

# Обзор некоторых подходов к шумоподавлению

---

Юрий Чернышов

Video Group

CS MSU Graphics & Media Lab



# Содержание

---

- **Введение**
  - Источники и виды шумов
  - Цели шумоподавления
- Алгоритмы
  - Общие сведения
  - STVF
  - 3D K-SVD
  - MMCF
  - BM3D-SARCA
- Заключение



# Источники и виды шумов

---

- Старение носителей информации (например, отслоение эмульсии)
- Артефакты кодирования



Ringing

# Источники и виды шумов

- Зернистость аналоговых носителей
- Проблемы при передаче данных по каналам
- Плохие сенсоры и условия съемки (белый Гауссов шум)

Echo added

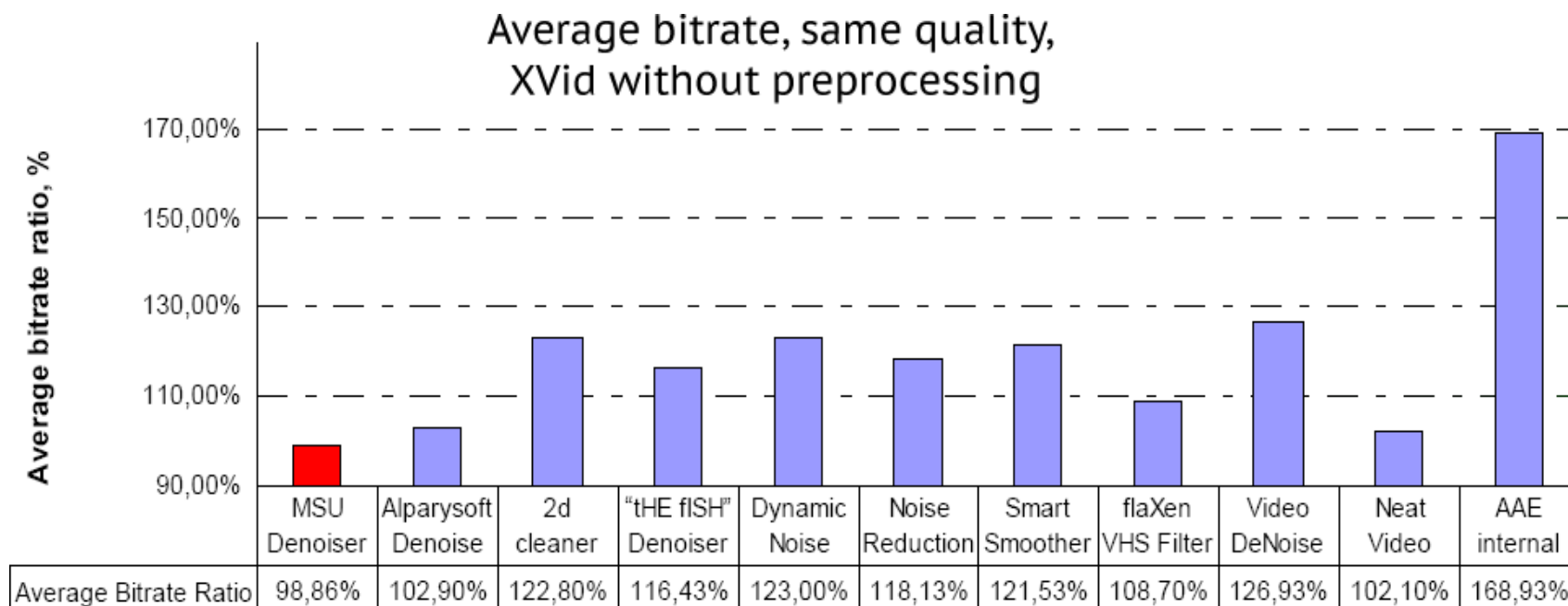


Grain



# Цели шумоподавления

Хорошее шумоподавление повышает степень сжатия:



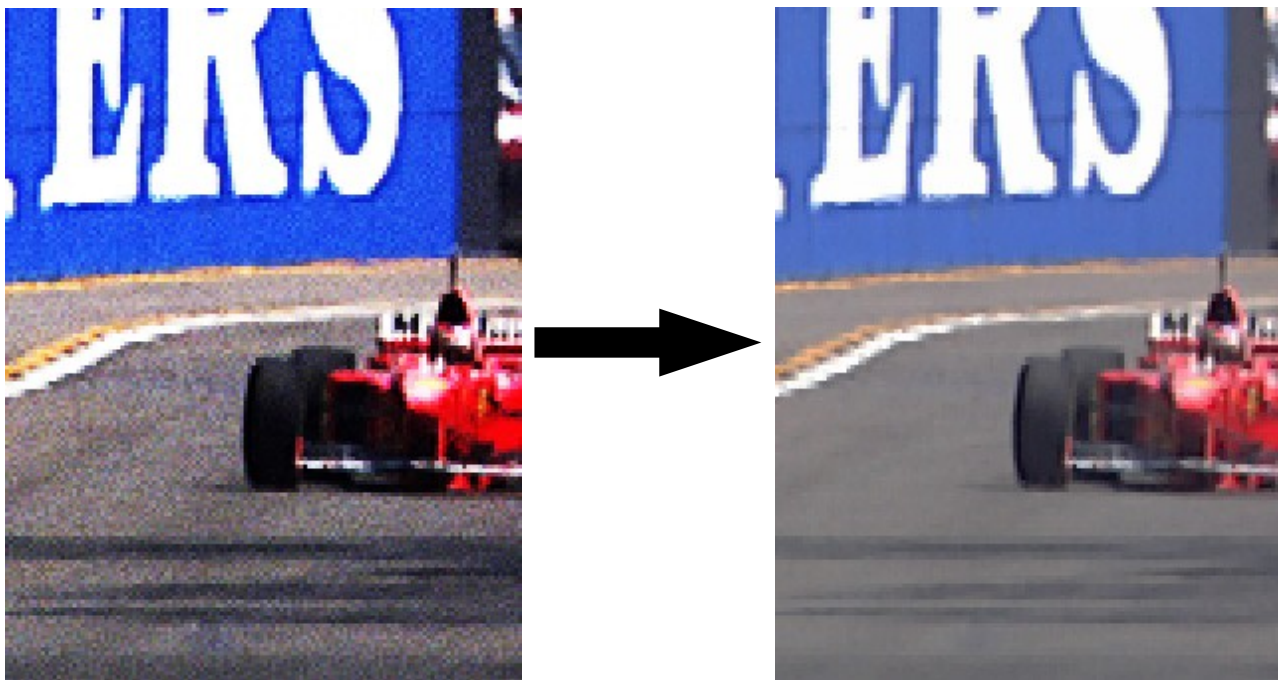
[«YUVSoft Video Denoiser Comparison», 2007](#)



# Цели шумоподавления

---

Увеличивается и субъективное качество изображения:



[«YUVSoft Video Denoiser Comparison», 2007](#)



# Содержание

---

- Введение
  - Источники и виды шумов
  - Цели шумоподавления
- Алгоритмы
  - **Общие сведения**
  - STVF
  - 3D K-SVD
  - MMCF
  - BM3D-SARCA
- Заключение



# Классификация алгоритмов

---

Три вида алгоритмов:

- Временные – усреднение текущего кадра с предыдущими или последующими
- Пространственные – усреднение текущей области (пикселя) кадра с его окружением
- Пространственно-временные – наиболее эффективные, сочетают оба подхода





# Последние достижения

---

Последние достижения основаны на:

- Gaussian Scale Mixtures
  - Bilateral filtering
  - NLM
- Learned Dictionaries
  - K-SVD
  - K-LLD
- Частотная фильтрация
- Markov Random Field

[«BM3D Image Denoising with Shape-Adaptive Principal Component Analysis» \(2009\)](#)



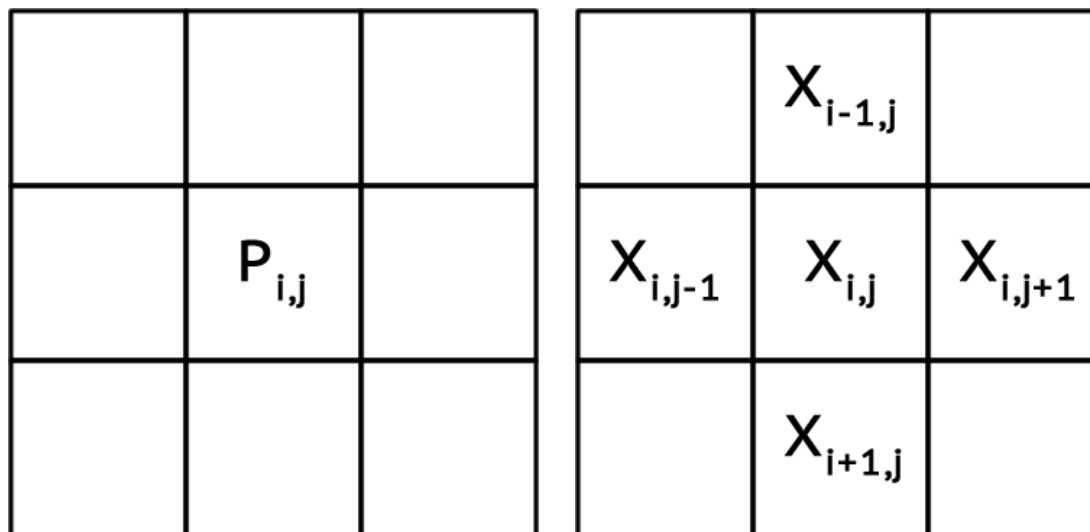
# Содержание

---

- Введение
  - Источники и виды шумов
  - Цели шумоподавления
- Алгоритмы
  - Общие сведения
  - **STVF**
  - 3D K-SVD
  - Multihypotheses temporal-only algorithm
  - BM3D-SAPCA
- Заключение

# STVF: определение шума

Алгоритм подавляет как белый, так и импульсный шум:



Пиксель поврежден импульсным шумом, если он достаточно (больше, чем на  $T_1$ ) отличается от каждого из пяти своих соседей.

