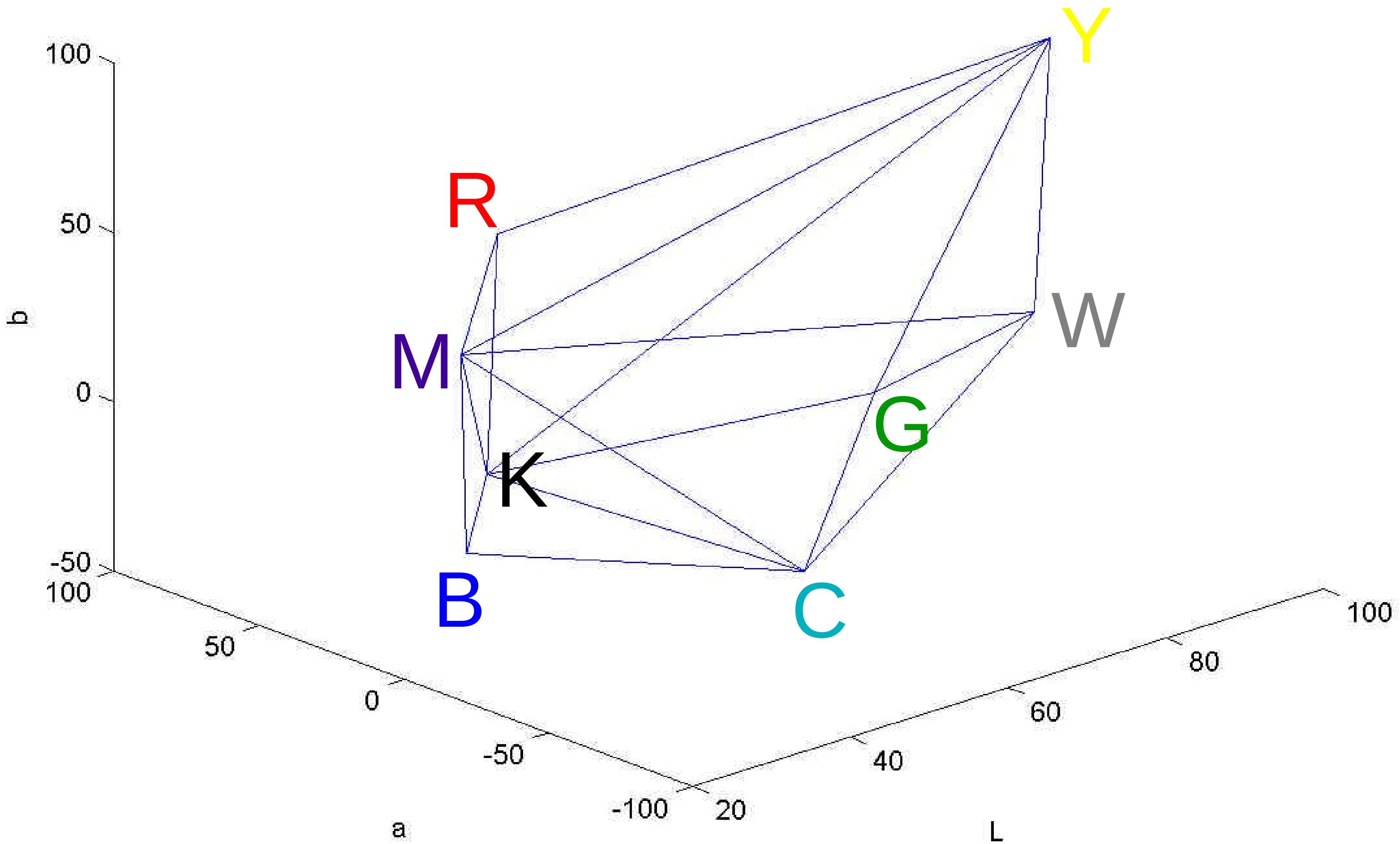
3-dimensional color space
(CMY space)



CIELab space

Convex hull is a polygon with 12 faces

(data from Kouzaki et al, Proc. SPIE, vol. 3648, pp. 470-479, 1999)



Error diffusion

- A popular technique for high quality digital halftoning is error diffusion. Main idea: the error induced by choosing an output color is distributed to neighboring pixels. The output color is chosen to be the closest one to the modified input = current input + errors from neighbors.

An error diffusion algorithm

is defined by the following steps:

1. Choose an enumeration of the pixels.
2. At each pixel location, add to the input I(i) a weighted average of the previous errors in some neighborhood to obtain the modified input M(i).
3. Choose O(i) an element of V closest to M(i).
4. Define the error e(i) as M(i)-O(i).

A generalized error diffusion algorithm

is defined by the following steps:

1. Choose an enumeration of the pixels.
2. At each pixel location, add to the input I(i) a weighted average of the previous errors in some neighborhood to obtain the modified input M(i).
3. Choose O(i) an element of V ~~closest to M(i)~~.
4. Define the error e(i) as M(i)-O(i).

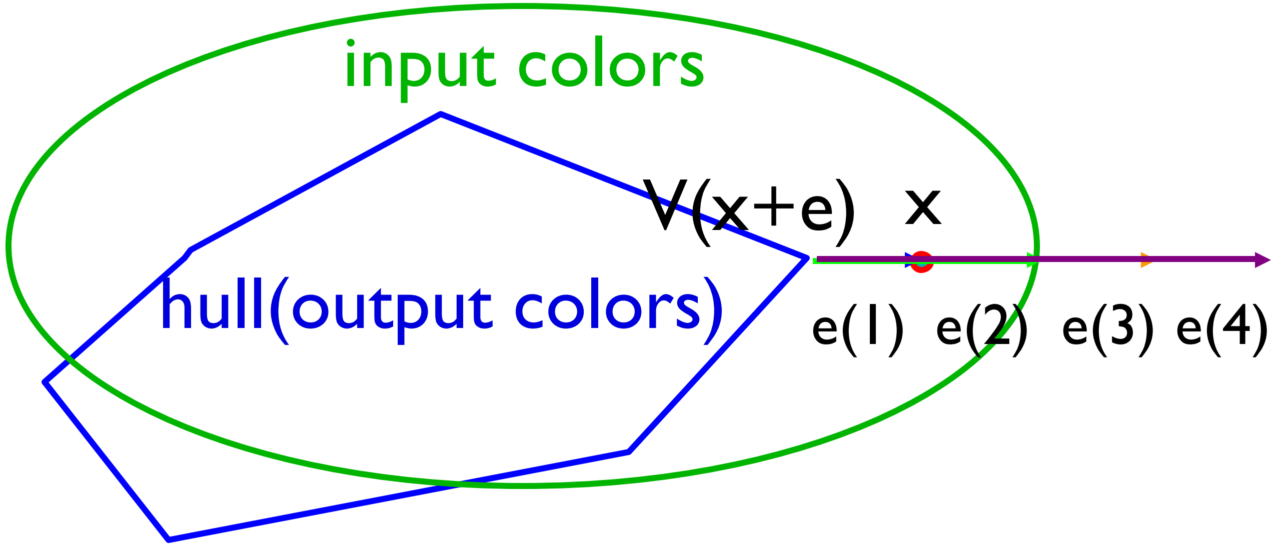
Algorithm is not necessarily deterministic.

