

Деинтерлейсинг

Дмитрий Ватолин,
Александр Плошкин
Video Group
CS MSU Graphics&Media Lab

Содержание

- Мир аналогового телевидения
 - Форматы вещания NTSC, SECAM, PAL
 - Виды развёрток
- Деинтерлейсинг
- Эвристические алгоритмы деинтерлейсинга
 - Temporal
 - Spatial
 - Motion Adaptive
 - Motion-Compensation-Based
- Нейросетевые алгоритмы
- Задание

Аналоговое телевидение

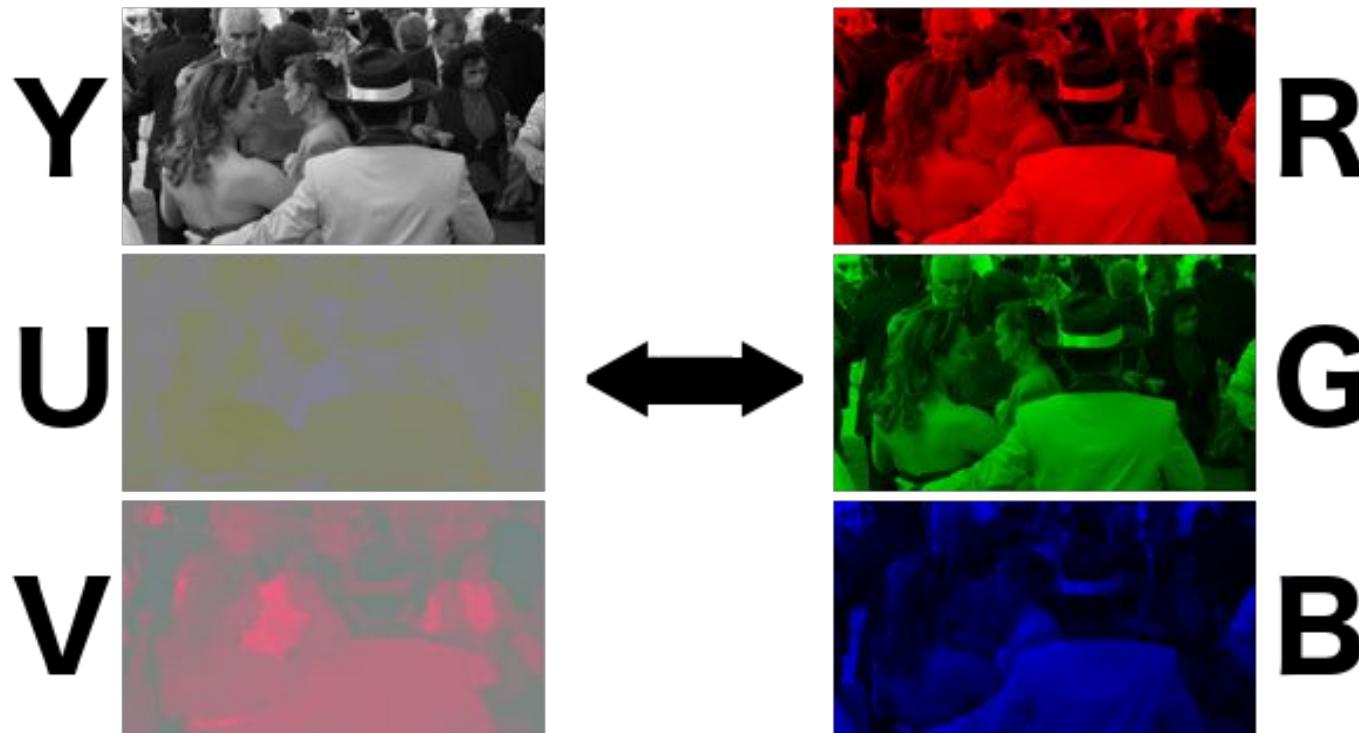


- Вещание начато в 1953 году
- Наибольшее распространение в странах Америки
- Совместимость с ЧБ:

$$Y = 0.299R + 0.587G + 0.114B$$

- Передача компонент Y , U , V , причём на компоненту яркости (Y) приходится большая ширина канала
- Система чувствительна к амплитудно-фазовым искажениям, которые влияют на цвет
- “*No True Skin Color*”