[«A novel content-adaptive video denoising filter» \(ICASSP 2005\)](#)

# STVF: фильтрация

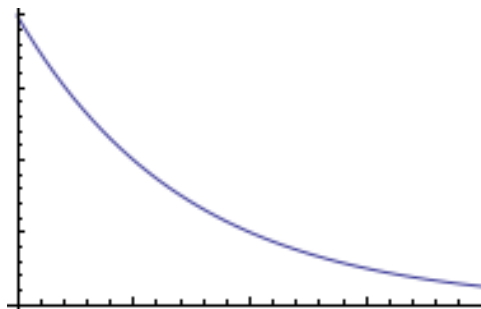
Подавление импульсного шума:

$$y_{ij} = \frac{x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}}{4}$$

Подавление белого шума:

$$y_{ij} = \frac{\sum_{x \in S} f(|x - x_{ij}|) * x}{\sum_{x \in S} f(|x - x_{ij}|)}, \text{ where } S = \{x_{i-1,j}, x_{i+1,j}, x_{i,j-1}, x_{i,j+1}, p_{ij}\}$$

$$f(i) = 2^{\lfloor T_1/8 \rfloor - \lfloor i/8 \rfloor}$$



[«A novel content-adaptive video denoising filter» \(ICASSP 2005\)](#)



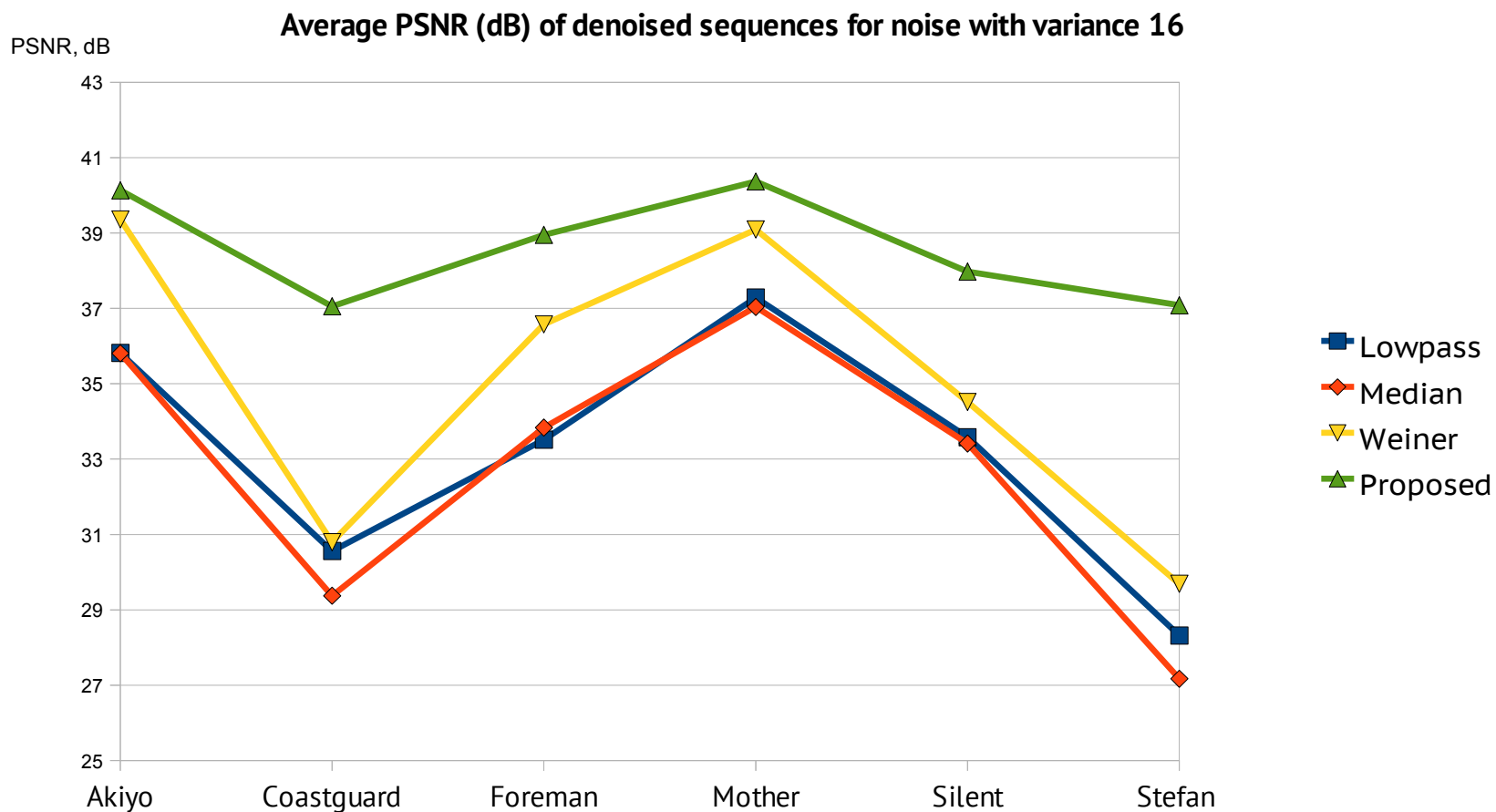
# STVF: регуляризация

Нововведением стала регуляризация изображений:

$$z_{ij} = \begin{cases} y_{ij}, & |(y_{ij} - x_{ij})| \leq T_2 \vee x_{ij} \text{ marked as impulsive} \\ x_{ij} - T_2, & y_{ij} < x_{ij} - T_2 \\ x_{ij} + T_2, & y_{ij} > x_{ij} + T_2 \end{cases}$$

[«A novel content-adaptive video denoising filter» \(ICASSP 2005\)](#)

# STVF: результаты (1)



[«A novel content-adaptive video denoising filter» \(ICASSP 2005\)](#)



# STVF: результаты (2)

Source



Noisy



[«A novel content-adaptive video denoising filter» \(ICASSP 2005\)](#)



# STVF: результаты (3)

Source



Proposed



[«A novel content-adaptive video denoising filter» \(ICASSP 2005\)](#)





# STVF: результаты (4)

Noisy



Proposed



[«A novel content-adaptive video denoising filter» \(ICASSP 2005\)](#)



# STVF: ВЫВОДЫ

---

Достоинства:

- Скорость (линейная сложность, один проход)
- Независимость вычислений

Недостатки:

- Неочевидность значений  $T_1$  и  $T_2$
- Подавление только пиксельных шумов

[«A novel content-adaptive video denoising filter» \(ICASSP 2005\)](#)



# Содержание

---

- Введение
  - Источники и виды шумов
  - Цели шумоподавления
- Алгоритмы
  - Общие сведения
  - STVF
  - **3D K-SVD**
  - Multihypotheses temporal-only algorithm
  - BM3D-SAPCA
- Заключение



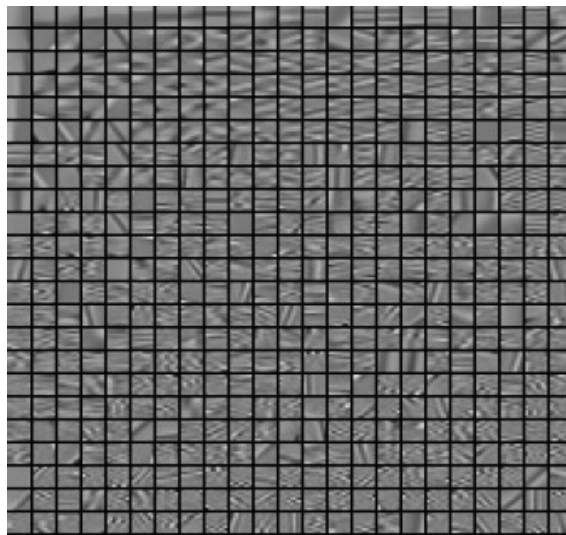
# K-SVD: идея метода

---

Назовем патчем некоторый блок изображения.