One-step error diffusion algorithm

- In one spatial dimension, the simplest case is: send the error to the next pixel.
- $M(i) = I(i) + e(i-1)$, $e(i) = M(i) - v(M(i))$, $v(M(i)) = O(i)$.
- In this case, $e(i) = \sum_{m \leq i} I(m)-O(m)$.

Are the errors e(i) bounded ?

- If the set of input colors is not in the convex hull of the set of output colors, then there is a sequence of inputs which make the errors diverge.
- Let I(i) = x for all I



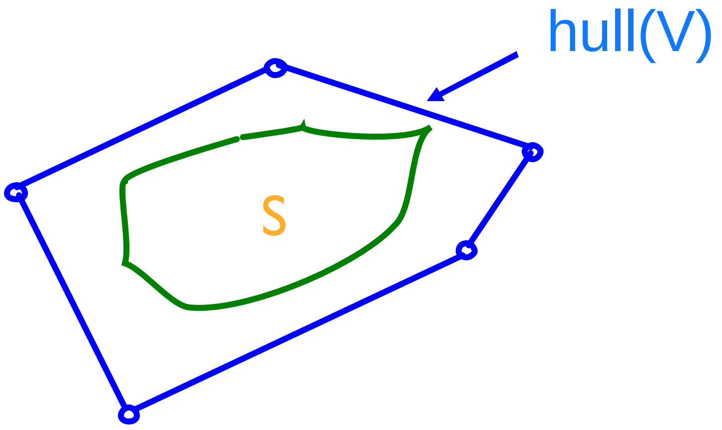
- Also true for generalized error diffusion.

What about the converse?

- Assume input color is in the convex hull of output colors.
- Consider the following algorithm:
  1. Map color space into unit simplex by sending vectors in V to unit vectors.
  2. Perform error diffusion on the unit simplex.
  3. Project back onto color space.
- Map in 1. is generally not unique.
- This algorithm has convex bounded invariant regions for the error, independent of the input. This implies Bounded-Input-Bounded-State (BIBS) stability.

Conditions for bounded error

- S is contained in the convex hull of V if and only if there exists a generalized error diffusion algorithm such that all bounded images generate bounded errors.



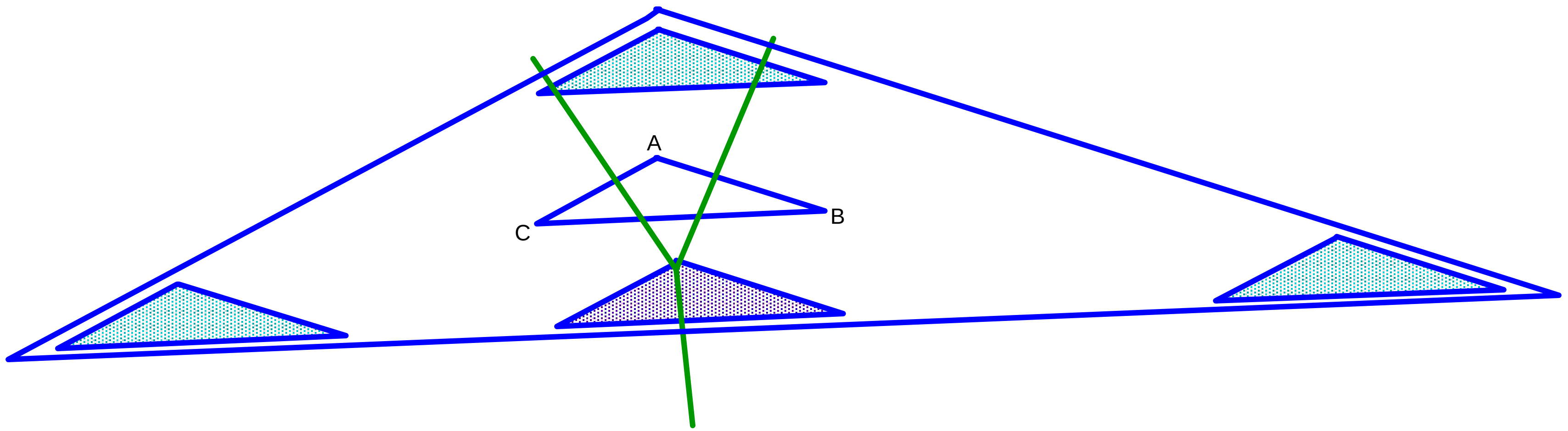
What about (classical) error diffusion?

- Bounded errors also holds for classical error diffusion.
  - Easy to prove for 2-D color spaces.
  - Much harder to prove in general.
- Rewrite equation as nonautonomous discrete-time dynamical system:

  $X_i = x_{i-1} + I_i - v(x_{i-1}) = f_{I_i}(x_{i-1})$

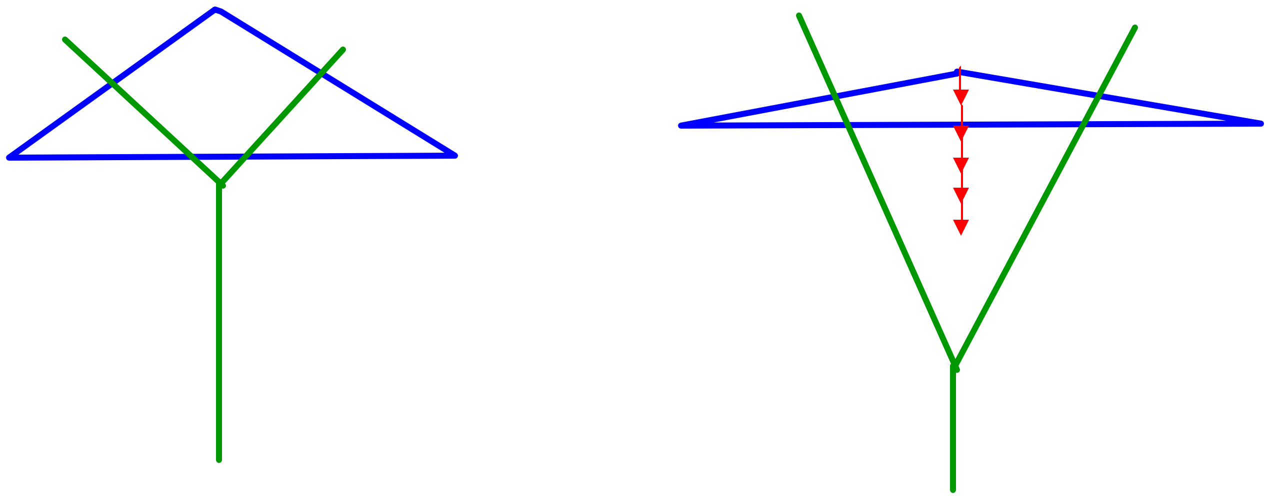
  where the state x_i is the modified input.
- Show existence of convex bounded invariant set for modified input, regardless of I_i.
- Bounded modified input implies bounded error.

Convex invariant region in dimension 2



- To prove theorem in color space of dimension 2, flow the sides at same speed for some time large enough
- Robust algorithm: small uncertainty in Voronoi boundary will result in (almost) the same invariant region.

- Error can be large for small input gamut.



- This can lead to color smear and other artifacts.
- The generalized error diffusion algorithm obtained by lifting polygon to unit simplex has bounds which depends on number of vertices of polygon. This lessens the problem of large errors.
- There exists an algorithm whose bound depends on dimension of polygon.

Weighted averages

- Although these results are given for one-step error diffusion, it can be shown that bounded modified input in one-step ED implies bounded modified input in ED with general weighted averages of error.
- Bounded error in simplex-based generalized ED also true for weighted averages of error.

When is the output image O close to the input image I?

- Closeness criterion should be related to the human visual system (HVS) and be amenable to mathematical analysis.
- Use linear shift-invariant low-pass filter as model of HVS.
- Goal is to make low-pass filtered version of image close to low-pass filtered version of halftoned image.

Error diffusion and HVS

- HVS-based error metric *d(O,I)* between images: norm of difference after HVS lowpass filtering.

$$d(O,I) = \left\| L(O) - L(I) \right\|$$

- What is the relationship between error diffusion and this error metric?
  - Bounded error in (generalized) ED implies boundedness of *d(O,I)*.
  - The more low-pass the filter is, the smaller *d(O,I)* is. This validates the operation of vector error diffusion.

Multiple polytopes

- Consider the case where at each time the set of output vectors V is chosen from $V_1, \ldots, V_m$.
- At each time the input is in the convex hull of V.
- If S_i is an invariant region for the case with a single polytope V_i, then $\sum_k S_k$ is an invariant region for this problem.
- Another bound

$$\|e(t)\| \leq \binom{n-1}{2} \sup_{\sum b_j = 0, |b_j| \leq 1} \left\| \sum b_j v_j \right\|$$

where $\{v_j\} = \cup_i V_i$.

Some applications beyond halftoning

- Problems in scheduling
- Image watermarking
  - Data hiding/steganography
  - Self-repairing images
- Images under different viewing conditions

Online scheduling of variable task requests

- n tasks compete for resources where at each time interval 1 task is scheduled on the resources and each task requests a specific proportion of the resource capacity.
  - Online: schedule is made as the requests are received.
  - Variable task requests: changes at each time interval.
  - Lag for task *i*: difference between accumulated requests and the number of times the task is scheduled.
  - Sup-norm lag bound $\max_i$ |lag of task i|.
  - Full utilization of resource: sum of task requests = 1.
- Also known as the Chairman Assignment Problem.
- Given an algorithm A, define B_A as maximum lag bound over all possible task requests and $B = \inf_A B_A$, where A ranges over all on-line algorithms.

Chairman assignment problem

- Think of the tasks as output colors and the task request as an input in the convex hull.
- Error diffusion on unit simplex give a on-line scheduling algorithm that generates a schedule with a bound

$$B_A = \sum_{j=2}^{n} \frac{1}{j} = H(n) - 1 < \ln(n)$$

where n is the total number of tasks.

- As far as we know, only known bound for on-line algorithms.
- Furthermore, error diffusion is optimal among all on-line algorithms, i.e. $B = B_A$.

Generalized Carpool Problem

- Same as the Chairman Assignment Problem, except that the chosen task at each time must be among tasks whose task request is nonzero.
- Modify error diffusion to pick the output only from among those tasks.

$$B_A = \frac{n-1}{2}$$

- For this problem, error diffusion is also optimal among all on-line algorithms, i.e. $B_A = B$.

Carpool Problem

- Same as the Carpool problem, except that at each time t, the task requests are either 0 or 1/b for some integer b (which changes at each time)
- For this problem, we can show:

$$\frac{3(n-1)}{8}-\frac{1}{4}+\frac{1}{4n}<B\leq\frac{n-1}{2}-\frac{1}{lcm(2,3,\ldots,n)}$$

where the upper bound is due to Error Diffusion.

Chairman assignment problem with multiple tasks

- Same as the Chairman Assignment Problem, except that now m tasks are chosen and at each time, the same task cannot be chosen more than d times. Each task request $\leq$ d, and sum of task requests = m.
- Modify error diffusion to pick $\lfloor m/d \rfloor$ tasks with largest modified input d times and the next largest task m mod d times.
- If d|m,

$$H\left(\left\lceil \frac{n}{m} \right\rceil\right) - 1 < B \leq \max\left(\frac{m}{2d} + \frac{1}{2} - \frac{m}{dn}, \frac{n}{2} + \frac{m}{dn} - \frac{m}{2d} - \frac{1}{2}\right)$$

where the upper bound is due to Error Diffusion.