Цветовая схема YUV



SECAM и PAL

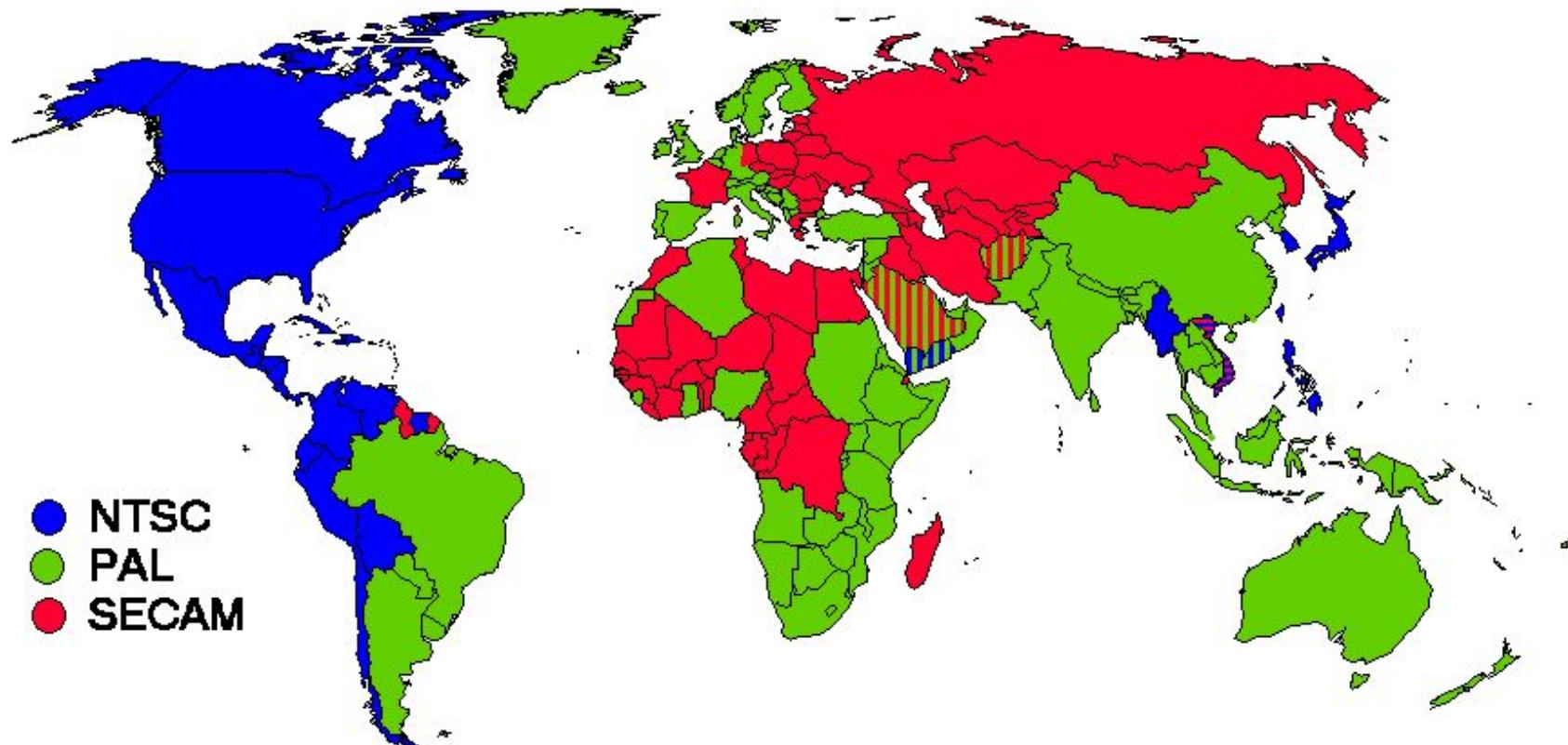
SECAM (*Séquentiel couleur à mémoire*), 1967 г.:

- Советско-французский формат цветного телевещания
- На тот момент — оптимум между качеством картинки и ценой декодера

PAL (*Phase Altering Line*), 1966 г.:

- Европейский формат цветного телевещания
- Наиболее качественная картинка и самый дорогой декодер
- В последствии — популярнейший формат цветного ТВ

Форматы вещания по регионам



Colour TV Systems of the World 1990

Содержание

- Мир аналогового телевидения
 - Форматы вещания NTSC, SECAM, PAL
 - **Виды развёрток**
- Деинтерлейсинг
- Эвристические алгоритмы деинтерлейсинга
 - Temporal
 - Spatial
 - Motion Adaptive
 - Motion-Compensation-Based
- Нейросетевые алгоритмы
- Задание

Виды развёрток

Прогрессивная (Progressive):

- Обычное покадровое отображение видео
- Имеет ту же частоту, что и система вещания

Чересстрочная (Interlaced):

- Автор — Л.С. Термен (1927 г.)
- Каждый кадр разделён на 2 полукадра (поля) — чётные/нечётные строки кадра
- В каждый момент времени отображается 1 полукадр
- Имеет удвоенную частоту