Представим каждый патч кадра как линейную комбинацию известных патчей (атомов) из словаря.

В словаре – порядка 300 атомов  $6 \times 6$  пикселей каждый.



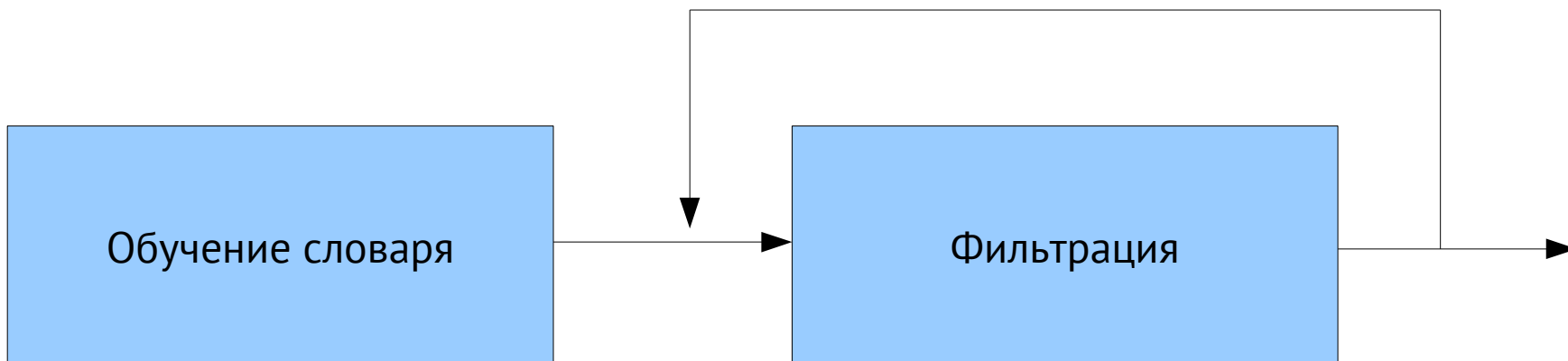
[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)



# K-SVD: словарь

Качество работы алгоритма напрямую зависит от словаря.

Необходимо обучать словарь.



[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)



# K-SVD: обучение словаря

Смоделируем изображение:  $Y = X + V$ ,  $V \sim N(0, \sigma^2)$

Введем некоторую функцию штрафа:

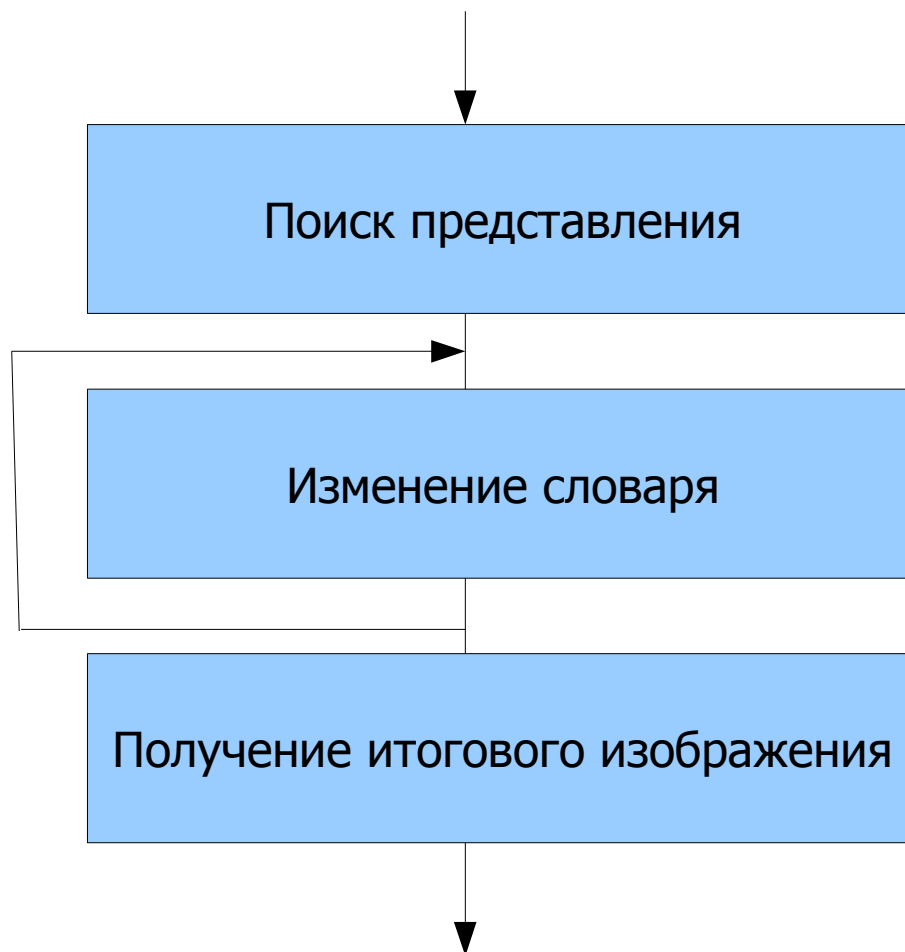
$$f_{error}(\{\alpha_{ij}\}_{ij}, X) = \lambda \|Y - X\|_2^2 + \sum_{i,j \in \Omega} \|D \alpha_{ij} - R_{ij} X\|_2^2 + \sum_{i,j \in \Omega} \mu_{ij} \|\alpha_{ij}\|_0$$

Смысл функции штрафа: результат слабо отличается от исходного изображения, каждый патч изображения (извлекается оператором  $R$ ) представляется вектором коэффициентов, при этом эти коэффициенты малы.

[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)



# K-SVD: фильтрация



[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)



# K-SVD: поиск представления

---

Предположим, что  $X$  фиксирован.

$$\hat{\alpha}_{ij} = \arg \min_{\alpha} (\|D\alpha - R_{ij}X\|_2^2 + \mu\|\alpha\|_0)$$

Для минимизации будем использовать OMP-алгоритм

- Пусть  $d$  – словарь,  $r$  – раскладываемый вектор
- Найдем атом с наибольшей корреляцией:  $k = \arg \max_k |d_k^T r|$
- Вычтем полученную величину из  $r$
- Повторим операцию достаточное число раз

[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)





# K-SVD: изменение словаря

---

Добавим в словарь то представление каждого патча, которое мы получили на предыдущем этапе.

$$D = D \cup D \alpha_{ij} \text{ для всех } i, j$$

[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)

# K-SVD: получение результата

Для получения итогового результата после нескольких итераций минимизируем следующий функционал, исходя из предположения, что разложения для каждого патча фиксированы.

$$\hat{X} = \arg \min_X (\lambda \|X - Y\|_2^2 + \sum_{i,j} \|D \alpha_{ij} - R_{ij} X\|_2^2)$$

[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)