Pixelwise transformation on sets of images

- A general framework of operations on sets of images which has several applications. Inputs are n images $A_1, \ldots, A_n$, outputs are m images $B_1, \ldots, B_m$.
- Consider an image transformation Φ which acts on an n-tuple of pixels and produce an m-tuple of pixels: $\Phi(p_1,\cdots,p_n) = (q_1,\cdots, q_m)$.
- Applying this pixelwise to a set of n images generates a set of m images.
- 2 interpretations:
  1. The m images are watermark images generated by Φ.
  2. The n+m images correspond to how a single image appears under different viewing conditions.
- Goal is to match these n+m images with some predefined n+m images.
- Perfect matching is generally impossible since there are more sets of n+m images than there are sets of n images.
- But if we are looking at images from a distance, then only need to match lowpass filtered images and halftoning provides a solution to the problem.

Pixelwise transformation on 2 images.

- Consider the example: $n = 2$, $m = 1$. Operation acts on 2 images A_1 and A_2 producing a third image B.
- The pixelwise operation is $B(i,j) = A_1(i,j)$ **o** $A_2(i,j)$
- **o** can be any operation with two operands. If we are overlaying images, **o** is the OR operation.
- **o** can be asymmetric, i.e. a **o** b $\neq$ b **o** a.

Halftoning in Cartesian product of color spaces.

- Input to the digital halftoning algorithm is the composite image $A'_1 \times A'_2 \times B'$.
- The set of output vectors are (0,0,0 **o** 0), (1,0,1 **o** 0), (0,1,0 **o** 1), (1,1,1 **o** 1).
- Apply halftoning algorithms such as error diffusion to construct output image $A_1 \times A_2 \times B$ which minimizes the error defined as:

$$E = v_1||L(A_1-A'_1)|| + v_2||L(A_2-A'_2)|| + v_3||L(B-B')||$$

where L is the lowpass operator model of the human visual system (HVS).

Gamut mapping

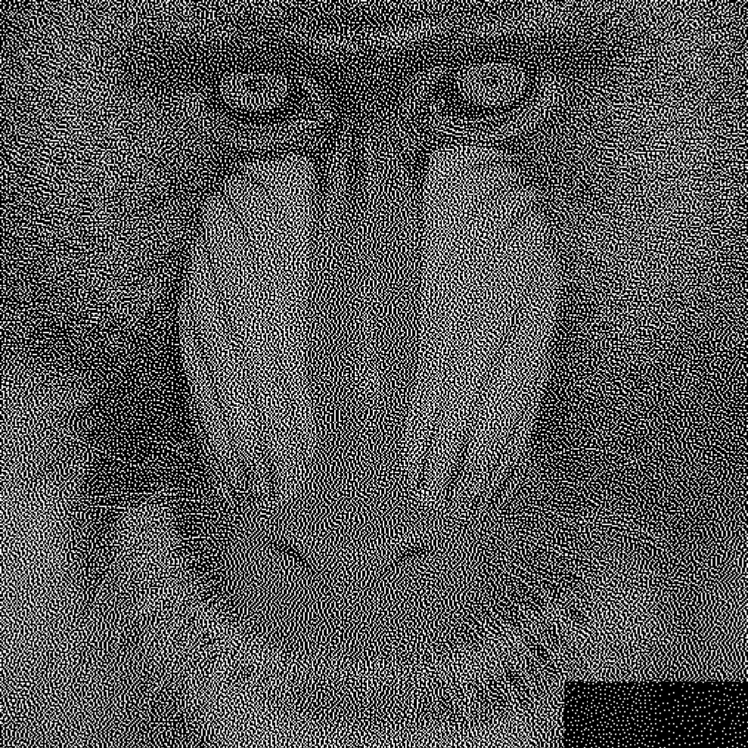
- It was shown before that for the error to remain bounded, pixels of the input image should lie inside the convex hull of the output vectors.
- Gamut mapping is needed to ensure this is the case.
- Assume affine gamut mapping and find optimal gamut mapping using nonlinear programming.
- Obtain gamut mapping for a specific set of images or for a range of images.
- Convex hull condition can be violated mildly to reduce severity of gamut mapping.

Priority of images

- The fidelity of these n+m images can be adjusted depending on the application.
- This adjustment can be done both in the error calculation of the digital halftoning algorithm and in gamut mapping computation.
- For example, since the fidelity of the watermark image is usually less important that the watermarked images, the extracted watermark image can take on more distortion.

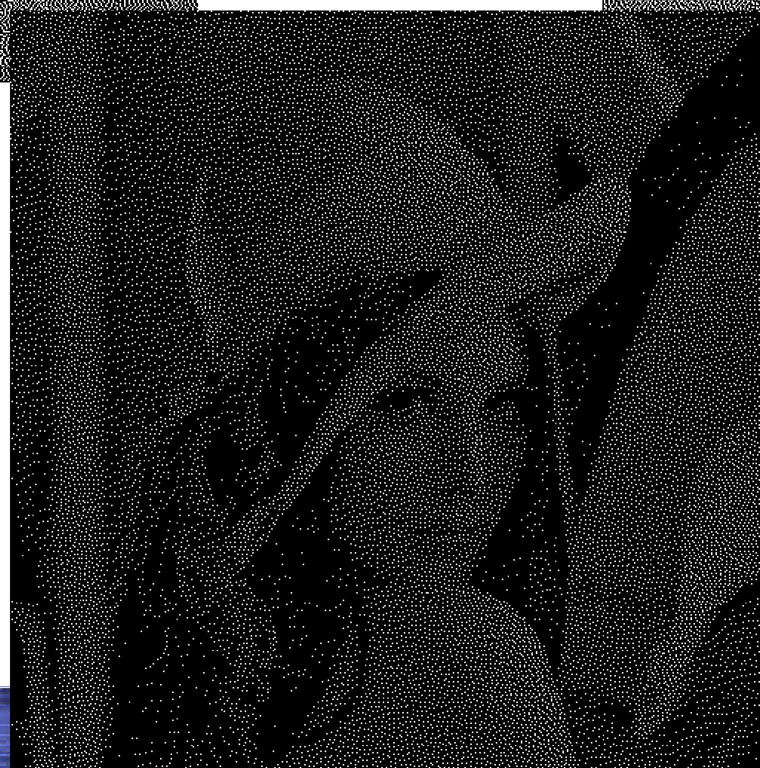
Watermarking via overlays of images

- Recently, several watermarking schemes are proposed where watermark extraction is done by overlaying two halftone images. This allows the extraction of the watermark without any complicated signal processing.
- We can do this type of watermarking using the previous framework with $n = 2$, $m = 1$.