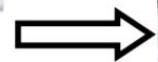
Чересстрочная развёртка

Общая схема

Поле #5



Поле #6



Кадр #3



Чересстрочная развёртка

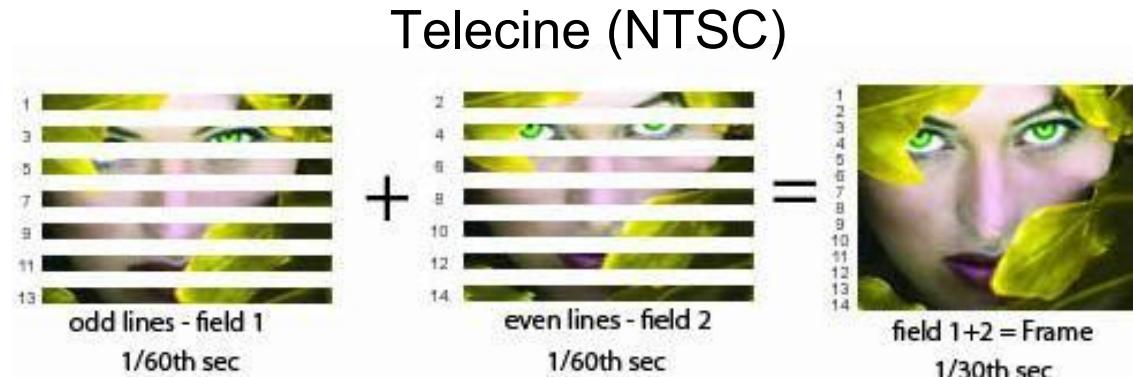
Пример



Чересстрочная развёртка

Способы создания

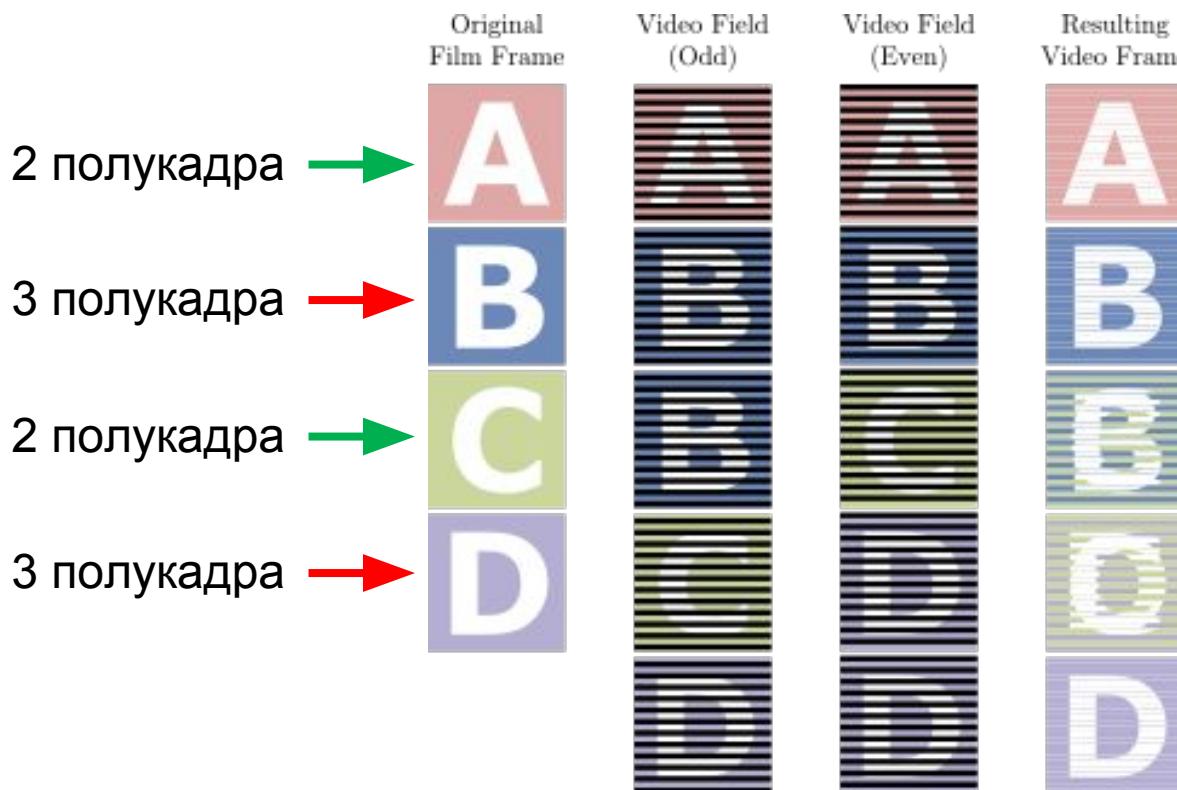
1. Съёмка сразу в interlaced-формате (50 или 60 полукадров в секунду)
 - Оптимальное качество
 - Невозможность конвертации в progressive
2. Метод Telecine: обычная съёмка в 24 fps и преобразование в нужный interlaced-формат



Telecine

NTSC 2:3

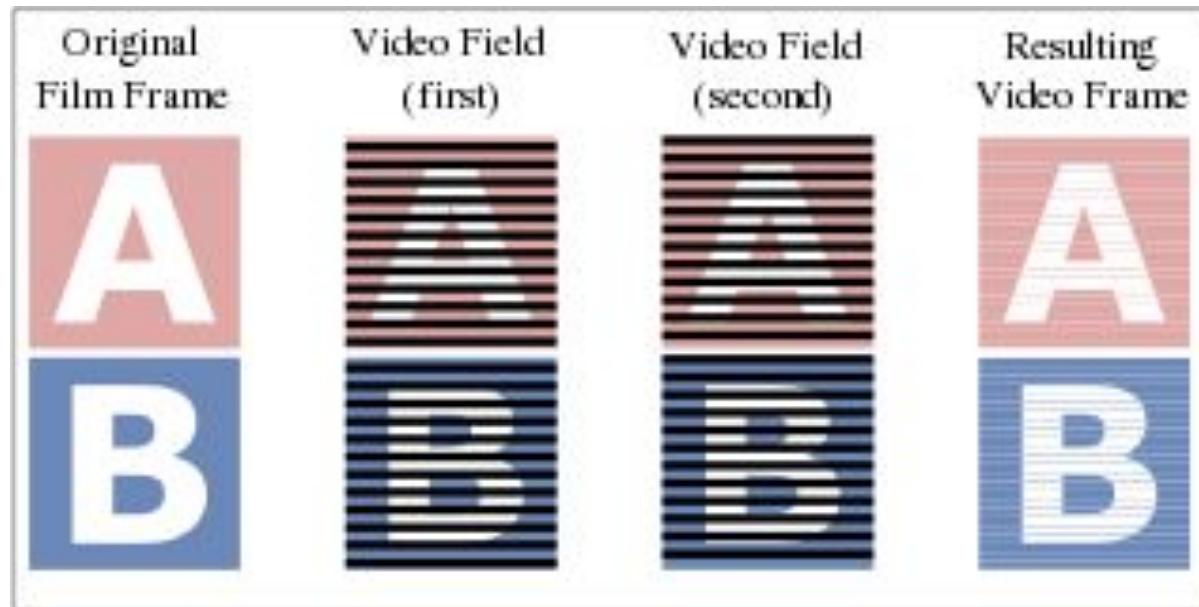
- NTSC 2:3 — преобразование 24 fps в 60 fps; для этого происходит чередование 2-3 полукадра на кадр



Telecine

PAL 2:2

- NTSC 2:3 — преобразование 24 fps в 60 fps; для этого происходит чередование 2/3 полукадра на кадр
- PAL 2:2 — деление каждого кадра на 2 полукадра и ускорение потока на 4%; звук искажается



Telecine

PAL 2:2:2:2:2:2:2:2:2:2:3



- NTSC 2:3 — преобразование 24 fps в 60 fps; для этого происходит чередование 2/3 полукадра на кадр
- PAL 2:2 — деление каждого кадра на 2 полукадра и ускорение потока на 4%; звук искажается
- NTSC 2:2 — аналогично PAL 2:2 без ускорения, если частота источника 30 fps
- PAL 2:2:2:2:2:2:2:2:2:2:3 — каждый 12-й кадр делится на 3 полукадра

Содержание

- Мир аналогового телевидения
- **Деинтерлейсинг**
- Эвристические алгоритмы деинтерлейсинга
 - Temporal
 - Spatial
 - Motion Adaptive
 - Motion-Compensation-Based
- Нейросетевые алгоритмы
- Задание

Деинтерлейсинг

Деинтерлейсинг (Deinterlacing) — процесс создания кадров из полукадров чересстрочного видео для последующего вывода на экран с прогрессивной развёрткой



