Обработка видео

Denoising u Deblocking

Дмитрий Ватолин

Московский Государственный Университет
CS MSU Graphics&Media Lab

Version 1.0

Благодарности

Автор выражает признательность Сергею Гришину и Дарье Калинкиной за помощь в подготовке этих слайдов

Denoising: Содержание

GRAPHICS & MEDIA LAB
VIDEO GROUP

- **Виды шума и их источники**
- **Методы подавления шума**
  - Методы, работающие в пространственной области
  - Методы, работающие во временной области
    (как использующие, так и не использующие компенсацию движения)
  - Методы удаления вертикальных царапин со старых кинопленок
- **MSU Noise Remover**
- **Методы оценки качества шумоподавления**

Виды шума

- Самый распространенный – **белый гауссовский шум** (описывается следующей функцией плотности распределения: $p(d) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}}$)
- **Биение пикселов** (изолированные точки на изображении, значение которых значительно отличается от значений окружающих их точек)
- **Вертикальные царапины** (характерны для старых черно-белых видеозаписей)

Происхождение шума

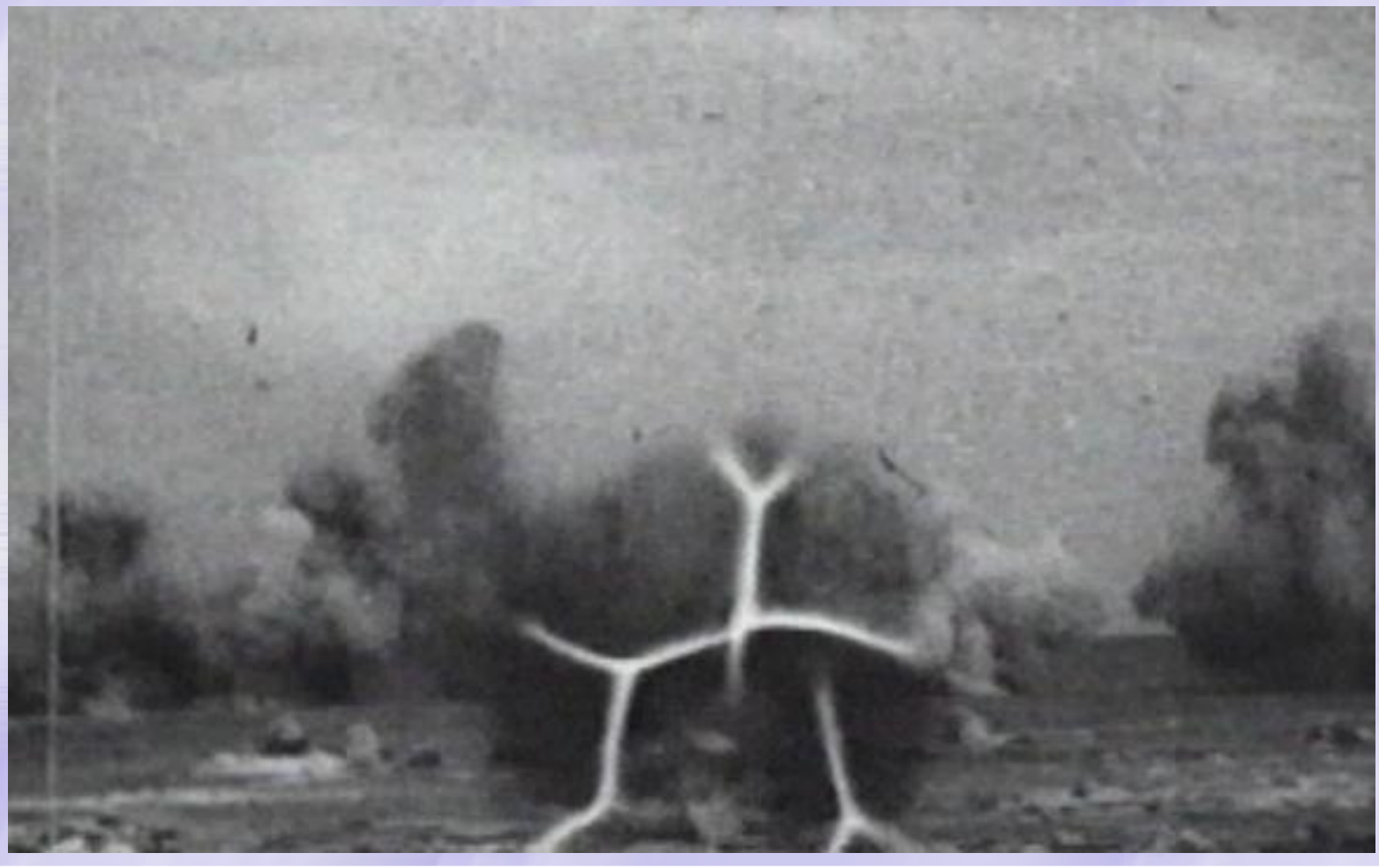
GRAPHICS & MEDIA LAB
VIDEO GROUP

- Источниками белого и импульсного шума могут быть:
  - неидеальное оборудование для захвата изображения (видеокамера и.т.п.)
  - помехи при передаче по аналоговым каналам
  - неточности при выделении яркостного и цветоразностных сигналов из аналогового композитного сигнала
  - и. т. д.
- Вертикальные царапины возникают при механическом повреждении эмульсии на пленке.