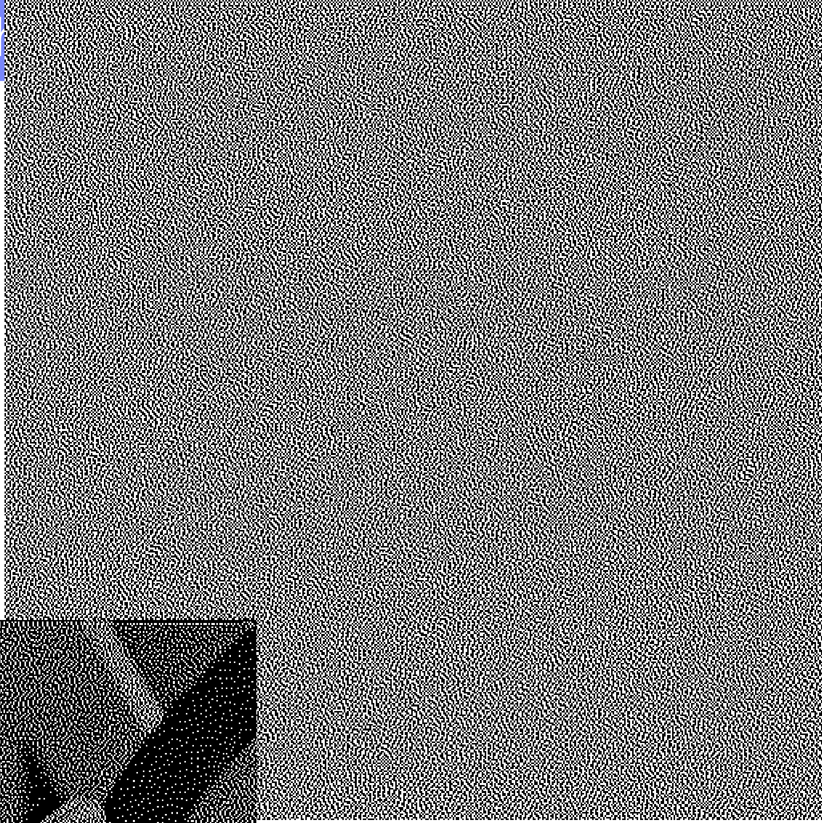


OR

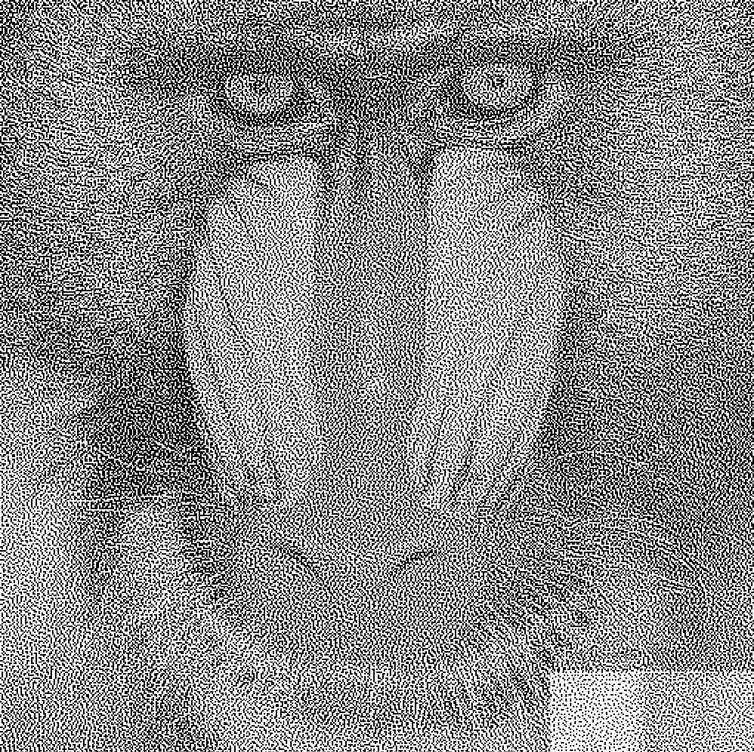
=



OR



=

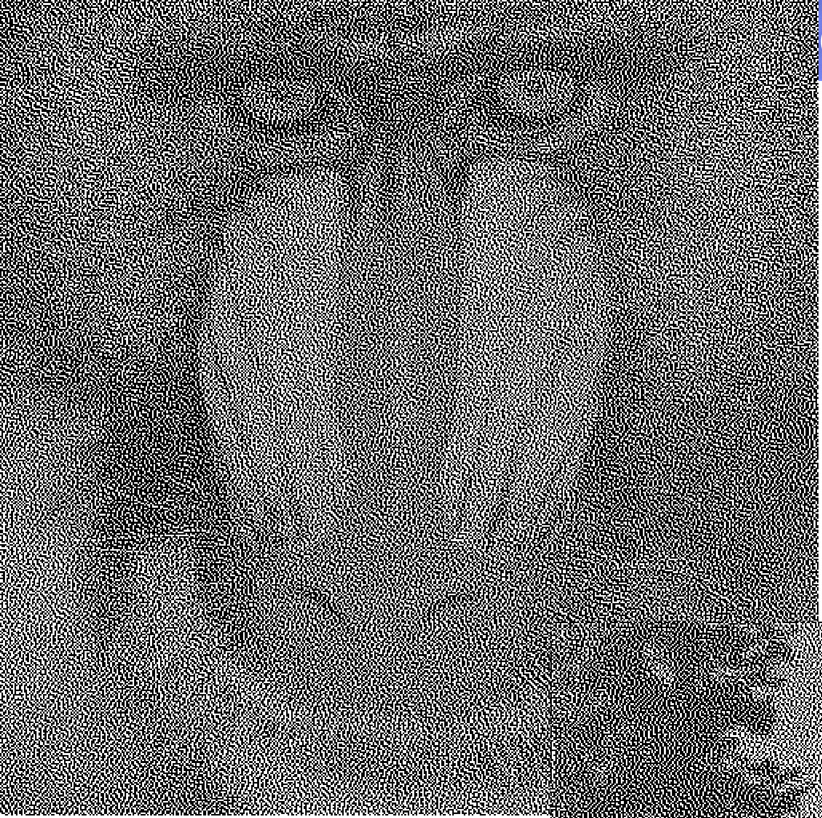


XOR

=

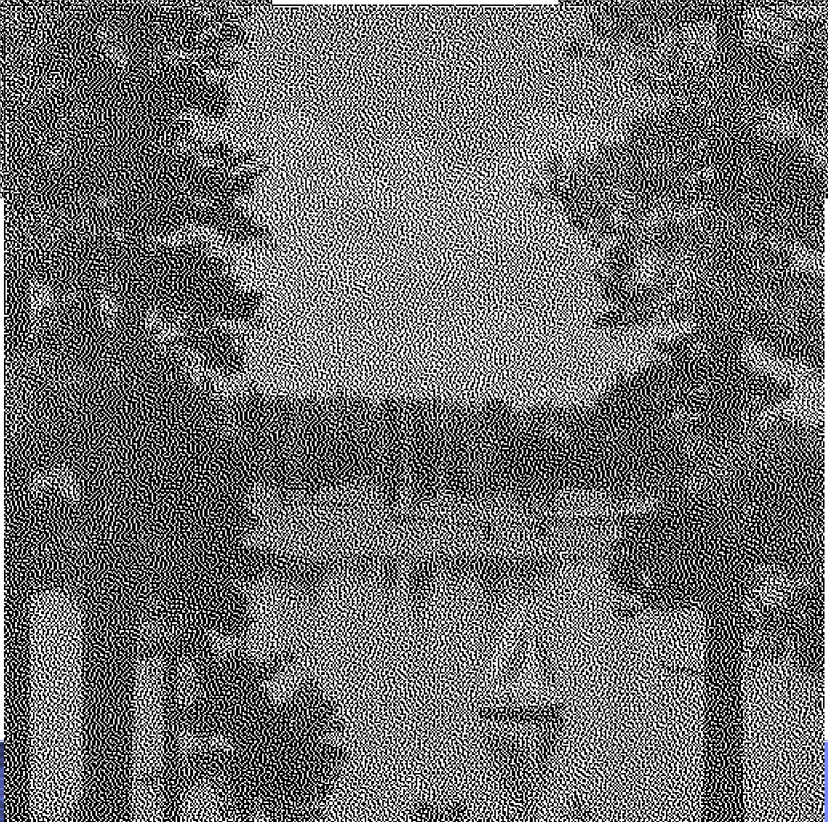
Combining multiple images

- This approach can be applied to the case $n > 2$, $m > 1$.
- For instance, consider $n = 3$, $m = 2$. Three images are combined in two different ways to produce two watermark images.



1

2



3

1 OR 2

1 OR 3

Overlaying an image with a rotated version of itself

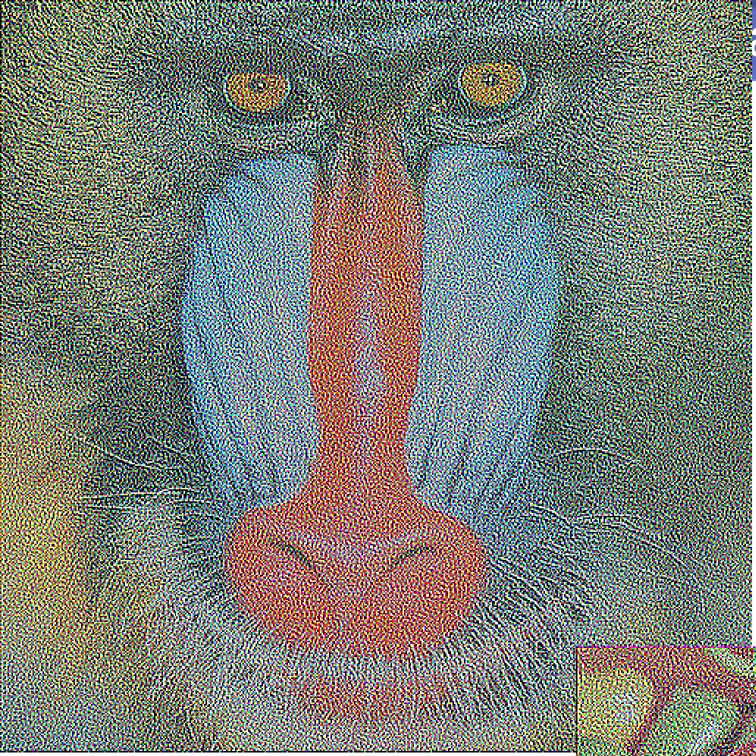
- In this application, A_1 is a rotated version of A_2, and this constraint is incorporated into the halftoning algorithm.