Деинтерлейсинг

В зависимости от метода интерлейсинга

1. Telecine

Если известен метод конвертации из 24 fps в interlaced-формат, то можно без потерь восстановить исходное видео

2. Съёмка в формате вещания

В этом случае полукадры идут с интервалом в 1/50 с (1/60 с), поэтому задача состоит в интерполяции недостающих строк в каждом кадре

Постановка задачи

Для каждого полукадра (поля) из видео в чересстрочном формате восстановить пиксели в недостающих строках

$$p_k(x, y) = \begin{cases} p_k(\lfloor x/2 \rfloor, \lfloor y/2 \rfloor), & \text{если } y \equiv_2 k \\ \tilde{p}_k(x, y), & \text{иначе} \end{cases}$$

- x, y — координаты пикселя,
- k — номер кадра,
- p — пиксель в исходном видео,
- $\tilde{p}(\cdot)$ — интерполирующая функция,
- p — пиксель в итоговом видео

Классификация алгоритмов дейнтерлейсинга



Пространственные (Spatial)

- Line Repetition
- Line Average (LA, Bob)
- Edge LA (ELA)
- EEDI2

Временные (Temporal)

Не использующие ME

- Field Repetition
- Inter-frame LA
- Yadif/Yadifmod
- TDeint

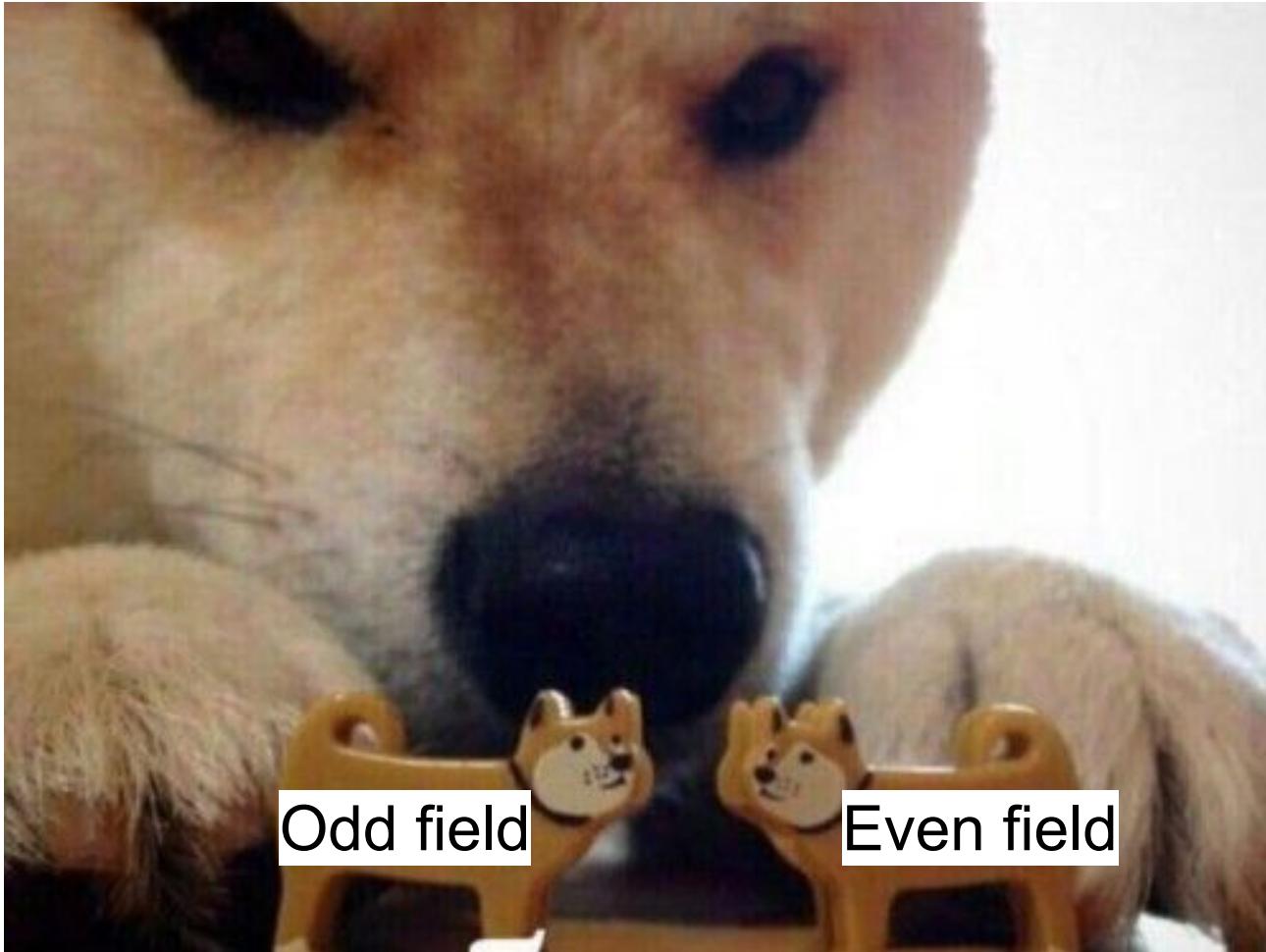
На основе ME

- MCBob

Содержание

- Мир аналогового телевидения
- Деинтерлейсинг
- Эвристические алгоритмы деинтерлейсинга
 - Spatial, Temporal, Spatio-Temporal, Edge-based
 - Motion Adaptive
 - Motion-Compensation-Based
- Нейросетевые алгоритмы
- Задание