# 3D K-SVD: расширение для видео

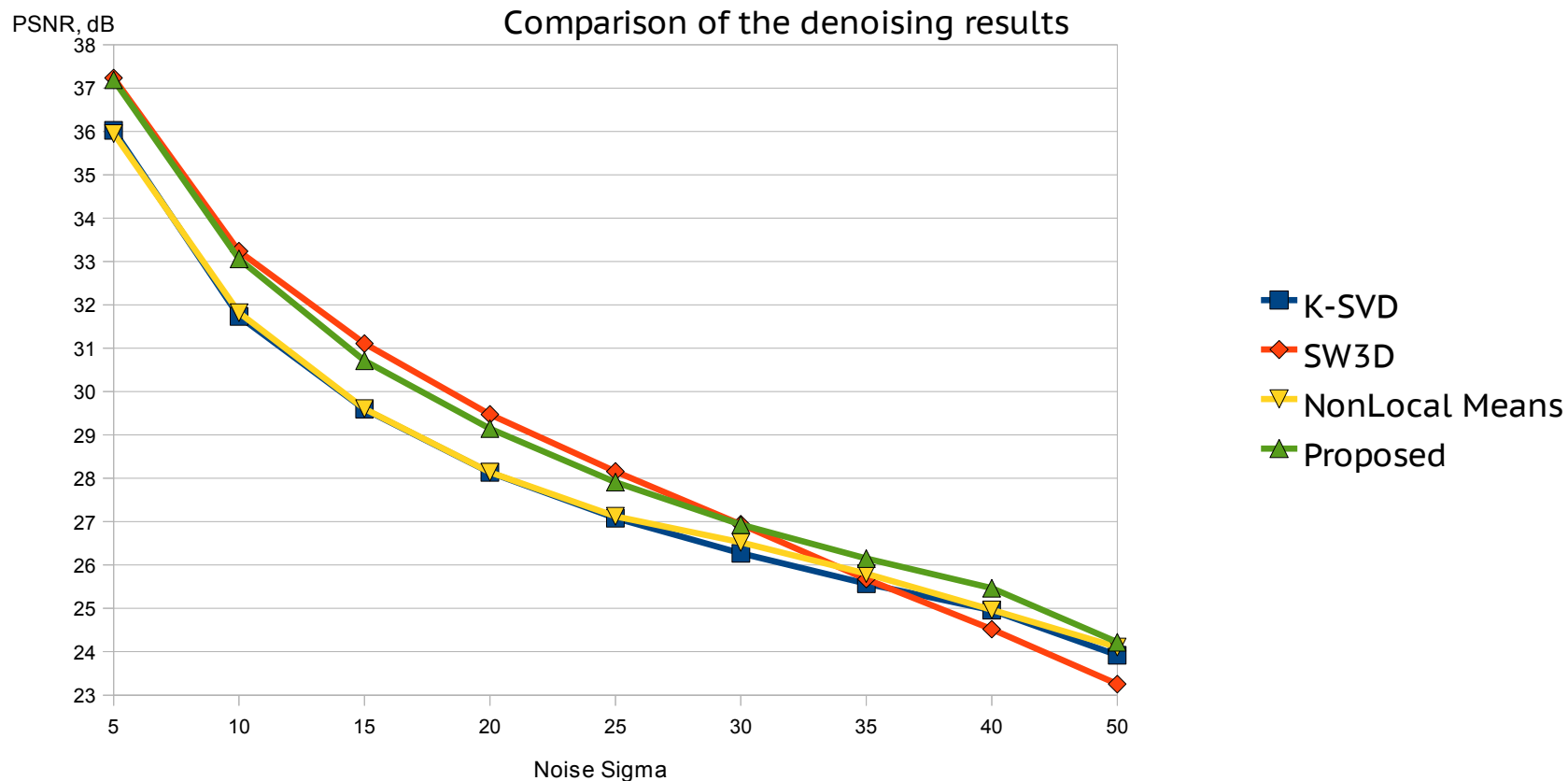
- Изменение функции штрафа:

$$f_{video}^{t \pm \Delta t}(\{\alpha_{ijk}\}_{ijk}, X_t, D_t) = \lambda \|X_t, Y_t\|_2^2 + \sum_{i,j \in \Omega} \sum_{k=t-\Delta t}^{t+\Delta t} \mu_{ijk} \|\alpha_{ijk}\|_0 + \sum_{i,j \in \Omega} \sum_{k=t-\Delta t}^{t+\Delta t} \|D_t \alpha_{ijk} - R_{ijk} X\|_2^2$$

- Словарь для текущего кадра будет похож на словарь для предыдущего
- Использование трехмерных атомов (6x6x5 пикселей)

[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)

# 3D K-SVD: результаты (1)



[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)



# 3D K-SVD: результаты (2)

Source

Noisy

Filtered



[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)





# 3D K-SVD: результаты (3)

Source

Noisy

Filtered



[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)

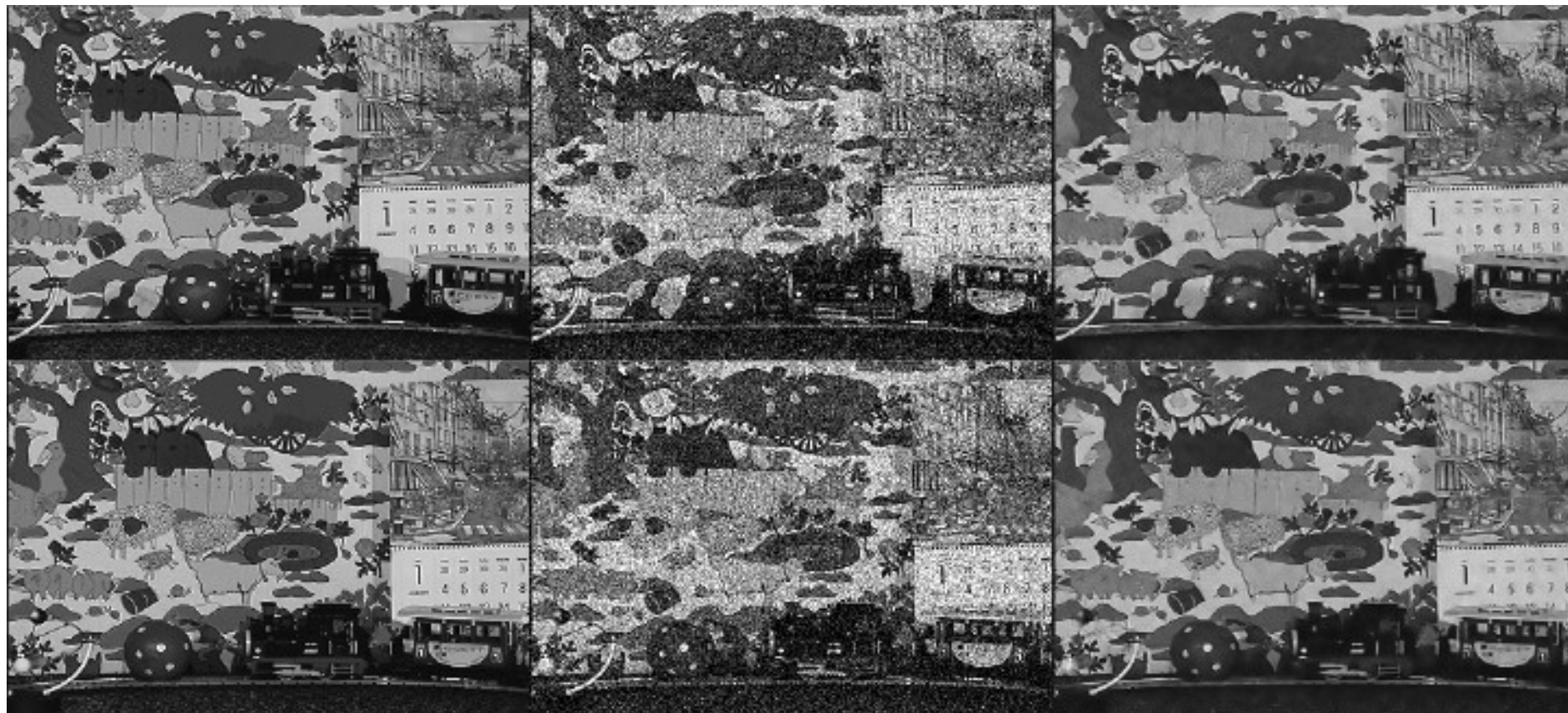


# 3D K-SVD: результаты (4)

Source

Noisy

Filtered



[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)