- Error diffusion is not appropriate as the halftoning algorithm because of the relationship between pixels which are far apart. An isotropic iterative halftoning algorithm is used instead.

OR

=

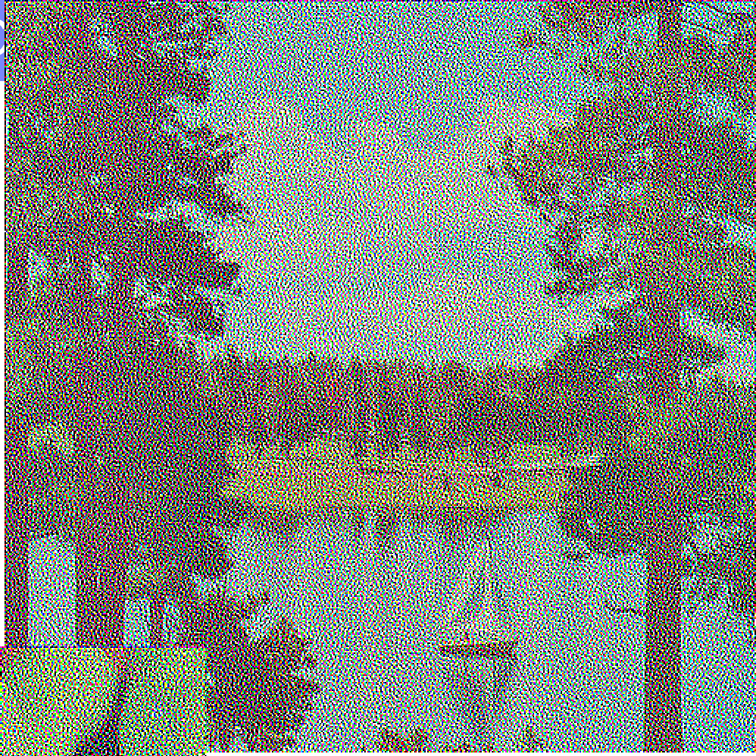
Multibit and Color images

- The same approach can be applied to generate multibit output images.
- For color images, extraction of watermark can be done in 2 ways.
  1. Each color plane is processed separately.
  2. Watermark extraction function is an n-ary function on multidimensional color space.
- Similarly, creation of the watermarked color images can be done by:
  1. Halftoning each color plane independently.
  2. Performing digital halftoning in the Cartesian product of multidimensional color spaces.