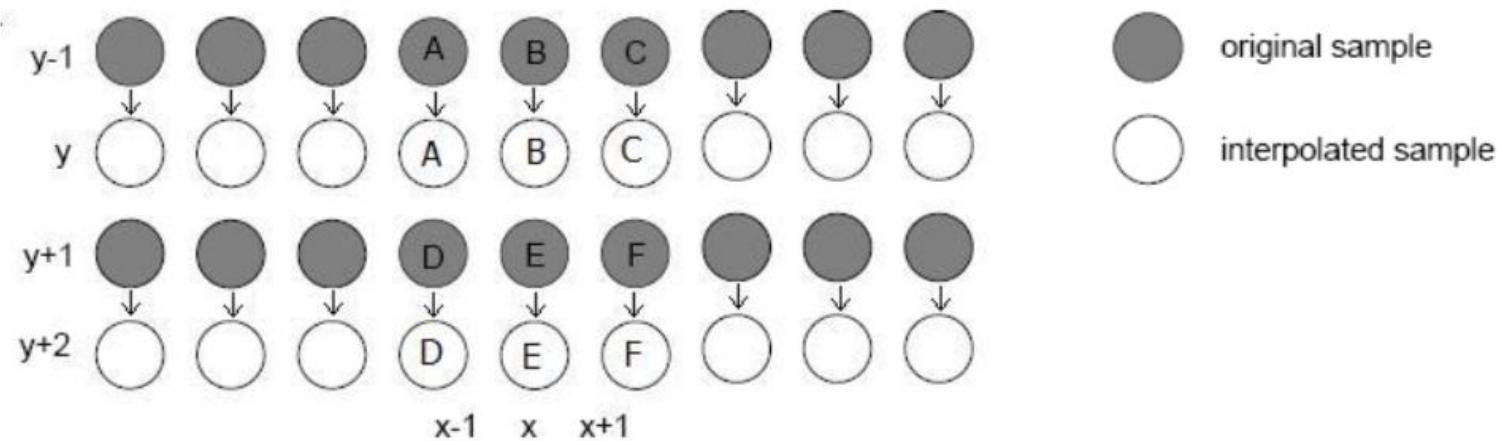
Spatial algorithms



Spatial Line Repetition

Line Repetition (LR)

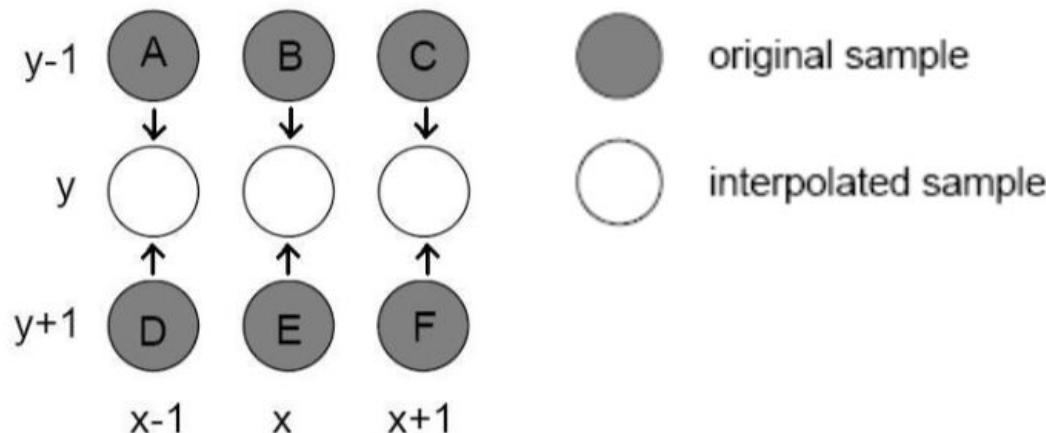
$$p_0(x, y, k) = \begin{cases} p(x, y, k), & (y+k)\%2 = 0, \\ p(x, y-1, k), & \text{иначе} \end{cases}$$