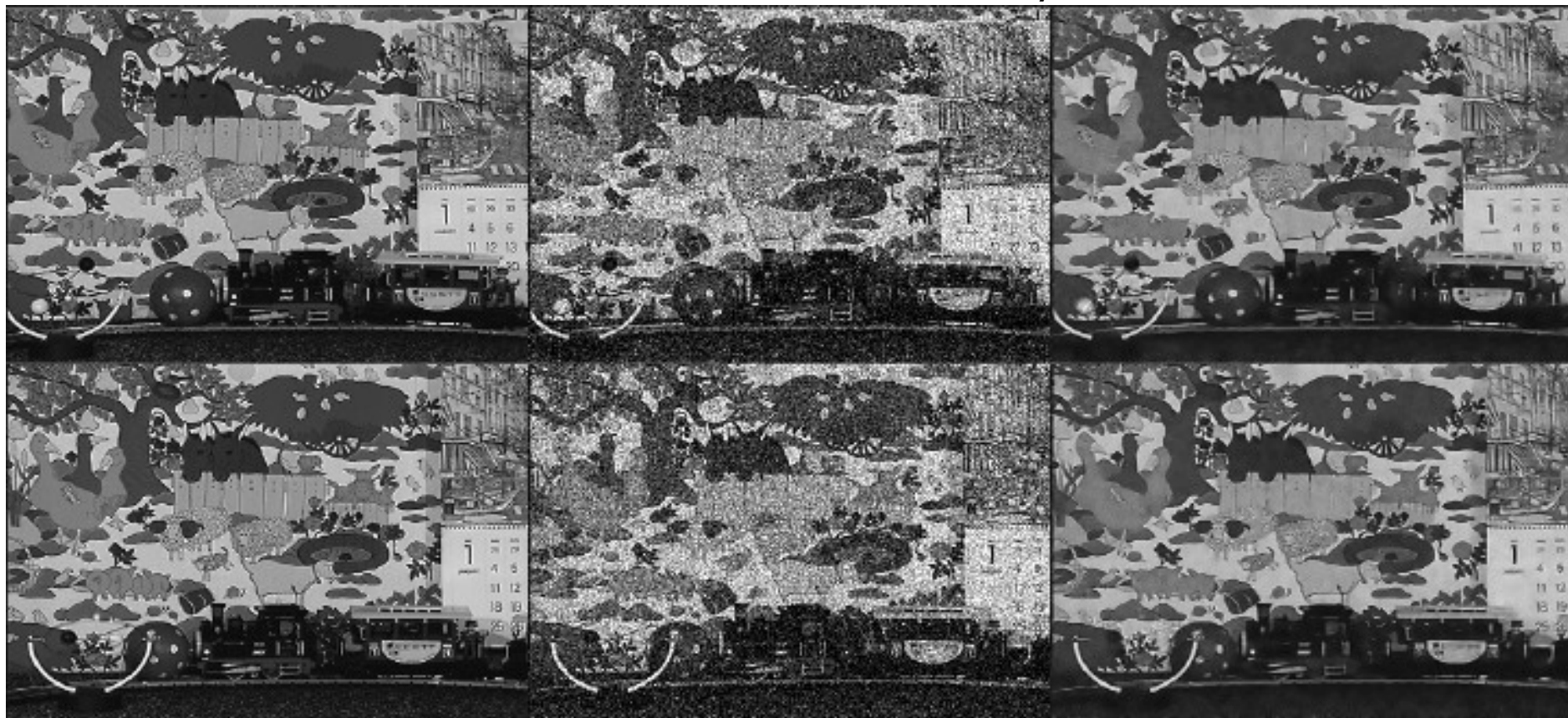


# 3D K-SVD: результаты (5)

Source

Noisy

Filtered



[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)





# 3D K-SVD: ВЫВОДЫ

---

## Достоинства:

- Успешное подавление сильных шумов
- Высокое субъективное качество изображения
- Баланс скорость/качество (число итераций)

## Недостатки:

- Низкая скорость работы
- Наличие словаря не допускает аппаратной реализации

[«Motion-Estimation-Free Algorithm for Video Denoising» \(SPIE 2007\)](#)



# Содержание

---

- Введение
  - Источники и виды шумов
  - Цели шумоподавления
- Алгоритмы
  - Общие сведения
  - STVF
  - 3D K-SVD
  - **MHMCf**
  - BM3D-SARCA
- Заключение



# МНМСФ: введение

---

Будем использовать только временную избыточность.

Для каждой точки найдем несколько соответствующих ей точек на предыдущих кадрах (назовем их гипотезами компенсации).

Усредним значения гипотез между собой.

[«A multihypothesis motion-compensated temporal filter for video denoising» \(ICIP 2006\)](#)



# МНМСФ: модель сигнала

---

Будем использовать следующую модель сигнала:

- $S_0$  – текущий пиксель изображения
- $Z_j$  – ошибка, возникшая из-за смещения объектов и неверной компенсации движения
- $C_j$  – гипотезы компенсации для  $S_0$
- $N_j$  – шум, который мы хотим подавить
- $C_m' = C_m + N_m$
- $S_0' = S_0 + N_0$

[«A multihypothesis motion-compensated temporal filter for video denoising» \(ICIP 2006\)](#)

# МНМСФ: фильтрация (1)

Оценим текущий кадр:

$$\vec{Y} = \vec{H} s_0 + \vec{Z} + \vec{N}, \text{ where } \vec{Y} = [s_0', c_1', \dots, c_n']^T, \vec{H} = [1, 1 \dots 1]^T$$

$$\hat{s}_0 = \vec{B} \vec{Y} + d, \text{ where } d = E Z_n$$

Постараемся минимизировать разницу между  $s_0$  и  $\hat{s}_0$ , основываясь на методе наименьших квадратов.

Оценим вектор  $\vec{B}$ ,

$$\vec{B} = (\vec{H}^T (\text{Cov}(\vec{Z} + \vec{N}))^{-1} \vec{H})^{-1} \vec{H}^T (\text{Cov}(\vec{Z} + \vec{N}))^{-1}$$

И значение  $d$ :

$$d = -\vec{B} \vec{Z}$$

[«A multihypothesis motion-compensated temporal filter for video denoising» \(ICIP 2006\)](#)



# МНМСФ: фильтрация (2)

$$\vec{B} = [b_0, b_1, \dots, b_N]$$

$$b_0 = \frac{\sigma_0^{-2}}{\sum_{k=1}^N \sigma_{p(k)}^{-2} + \sigma_0^{-2}}$$

$$b_\mu = \frac{\sigma_{p(\mu)}^{-2}}{\sum_{k=1}^N \sigma_{p(k)}^{-2} + \sigma_0^{-2}}$$

where  $\sigma_{p(\mu)}^2$  is the variance of  $(z_\mu + n_\mu)$

$$\sigma_{p(\mu)}^2 = \sigma_{z(\mu)}^2 + \sigma_\mu^2$$

[«A multihypothesis motion-compensated temporal filter for video denoising» \(ICIP 2006\)](#)



# МНМСФ: оценка параметров

---

Необходимо оценить значения  $\sigma_0, \sigma_{p(\mu)}, z_\mu$  :

$\sigma_0^2$  – среднее значение 10 минимальных дисперсий  
блоков 16x16

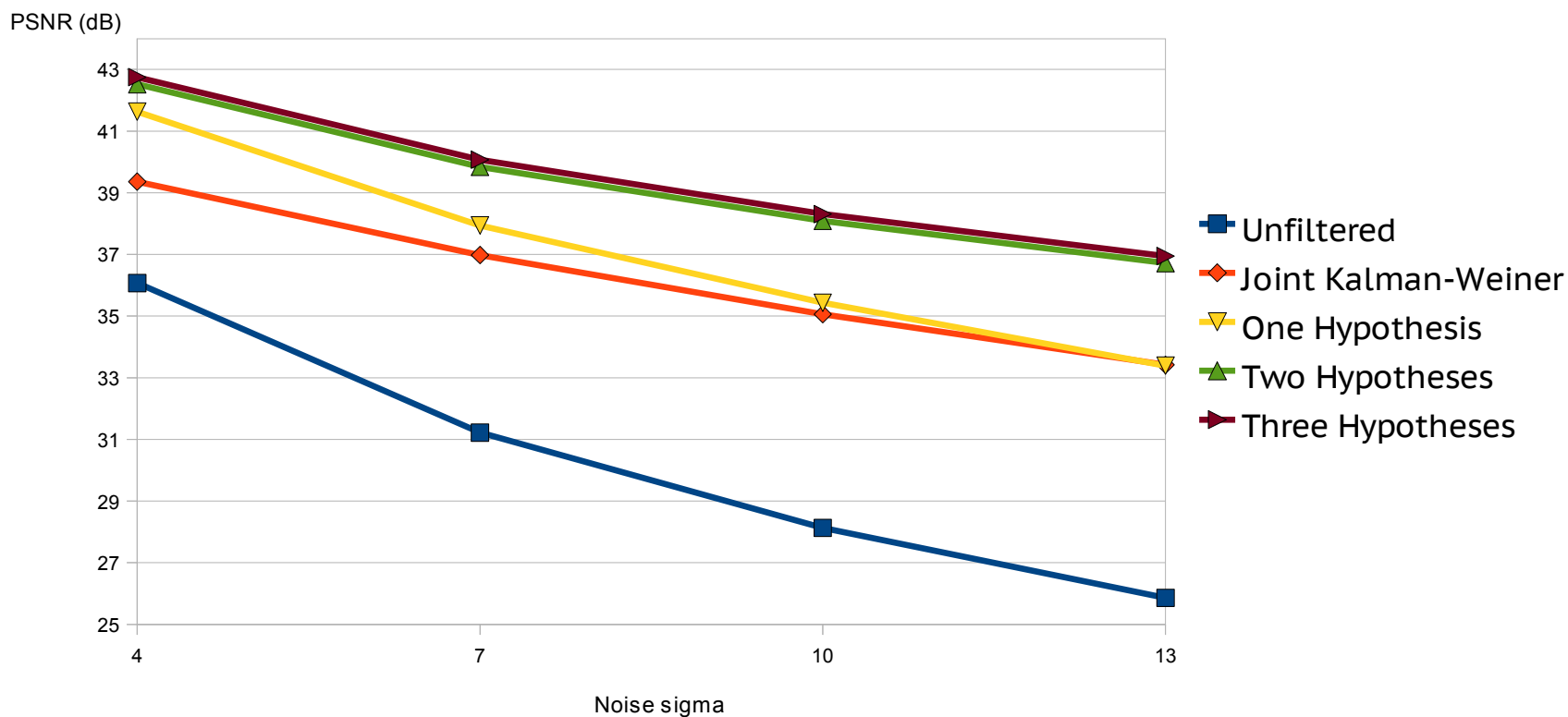
$z_\mu, \sigma_{p(\mu)}$ : возьмем кадр из шумов компенсации  
движения. Найдем блок, который соответствует  $z_\mu$ .  
Посчитаем среднее блока ( $z_\mu$ ) и дисперсию  $d = \sigma_{p(\mu)}^2 + \sigma_0^2$

Подставив значения, получим кадр, очищенный от шума.