XOR

RGB color planes



=

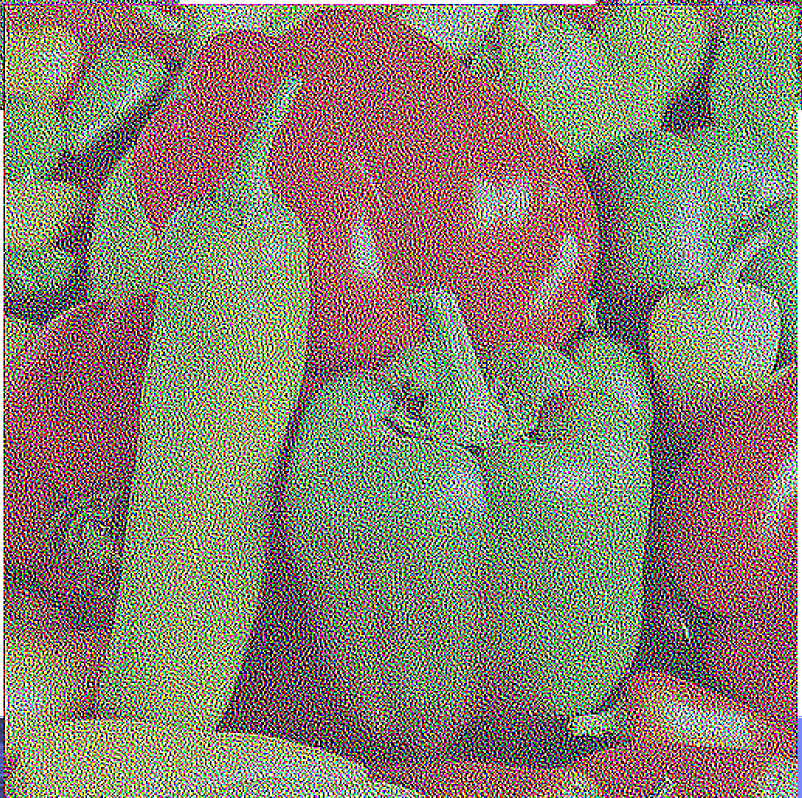
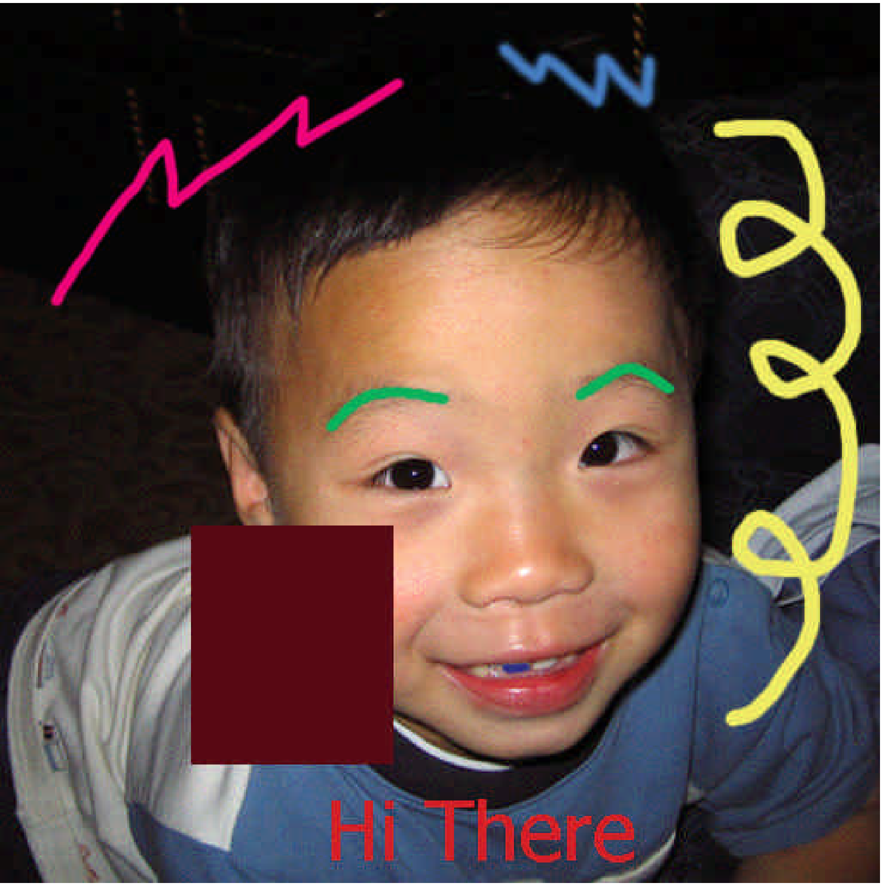


Image watermarking/self-repairing images

- In image and video watermarking, there is an interest in using watermarks to repair corrupted data.
- Using the same approach as before with n=1, m=1, embed a scrambled version of the same image into image.
- Scrambled version is used to authenticate and identify/repair corrupt data.
- Possible since watermark can have the same order of complexity as the cover image.