Шум в телевидении

Шум на старых фильмах

GRAPHICS & MEDIA LAB
VIDEO GROUP



Зачем нужно подавлять шум

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Улучшается визуальное качество изображения
- Подавление шума очень важно при сжатии видео:
  - Уменьшается межкадровая разница, и, как следствие, увеличивается степень сжатия
  - Лучше работают алгоритмы компенсации движения

Типы алгоритмов шумоподавления

GRAPHICS & MEDIA LAB
VIDEO GROUP

Все алгоритмы шумоподавления можно разделить на два типа:

- **Пространственные** (spatial)

  Основная идея : усреднение значений пикселей на каждом отдельном кадре.

  Проблема: могут пострадать мелкие детали, соизмеримые по размеру с шумом и четкость краев предметов.

- **Временные** (temporal)

  Основная идея: усреднение значений пикселей между кадрами.

  Проблема: при сильном движении появляются такие артефакты, как motion blur и ghosting.

Методы пространственного шумоподавления

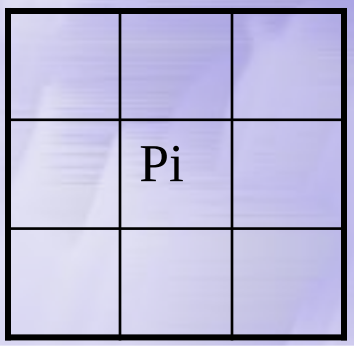
GRAPHICS & MEDIA LAB
VIDEO GROUP

- Линейное усреднение пикселов по соседям
- Вейвлет-преобразование
- Медианная фильтрация
- Гауссовское размытие
- Математическая морфология

2D Cleaner

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
for (each pixel of the current video frame){
      GetRGB (source_pixel, r, g, b);
      tot_red = tot_green = tot_blue = 0;
      count_red = count_green = count_blue = 0;
      for (each pixel in the specified radius){
            GetRGB (neighbour_pixel, r1, g1, b1);
   if  (abs(r1-r) < Threshold) tot_red += r1;  count_red ++;
   if  (abs(g1-g) < Threshold) tot_green += g1;  count_green ++;
   if  (abs(b1-b) < Threshold) tot_blue += b1;  count_blue ++;
      }
      destination_pixel = RGB (tot_red/count_red,
                               tot_green / count_green ,
                               tot_blue / count_blue );
}
```



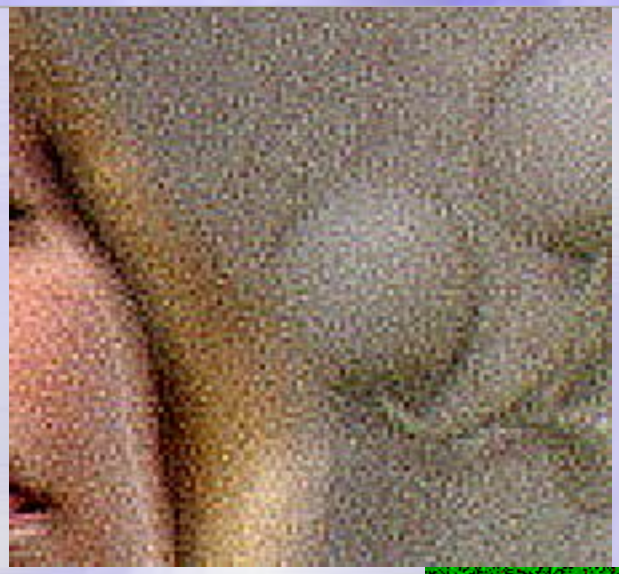
Параметры:

Threshold

Radius

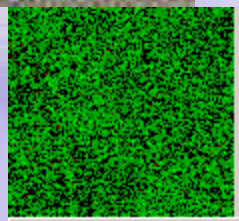
2D Cleaner: Results

GRAPHICS & MEDIA LAB
VIDEO GROUP



Source image

Processed image



1-st image: X:\Report\input.bmp
2-nd image: X:\Report\2d cleaner.bmp

MSE: 0,839	NMSE: 0,029	Max Diff RGB: 7,1
SNR: 15,352	PSNR: 40,761	Max Diff LUV: 4,2
Black: 32,8%	Green: 66%	Red: 1%

Comparison of the source and processed images using LUV Metric

Spatial Smoother

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
for  (i=0; i<=511; ++i)  square_tab[i] = (i-255) ^ 2;

for (each pixel of the current video frame){
      GetRGB (source_pixel, r, g, b);
      tot_red = tot_green = tot_blue = 0;
      count = 0;
      r_tab = square_tab[255 - r];
      g_tab = square_tab[255 - g];
      b_tab = square_tab[255 - b];
       ...
```