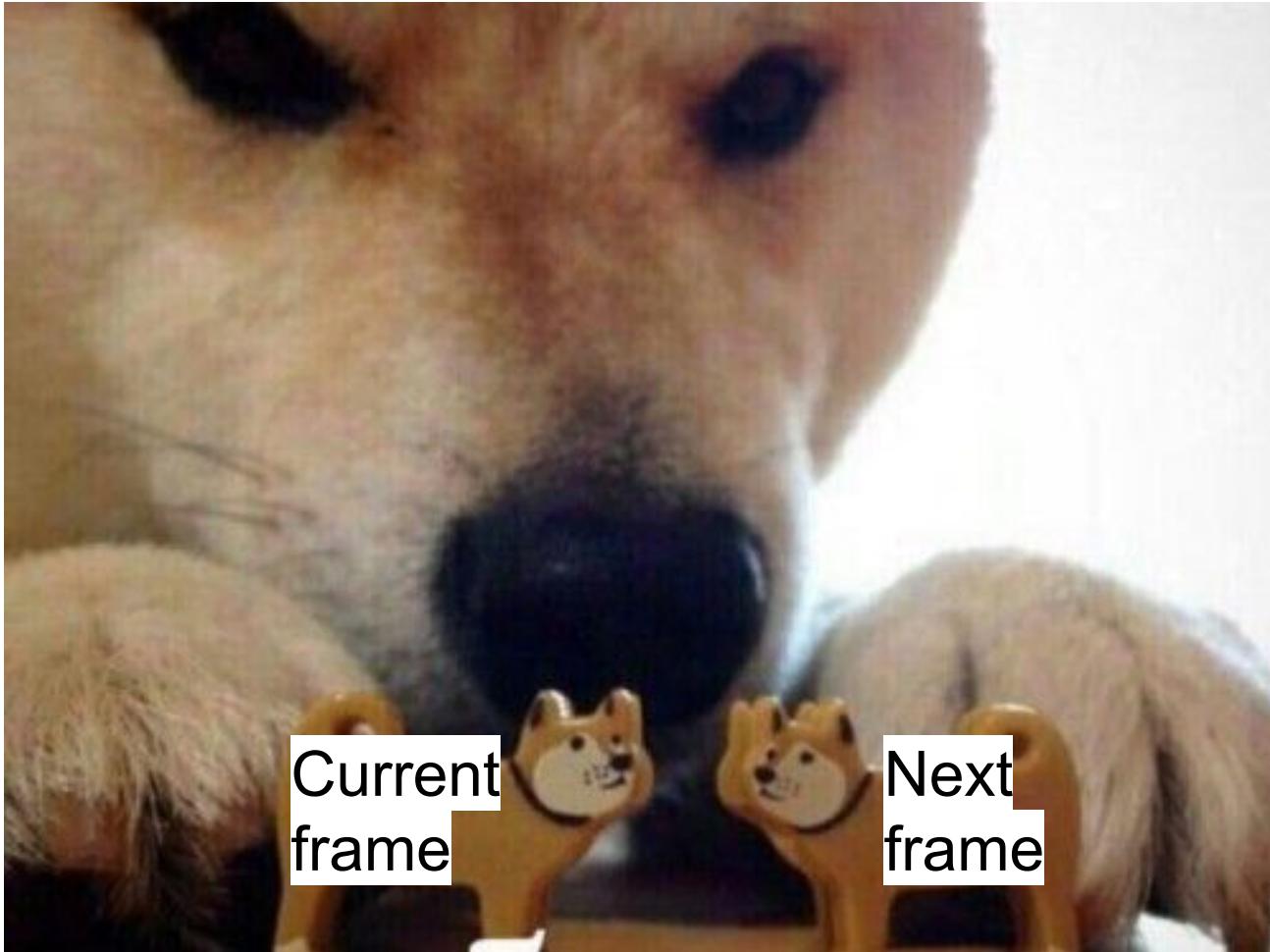
Spatial Bob

Line Averaging (LA) = “Bob”

$$p_0(x, y, k) = \begin{cases} p(x, y, k), & (y + k) \% 2 = 0, \\ \frac{1}{2} [p(x, y - 1, k) + p(x, y + 1, k)], & \text{иначе,} \end{cases}$$



Temporal algorithms

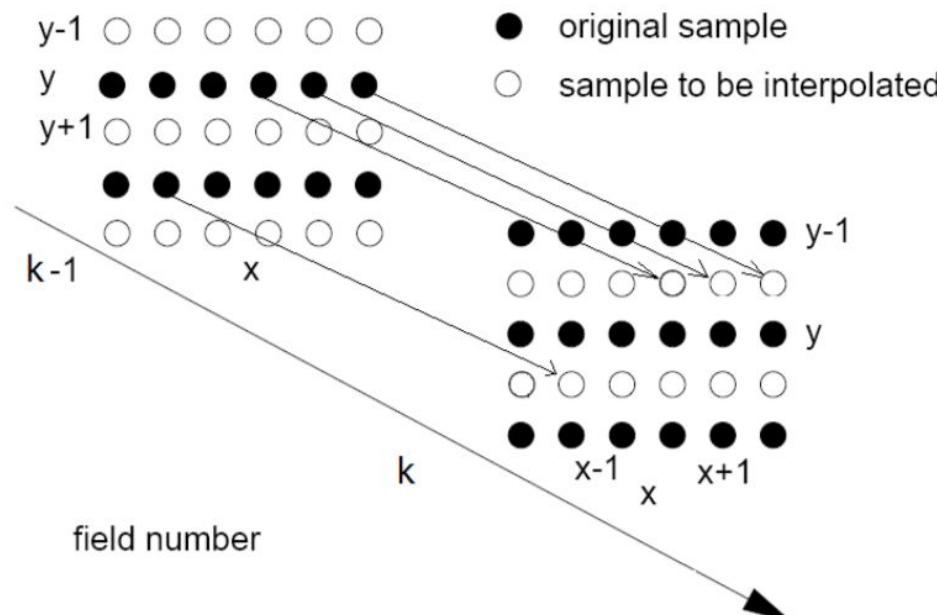


Temporal algorithms

Weave

Field Insertion (FI) = Field Repetition (FR) = “Weave”

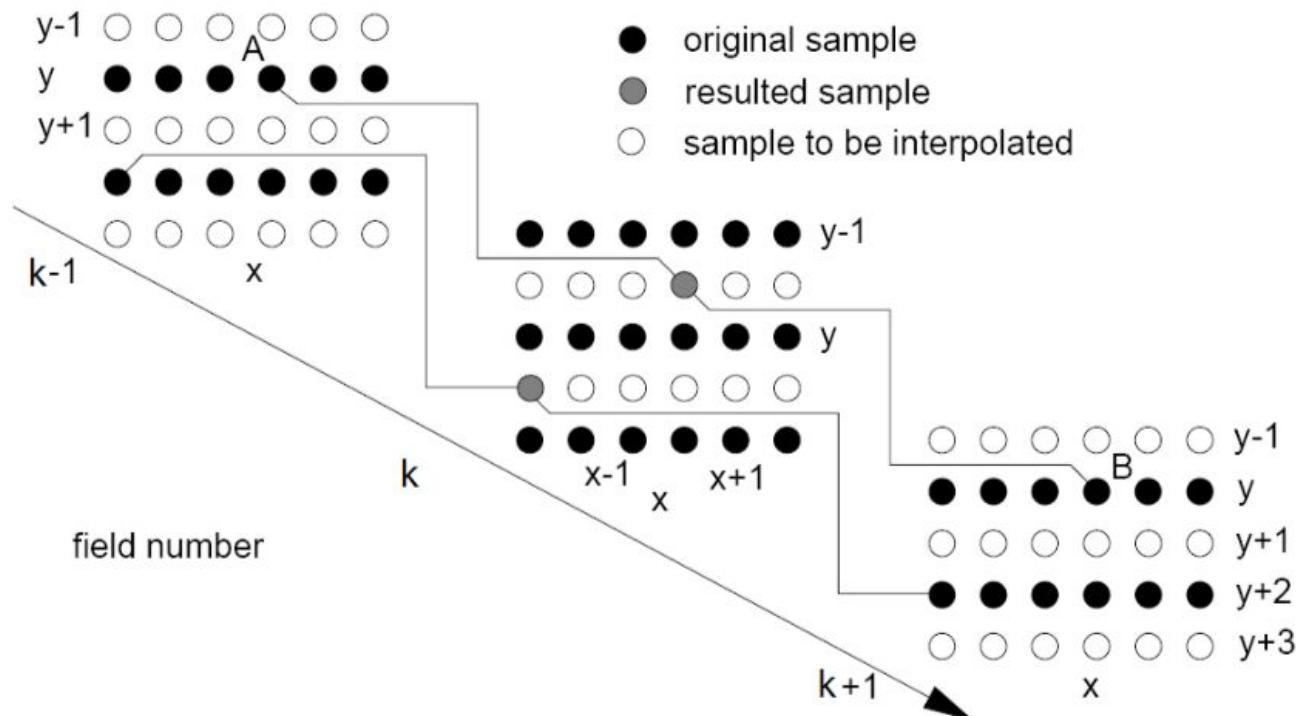
$$p_0(x, y, k) = \begin{cases} p(x, y, k), & (y + k) \% 2 = 0, \\ p(x, y, k - 1), & \text{иначе,} \end{cases}$$