[«A multihypothesis motion-compensated temporal filter for video denoising» \(ICIP 2006\)](#)

# MHMSF: результаты (1)

Average PSNR (db) for Akiyo sequence



[«A multihypothesis motion-compensated temporal filter for video denoising» \(ICIP 2006\)](#)





# МНМСФ: результаты (2)



Original



Noisy



# МНМСФ: результаты (3)



One Hypothesis



Two Hypotheses



# МНМСФ: ВЫВОДЫ

---

Достоинства:

- Высокая скорость
- Балансирование между скоростью и качеством

Недостатки:

- Требование хорошей компенсации движения
- Необходимость точной оценки параметров

[«A multihypothesis motion-compensated temporal filter for video denoising» \(ICIP 2006\)](#)

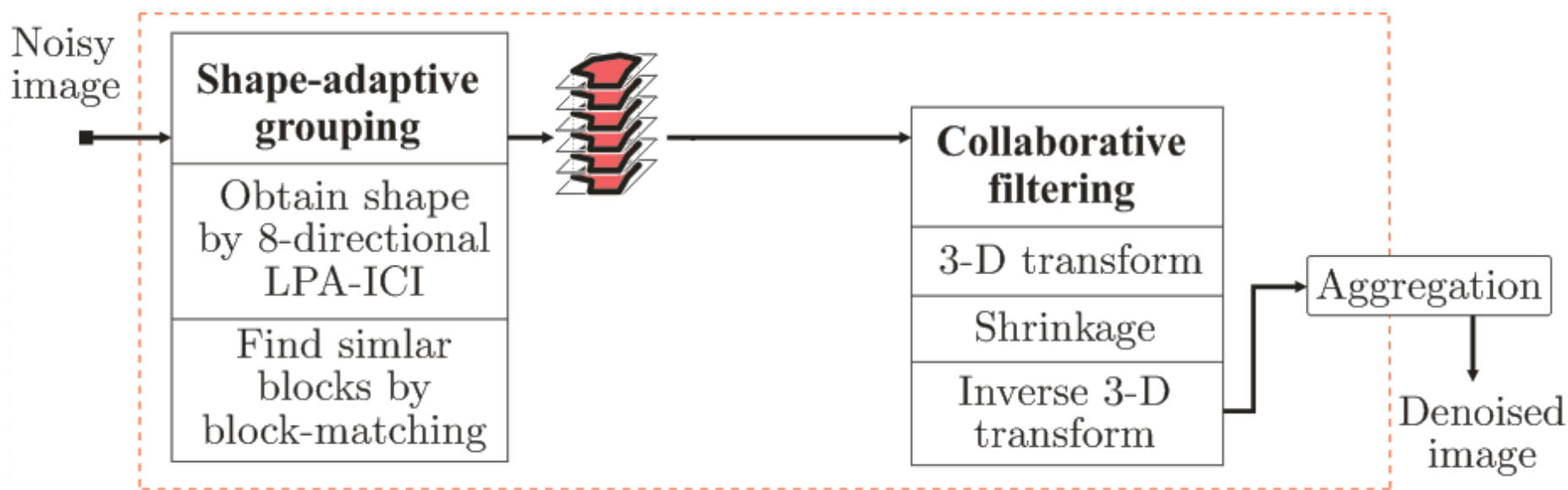


# Содержание

---

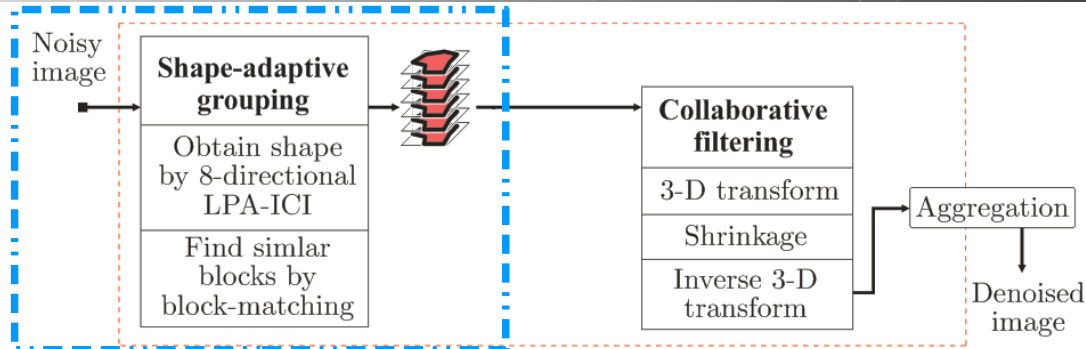
- Введение
  - Источники и виды шумов
  - Цели шумоподавления
- Алгоритмы
  - Общие сведения
  - STVF
  - 3D K-SVD
  - MMCF
  - **BM3D-SARCA**
- Заключение

# BM3D: схема алгоритма (1)



[«BM3D Image Denoising with Shape-Adaptive Principal Component Analysis» \(2009\)](#)

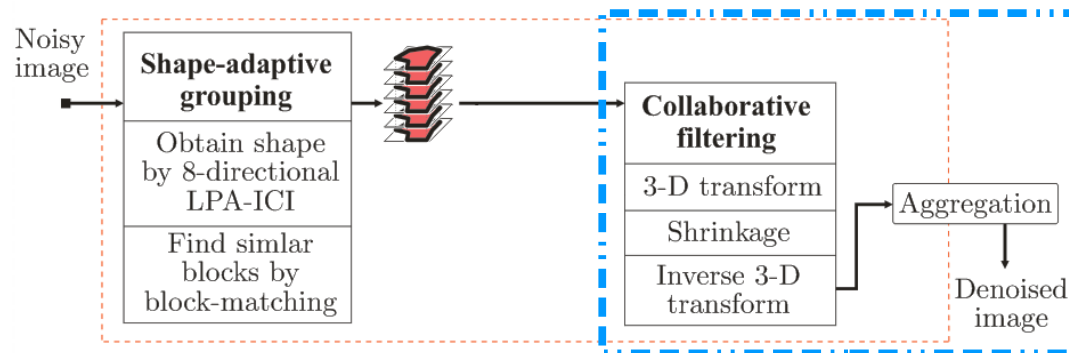
# BM3D: схема алгоритма (2)



- Возьмем пиксель, выделим его окружение (всего  $N_{el}$  пикселей)
- Найдем блоки, схожие со ссылочным (всего  $N_{br}$ )
- Сформируем трехмерный массив

[«BM3D Image Denoising with Shape-Adaptive Principal Component Analysis» \(2009\)](#)

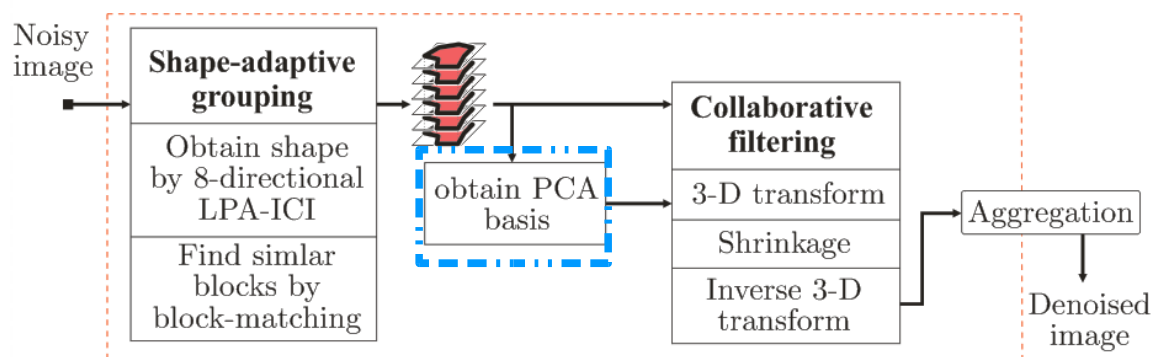
# BM3D: схема алгоритма (3)



- Применим 1D-преобразование (декомпозицию Хаара) перпендикулярно плоскости блоков
- Выполним сложение слоев (фильтр Виннера)
- Выполним обратное преобразование
- Запишем полученное изображение в результат

[«BM3D Image Denoising with Shape-Adaptive Principal Component Analysis» \(2009\)](#)

# SAPCA: расширение BM3D



После поиска блоков:

- $\frac{N_{br}}{N_{el}} \leq \tau$  : выполним обычную фильтрацию
- $\frac{N_{br}}{N_{el}} > \tau$  : выполним Space-Adaptive фильтрацию