Spatial Smoother

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
for (each pixel in the specified radius){
  GetRGB (neighbour_pixel, r1, g1, b1);
  square_error = (r_tab[r1] + g_tab[g1] + b_tab[b1])
                              >> Strength;
  if (square_error  > 16) square_error = 16;
  square_error = 16 - square_error ;
  tot_red += r1* square_error;
  tot_green += g1* square_error;
  tot_blue += b1* square_error;
  count += square_error;
  }
  destination_pixel = RGB (tot_red/count,
                           tot_green / count,
                           tot_blue / count);
}
```

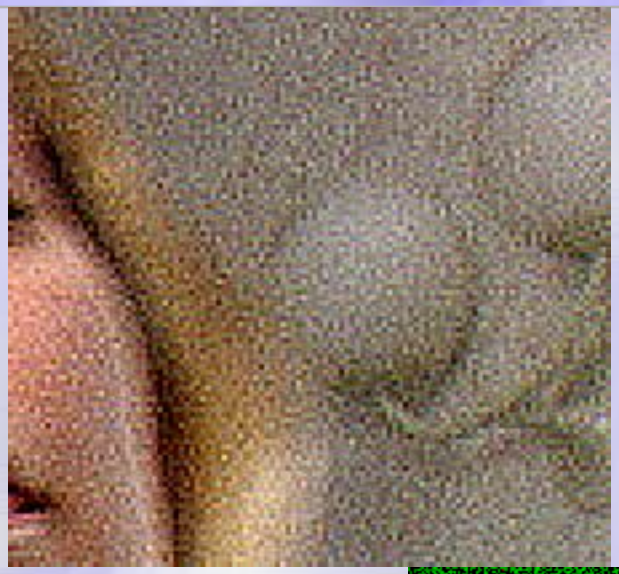
Параметры:

Diameter

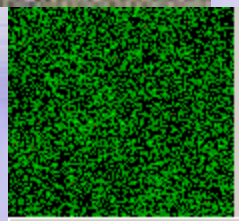
Strength

Spatial Smoother: Results

GRAPHICS & MEDIA LAB
VIDEO GROUP



Source image



Processed image

1-st image: X:\Report\input.bmp
2-nd image: X:\Report\spartial smoother.bmp

MSE: 0,363	NMSE: 0,012	Max Diff RGB: 4,6
SNR: 18,981	PSNR: 44,389	Max Diff LUV: 3,4
Black: 50,1%	Green: 49,7%	Red: 0,1%

Comparison of the source and processed images using LUV Metric

2-D Spatial Noise Filter (G. Haan)

GRAPHICS & MEDIA LAB
VIDEO GROUP

Это усредняющий по соседям фильтром с двумя порогами, рекурсивный по-вертикали *(он создавался для телевизором с небольшим объемом памяти, а элементы вертикальной задержки являются наиболее дорогими с этой точки зрения)*.

2-D Spatial Noise Filter

GRAPHICS & MEDIA LAB
VIDEO GROUP

Результирующее значение пикселя определяется по следующей формуле:

$$F_F(\underline{x}, t) = G(\underline{x}, t) \cdot \left(\sum_{\underline{n} \in N_1} K_1(\underline{x}, \underline{n}, t) \cdot F(\underline{x} + \underline{n}, t) + \sum_{\underline{n} \in N_2} K_2(\underline{x}, \underline{n}, t) \cdot F_F(\underline{x} + \underline{n}, t) \right)$$

$$K_1(\underline{x}, \underline{n}, t) = f(|F(\underline{x}, t) - F(\underline{x} + \underline{n}, t)|)$$

$$K_2(\underline{x}, \underline{n}, t) = f(|F(\underline{x}, t) - F_F(\underline{x} + \underline{n}, t)|)$$

- весовые коэффициенты, где

$$f(a) = \begin{cases} 1, & a \le Th \\ 0.25, & Th < a \le 4Th \\ 0, & a > 4Th \end{cases}$$

- монотонно убывающая функция
(Th – параметр, зависящий от уровня шума)

2-D Spatial Noise Filter

GRAPHICS & MEDIA LAB
VIDEO GROUP

$$F_F(\underline{x}, t) = G(\underline{x}, t) \cdot \left(\sum_{\underline{n} \in N_1} K_1(\underline{x}, \underline{n}, t) \cdot F(\underline{x} + \underline{n}, t) + \sum_{\underline{n} \in N_2} K_2(\underline{x}, \underline{n}, t) \cdot F_F(\underline{x} + \underline{n}, t) \right)$$

$$\frac{1}{G(\underline{x}, t)} = \sum_{\underline{n} \in N_1} K_1(\underline{x}, \underline{n}, t) + \sum_{\underline{n} \in N_2} K_2(\underline{x}, \underline{n}, t)$$

- нормализующий коэффициент

Пиксель смешивается с соседями, определяемыми следующими векторами (***w*** последовательно принимает значения *–1,0,1,0*):

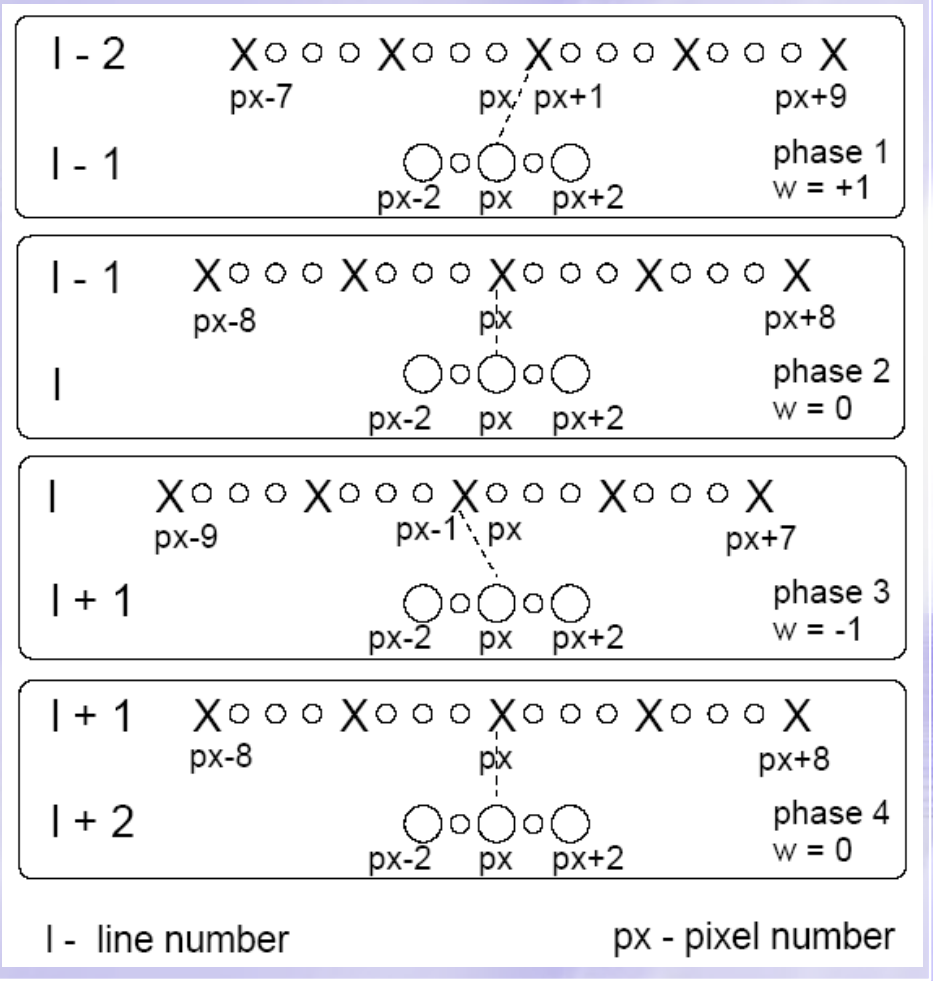
$$N_1 = \left\{ \begin{bmatrix} -2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\}$$