Temporal algorithms

Inter-field Line Averaging

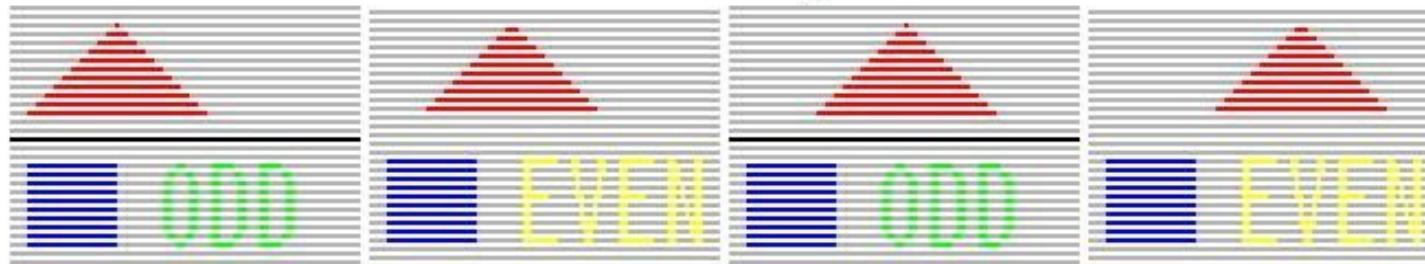
$$p_0(x, y, k) = \begin{cases} p(x, y, k), & (y+k)\%2 = 0, \\ \frac{1}{2}[p(x, y, k-1) + p(x, y, k+1)], & \text{иначе,} \end{cases}$$