[«BM3D Image Denoising with Shape-Adaptive Principal Component Analysis» \(2009\)](#)





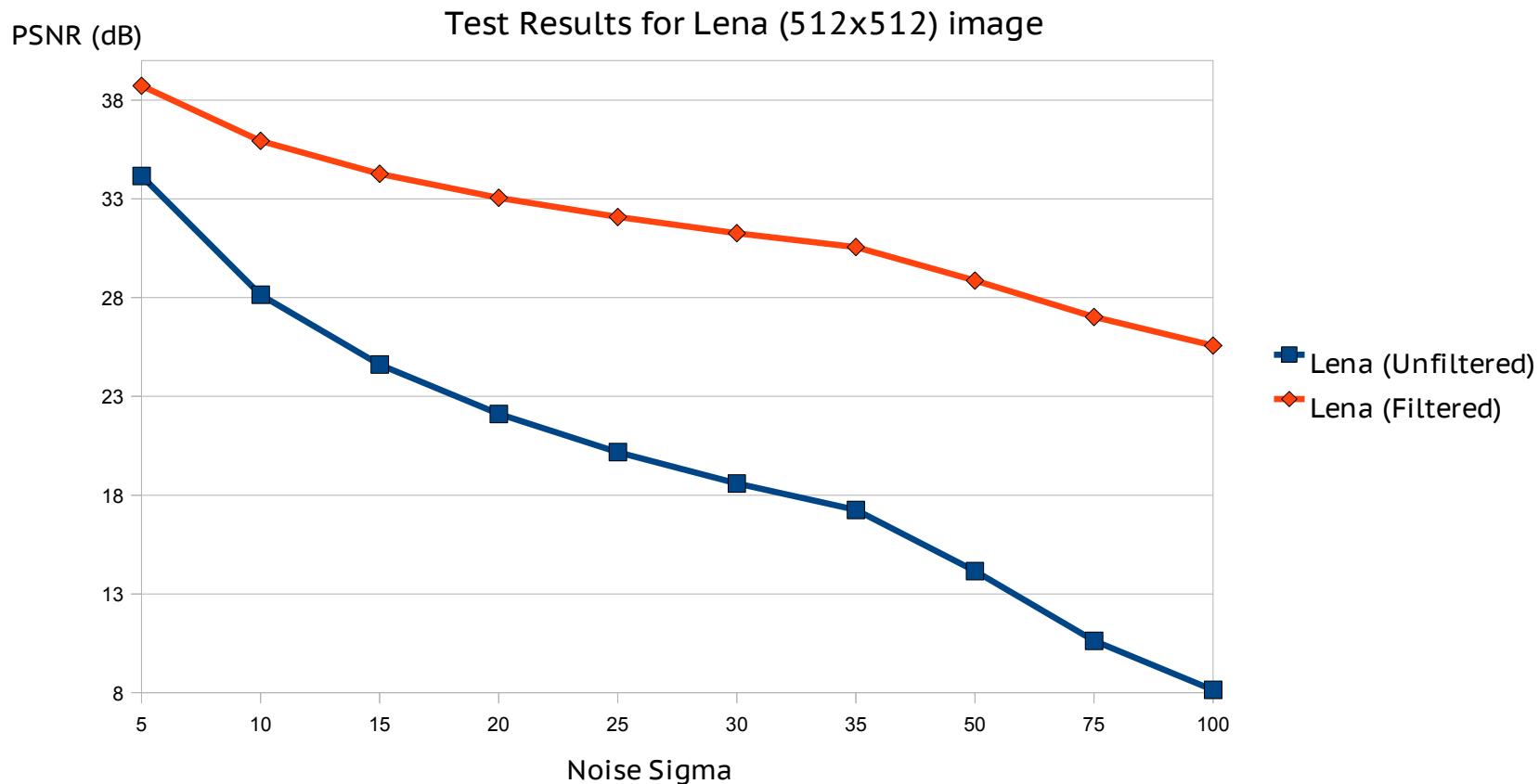
# SAPCA: схема алгоритма

- На входе –  $N_{br}$  наборов пикселей
- Представим каждый набор как столбец ( $v_i$ ) из  $N_{el}$  пикселей
- $C = [v_1, v_2, \dots, v_{N_{br}}][v_1, v_2, \dots, v_{N_{br}}]^T$  – матрица вторых моментов
- $U^T C U = S = \text{diag}(s_1, s_2, \dots, s_{N_{el}})$ , где  $U$  – ортонормированная матрица
- $s_i$  – отсортированные собственные значения матрицы
- Для декомпозиции возьмем первые  $N_{trim}$  столбцов  $U$

$$N_{trim} = |\{s_i | s_i > \lambda \sigma^2\}|$$

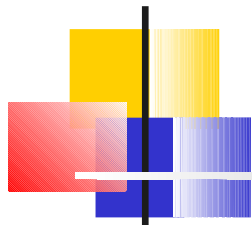
[«BM3D Image Denoising with Shape-Adaptive Principal Component Analysis» \(2009\)](#)

# BM3D-SARCA: результаты (1)



[«BM3D Image Denoising with Shape-Adaptive Principal Component Analysis» \(2009\)](#)

# BM3D-SARCA: результаты (2)



Source





# BM3D-SARCA: результаты (3)



Noisy ( $\sigma = 100$ )