$$N_2 = \left\{ \begin{pmatrix} -8+w \\ -1 \end{pmatrix}, \begin{pmatrix} -4+w \\ -1 \end{pmatrix}, \begin{pmatrix} w \\ -1 \end{pmatrix}, \begin{pmatrix} 4+w \\ -1 \end{pmatrix}, \begin{pmatrix} 8+w \\ -1 \end{pmatrix} \right\}$$

2-D Spatial Noise Filter

Иллюстрация работы фильтра на 4-х последовательных строках.

Вектор N_2 изменяется от строки к строке, таким образом фильтр работает циклически.



2-D Spatial Noise Filter

GRAPHICS & MEDIA LAB
VIDEO GROUP

Особенность данного фильтра:

- В большинстве других пространственных фильтрах пиксели смешиваются со своими непосредственными соседями и в симметричном окне.

  Аргументация: при такой обработке шумовые различия между пикселями в окне сглаживаются в наибольшей степени.

- В данном фильтре смешиваются пиксели, находящиеся на некотором расстоянии друг от друга.

  Аргументация: таким способом удается подавить низкочастотный шум, который более заметен, чем высокочастотный, и практически не затронуть наименее заметные высокие диагональные частоты в 2D спектре благодаря периодической смене смешиваемых пикселей.

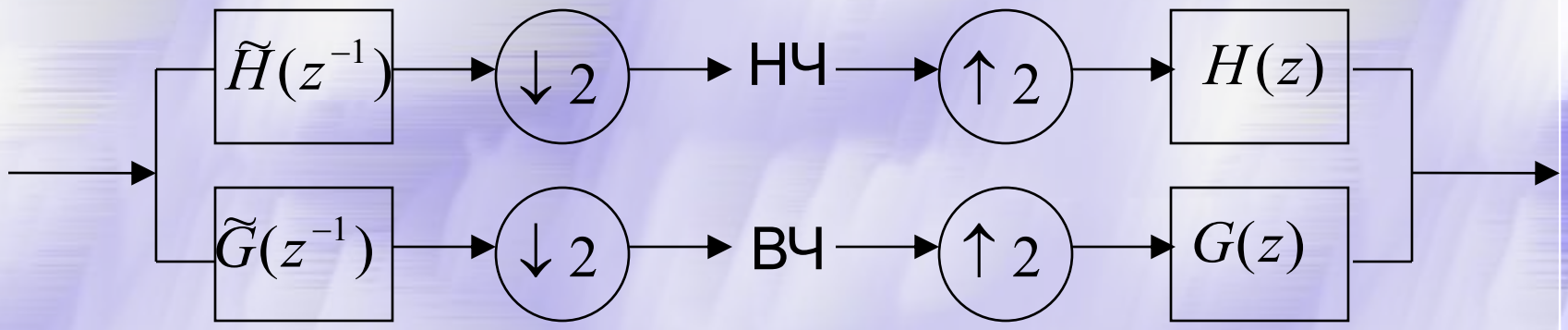
Вейвлеты

GRAPHICS & MEDIA LAB
VIDEO GROUP

Дискретное вейвлет-преобразование (DWT):

$\tilde{\mathbf{h}}$ $\tilde{\mathbf{g}}$ $\mathbf{h}$ $\mathbf{g}$

$\tilde{H}(z)$ $\tilde{G}(z)$ $H(z)$ $G(z)$



DWT (на примере фильтра Хаара)

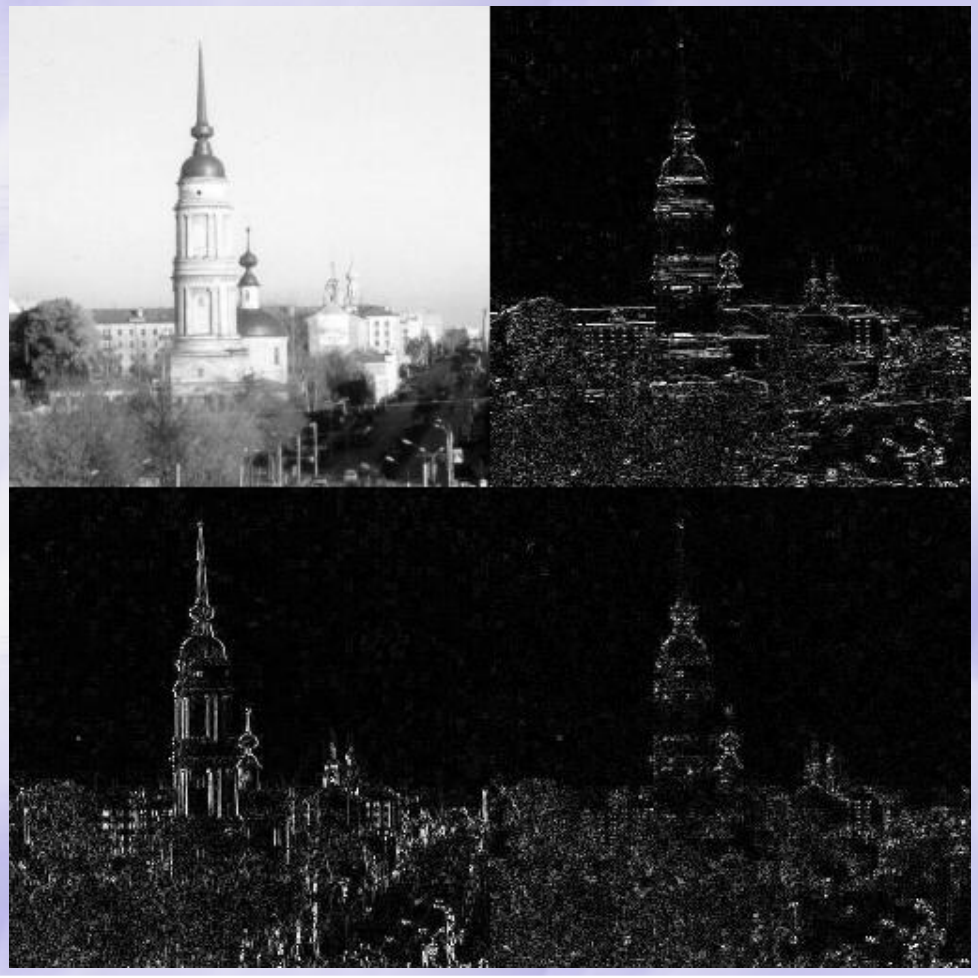
GRAPHICS & MEDIA LAB
VIDEO GROUP

$$\tilde{\mathbf{h}} = [1/2,\ 1/2] \qquad \mathbf{h} = [1,\ 1]$$
$$\tilde{\mathbf{g}} = [-1/2,\ 1/2] \qquad \mathbf{g} = [1,\ -1].$$

Resolution	Averages	Detail coefficients
3	[2, 2, 0, 2, 3, 5, 4, 4]	
2	[2, 1, 4, 4]	[0, –1, –1, 0]