Corrupted watermarked image

Recovered image

Images in multiple viewing conditions

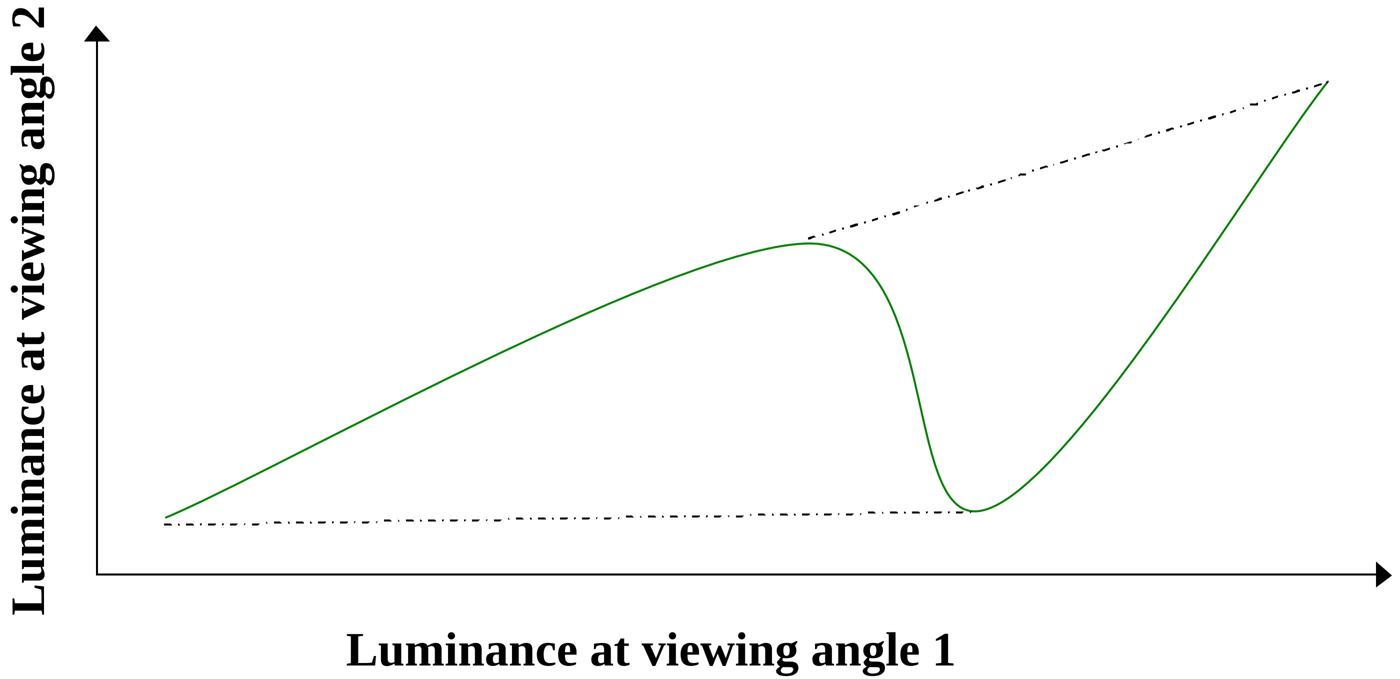
- Pixels behave differently depending on viewing conditions (VC) such as temperature, viewing angle, lighting, pressure, etc.
- If the VC's are independent and do not interact, then independent images can be shown on the various VC's.
- If the VC's are dependent, the question is whether different images can still be shown.
- Error diffusion on Cartesian products of color spaces provides a positive answer to this question.
- Different applications are possible, depending on the viewing condition.

Twisted Nematic (TN) mode LCD displays

- Used in many laptop computers and portable devices such as PDA, and digital cameras.
- Pixel intensity and color change dramatically with viewing angle.
- Exploit this shortcoming to embed two color images into the screen.
- Different images are displayed depending on viewing angle.
- Apply the above framework for n = 1, m = 1.

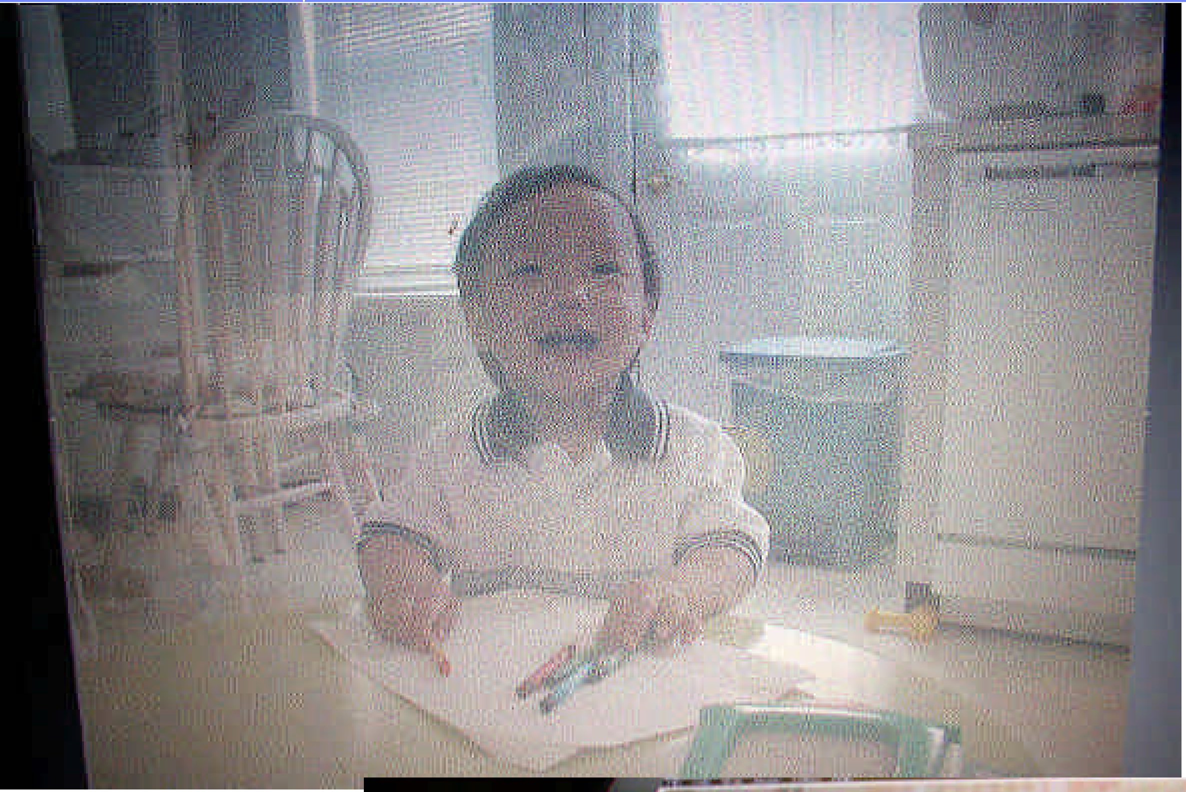
Error diffusion in Cartesian product of color spaces

- Use gamut map to transform image pairs into convex hull of curve and apply error diffusion on the space $S \times S$.



Angle 1

Angle 2



Apply to RGB channels separately

Angle 1

Angle 2

Apply to RGB channels separately

Angle 1

Angle 2