# BM3D-SARCA: результаты (4)

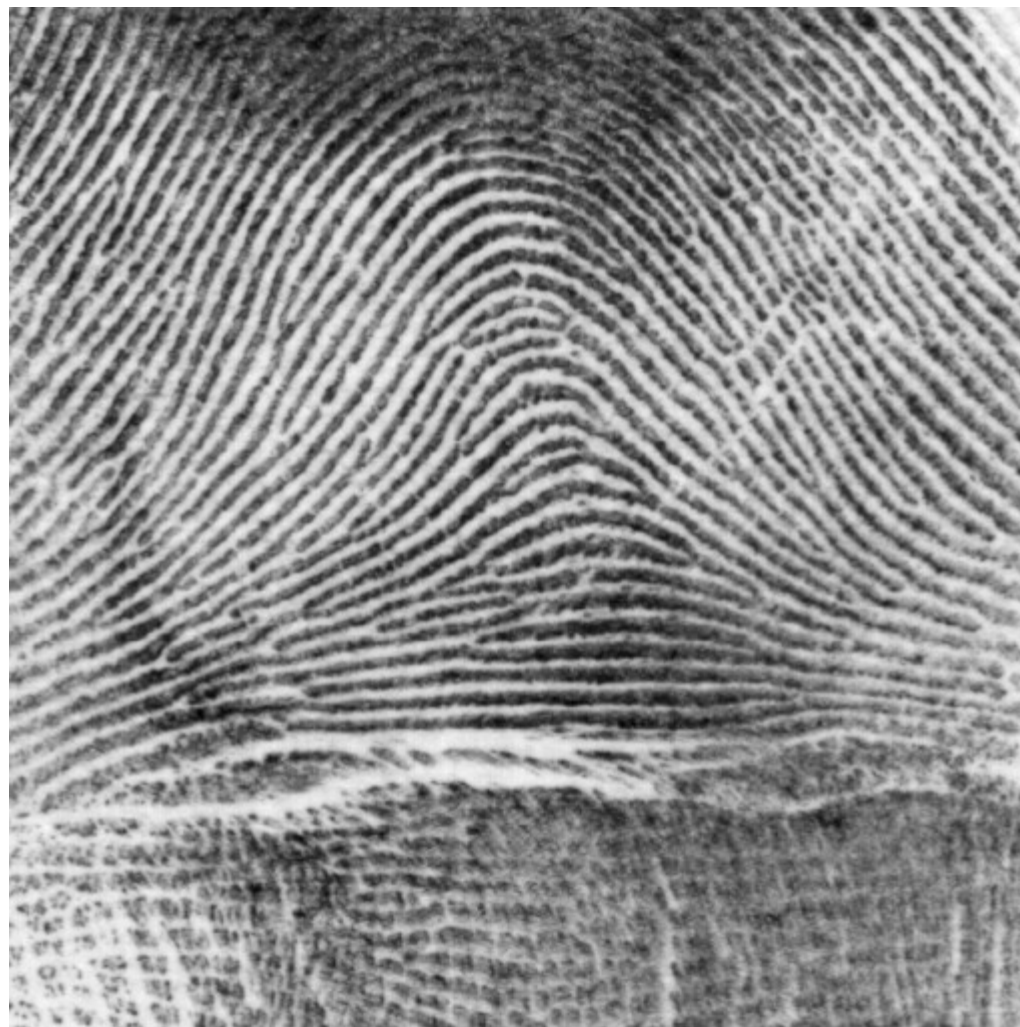


Filtered



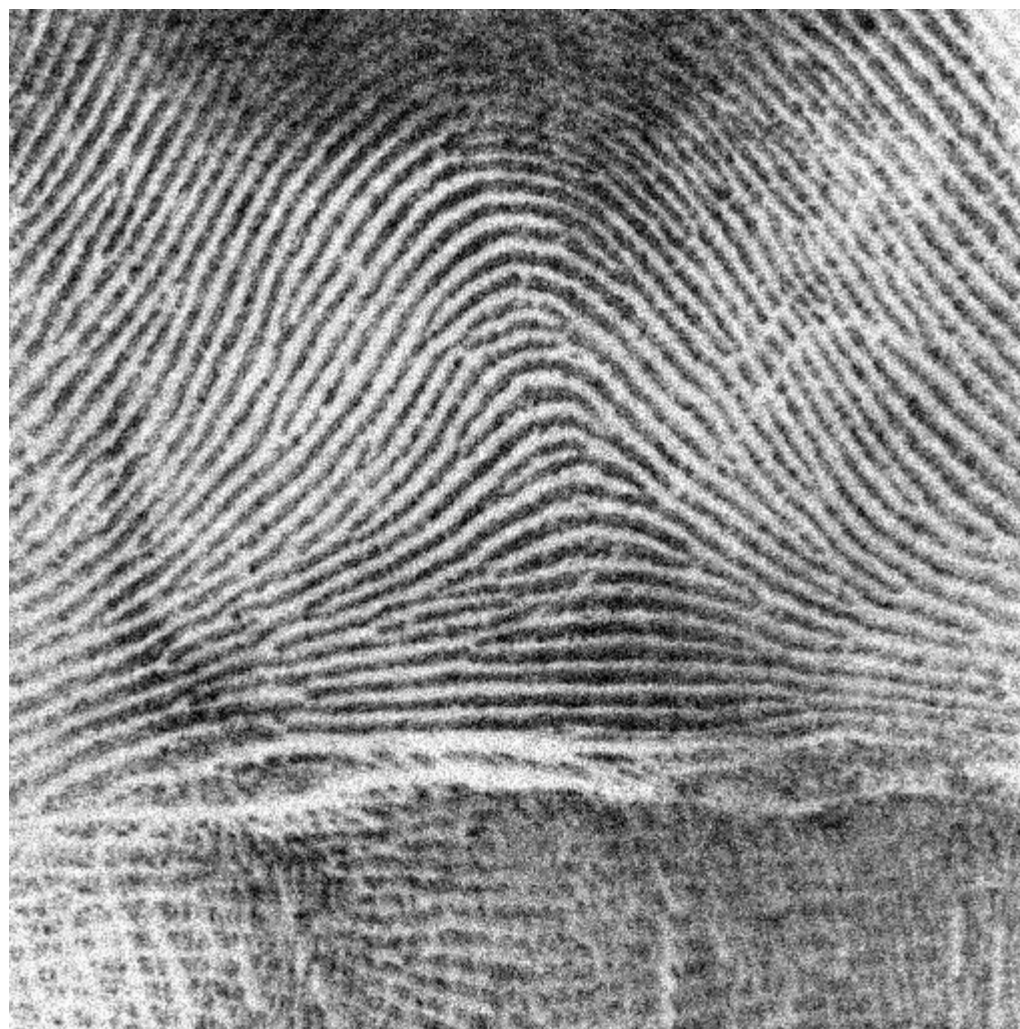
# BM3D-SARCSA: результаты (5)

Source





# BM3D-SARCSA: результаты (6)

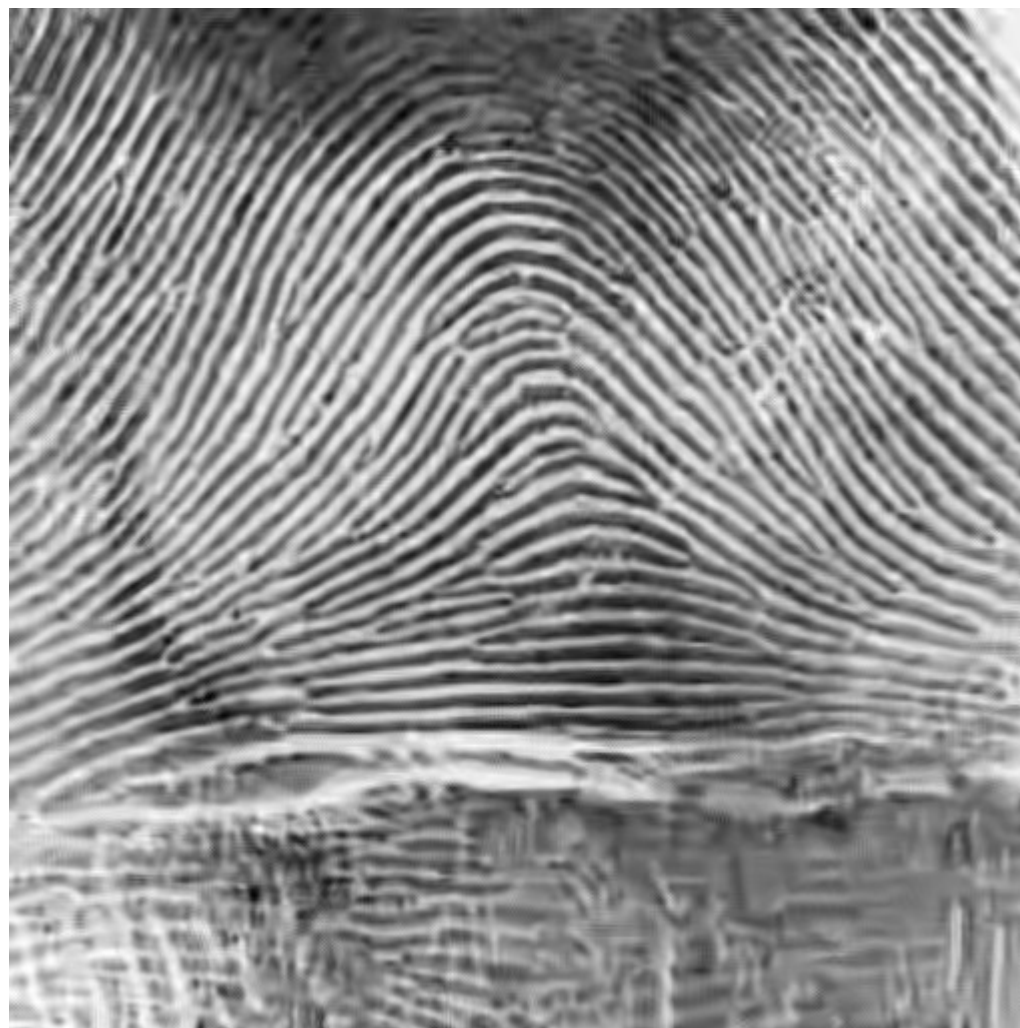


Noisy ( $\sigma = 50$ )





# BM3D-SARCA: результаты (7)



Filtered





# BM3D-SARSA: ВЫВОДЫ

---

## Достоинства:

- Высокий показатель PSNR на всевозможных уровнях шума
- Возможность итерирования для повышения качества

## Недостатки:

- Скорость (Pentium 2GHz, Matlab, 4 минуты на изображение 256x256)
- Артефакты при сильных шумах

[«BM3D Image Denoising with Shape-Adaptive Principal Component Analysis» \(2009\)](#)



# Содержание

---

- Введение
  - Источники и виды шумов
  - Цели шумоподавления
- Алгоритмы
  - Общие сведения
  - STVF
  - 3D K-SVD
  - MMCF
  - BM3D-SARCA
- **Заключение**



# Выводы

---

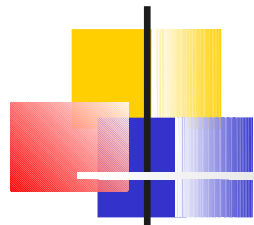
- Существует много различных подходов к поставленной задаче
- Большинство методов допускают баланс между качеством и скоростью
- Самым результативным (объективно) является последний метод
- Самым результативным (субъективно) является метод со словарем



# Список литературы

---

1. Liwei Guo, Oscar C. Au, Mengyao Ma, Zhiqin Liang, Carman K.M. Yuk «A multihypothesis motion-compensated temporal filter for video denoising» (*ICIP 2006*)
2. «YUVsoft Video Denoiser Comparison» (2007)
3. Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian «BM3D Image Denoising with Shape-Adaptive Principal Component Analysis» (2009)
4. Matan Protter, Michael Elad «Sparse and Redundant Representations and Motion-Estimation-Free Algorithm for Video Denoising» (*SPIE 2007*)
5. Tai-Wai Chan, Oscar C. Au, Tak-Song Chong, Wing-San Chau «A novel content-adaptive video denoising filter» (*ICASSP 2005*)



# Вопросы

---