1	[1.5, 4]	[0.5, 0]
0	[2.75]	[–1.25]

Haar wavelet decomposition: [2.75, –1.25, 0.5, 0, 0, –1, –1, 0]

2D DWT



2D DWT:
DWT применяется сначала к строкам, потом к столбцам изображения.

LL	HL
LH	HH

DWT и шумоподавление

Информация о шуме содержится в ВЧ-составляющих (вейвлет-коэффициентах) DWT.

Если на изображении **f** = [**f1**, ..., **fn**] присутствует белый шум с нулевым математическим ожиданием и дисперсией σ^2, то в силу линейности DWT для вейвлет-коэффициентов будет выполняться:

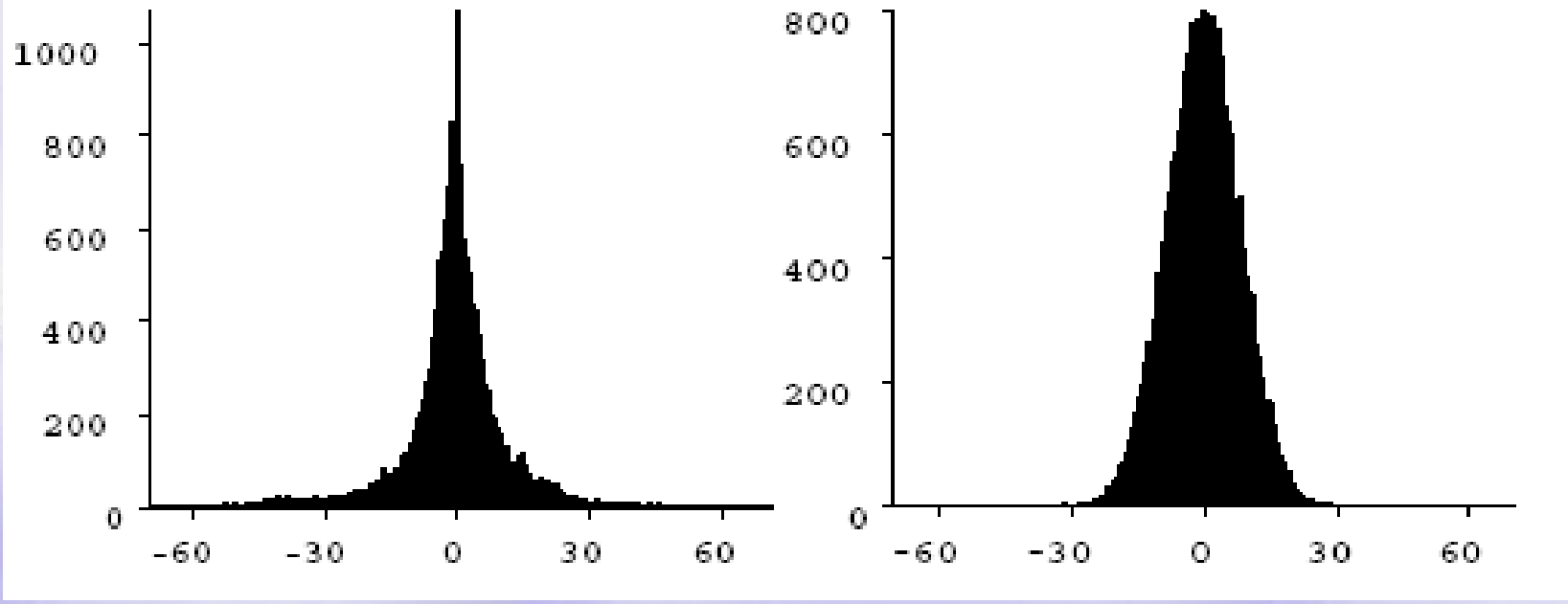
$$wi = yi + \epsilon i,\ i = 1, ..., n$$

где yi - вейвлет-коэффициенты без шума,

а $\epsilon i \sim N(0, \sigma^2)$ - шум

DWT и шумоподавление

Слева – гистограмма вейвлет-коэффициентов у чистого изображения, справа – у изображения с шумом.



DWT и шумоподавление

GRAPHICS & MEDIA LAB
VIDEO GROUP

- При шумоподавлении обычно используется вейвлет-преобразование без прореживания, то есть примененное к каждому пикселу.
- Шумоподавление выполняется путем уменьшения значений вейвлет-коэффициентов в зависимости от уровня шума и вероятности того, что данный коэффициент представляет собой шум.
- Проблема – как отличить шум от деталей изображения.

DWT и шумоподавление

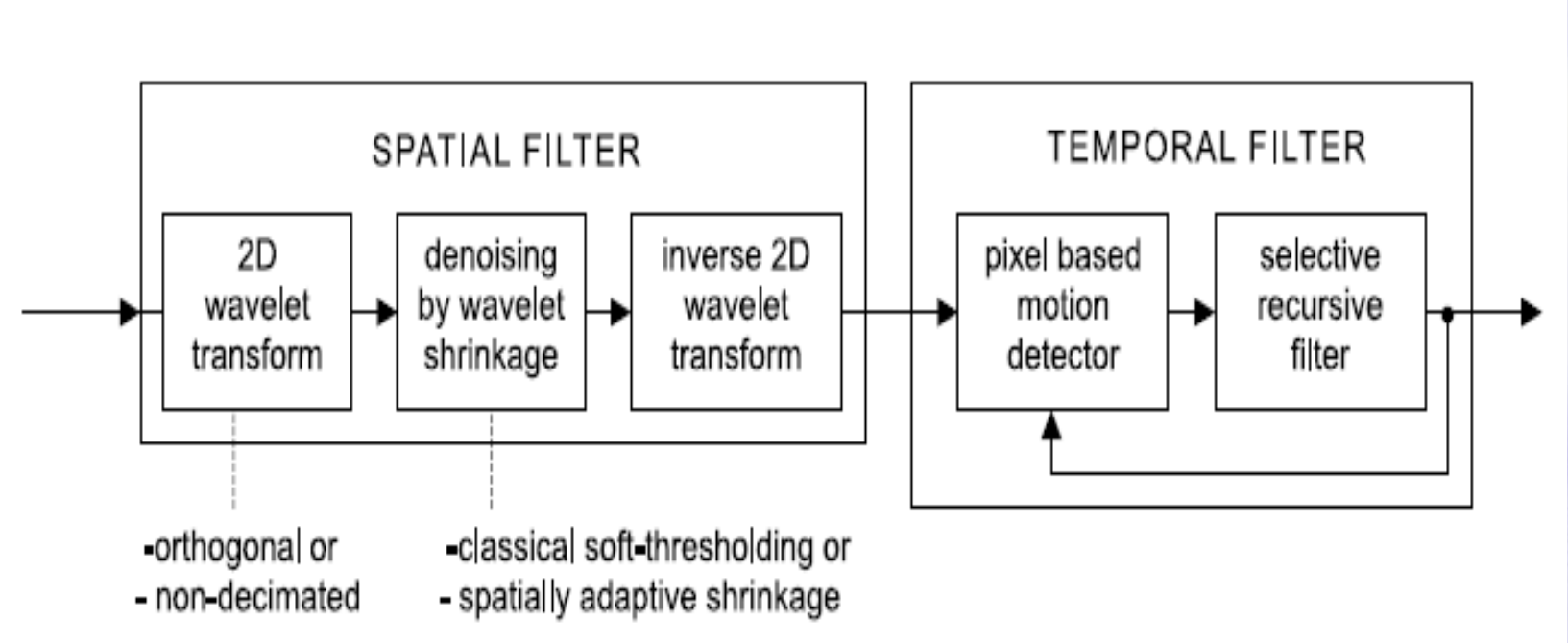
Можно выделить два основных подхода:

- **Thresholding**
  - hard-thresholding : $$\tilde{w}i = \begin{cases} 0, & |wi| \le T \\ wi, & |wi| > T \end{cases}$$
  - soft-thresholding: $$\tilde{w}i = \begin{cases} 0, & |wi| \le T \\ \mathrm{sgn}(wi)(|wi| - T), & |wi| > T \end{cases}$$
- **Spatially adaptive shrinkage**

Общая схема шумоподавления с использованием вейвлетов

GRAPHICS & MEDIA LAB
VIDEO GROUP

Обычно объединяют пространственный вейвлет-фильтр с некоторым временным фильтром.



Медианная фильтрация

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Медианная фильтрация – это стандартный способ подавления импульсного шума.
- Для каждого пиксела вычисляется среднее значение в некотором его окружении (окне) и присваивается этому пикселу:

$$med = \arg\min_{f_i \in W} \sum_j \left| f_i - f_j \right|$$

где W – окно размера (2n+1)x(2n+1)

Vector Median Filter (VMF)

Для цветных изображений используется векторный медианный фильтр (VMF):

$$med = \arg\min_{f_i \in W} \sum_{j=0}^{N-1} d\left(F_i, F_j\right)$$

где d – произвольная метрика
(Euclidean or city-block)

Однако чистый медианный фильтр размывает края, поэтому на практике практически не используется.

Fast Modified Vector Median Filter (FMVMF)

GRAPHICS & MEDIA LAB
VIDEO GROUP

Его модификация – быстрый векторный медианный фильтр (FMVMF).

В FMVMF используется окно с 4-мя соседями:

(F_0 – изменяемый пиксел)

	F_1	
F_4	F_0	F_2
	F_3	

FMVMF

Рассматриваемый центральный пиксел F_0 заменяется на F_k, где:

$$k = \begin{cases} 0, if \ -\beta + \sum_{j=1}^{N-1} d(F_0, F_j) \le \min_{F_i \in W \setminus F_0} \sum_{j=1}^{N-1} d(F_i, F_j) \\ i_{opt} := \arg \min_{F_i \in W \setminus F_0} \sum_{j=1}^{N-1} d(F_i, F_j), \\ \quad if \ \min_{F_i \in W \setminus F_0} \sum_{j=1}^{N-1} d(F_i, F_j) < -\beta + \sum_{j=1}^{N-1} d(F_0, F_j) \end{cases}$$

β - параметр (экспериментально равен 0.75, если цветовые компоненты нормализованы в интервале [0, 1])

Primum Non Nocere Vector Median Filter (PNN-VMF)

GRAPHICS & MEDIA LAB
VIDEO GROUP

- За основу взят FMVMF.
- Обход ведется по-горизонтали слева направо сверху вниз и в качестве F_1 и F_4 берутся уже ранее обработанные значения (предположительно, без шума).
- В FMVMF F_0 изменяется, если условие выполнялось хотя бы для одного из соседей, а в PNN-VMF - только если оно выполнено для ВСЕХ его соседей.

Two-pass Median-like Filter for impulse noise removal

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Первый шаг – PNN-VMF.
- На втором шаге выполняется одна из модификаций PNN-VMF - N2, N3, N4 – на основе того, сколько пикселей было изменено на первом шаге.

condition	triggered 2nd pass action
$FMP \le T_1$	no action, terminate
$(FMP > T_1) \wedge (FMP \le T_2)$	N4
$(FMP > T_2) \wedge (FMP \le T_3)$	N3
$FMP > T_3$	N2

Ni – это PNN-VMF, у которого для изменения центрального пиксела достаточно выполнения условия для Ni соседей из 4-х.

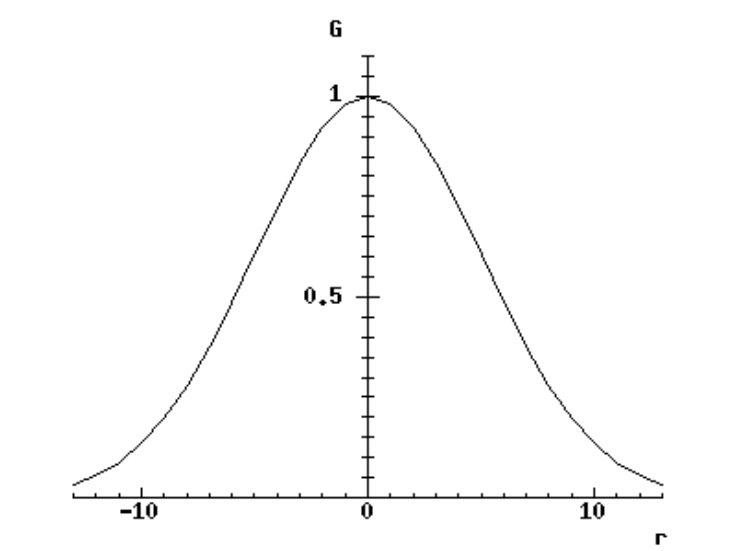
Для T1, T2 и T3 экспериментально были получены значения 0.04, 0.08 и 0.13 соответственно.

Гауссовское размытие

Это свертка по функции:

$$g(x,y) = Ae^{-\frac{x^2+y^2}{\sigma^2}}$$

Параметр σ задает степень размытия.



$$I'(i,j) = \sum_{x=-n}^{n}\sum_{y=-m}^{m} I(i-x)(j-y)\cdot\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{d^2}{2\sigma^2}}$$

$$d = \sqrt{x^2+y^2}$$

$$\begin{bmatrix} \frac{1}{4} & 1 & \frac{1}{4} \\ 1 & 5 & 1 \\ \frac{1}{4} & 1 & \frac{1}{4} \end{bmatrix}$$