Схематичное сравнение

Bob vs. Weave

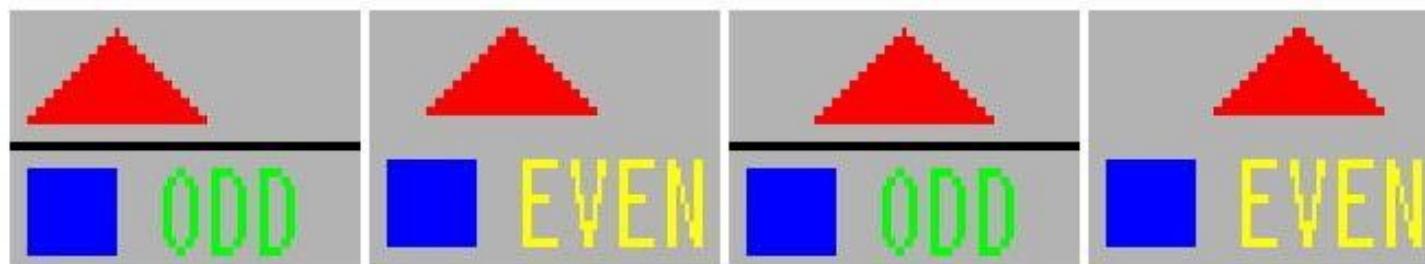
Deinterlacing



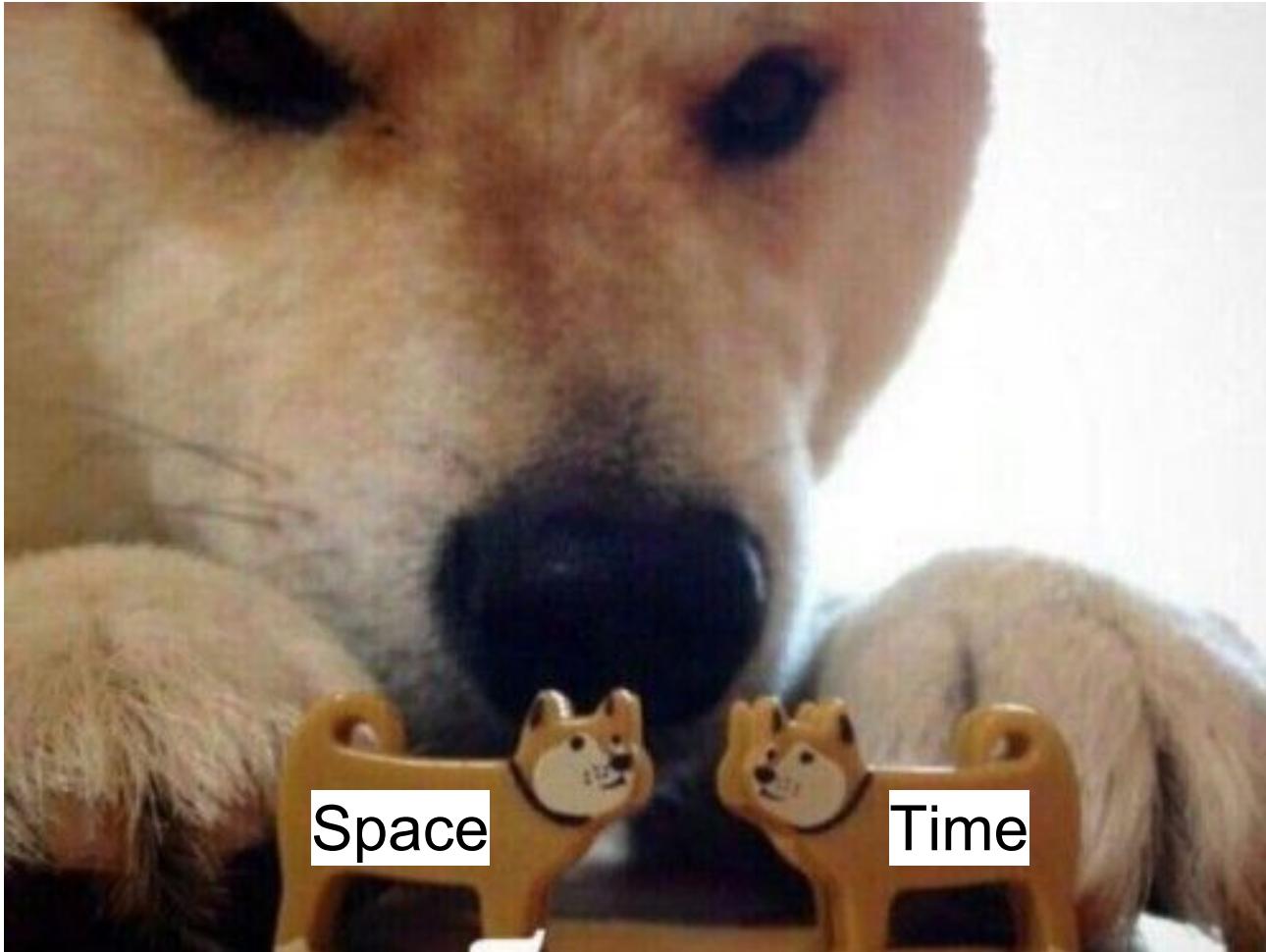
Weave



Bob



Spatio-temporal algorithms

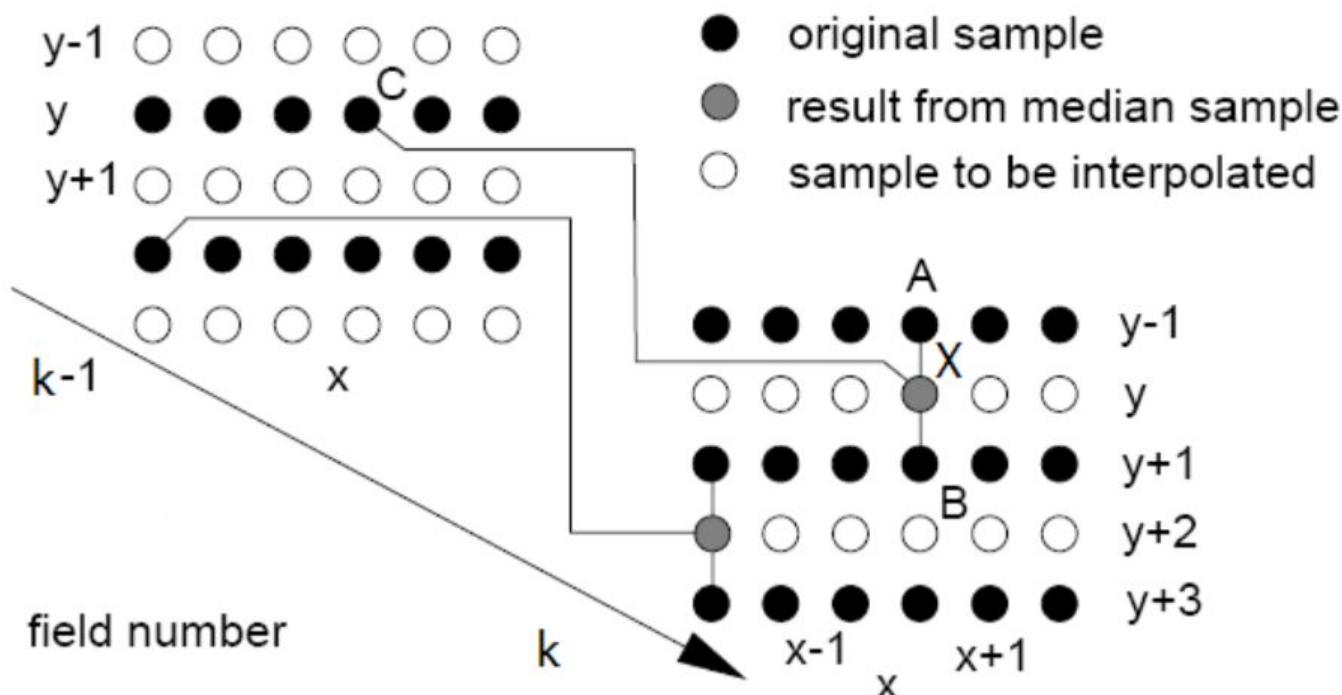


Spatio-temporal algorithms



Vertical Temporal Median Filter (VTMF)

$$p_0(X) = median(A, B, C)$$



Spatio-temporal algorithms

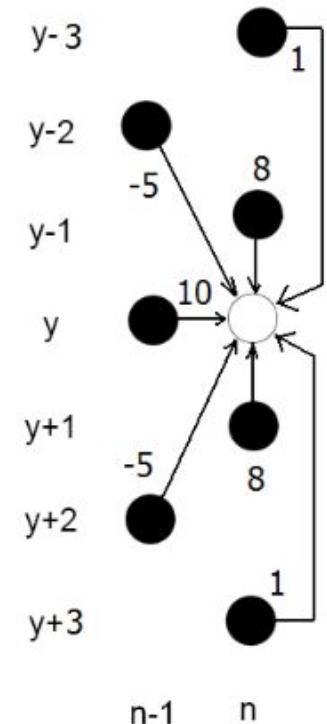
Vertical Temporal Linear Interpolation



Хитрый шаблон линейной интерполяции

$$p_0(x, y, k) = \begin{cases} p(x, y, k), & (y + k) \% 2 = 0, \\ \sum_k p(x, y + dy, k + dk) * h(dy, dk), & \text{иначе} \end{cases}$$

$$18 * h(dy, dk) = \begin{cases} 1,8,8,1, & (dy = -3, -1, 1, 3) \text{ AND } (dk = 0) \\ -5,10,5, & (dy = -2, 0, 2) \text{ AND } (dk = -1) \\ 0, & \text{иначе} \end{cases}$$



original sample



sample to be interpolated

Edge-based algorithms

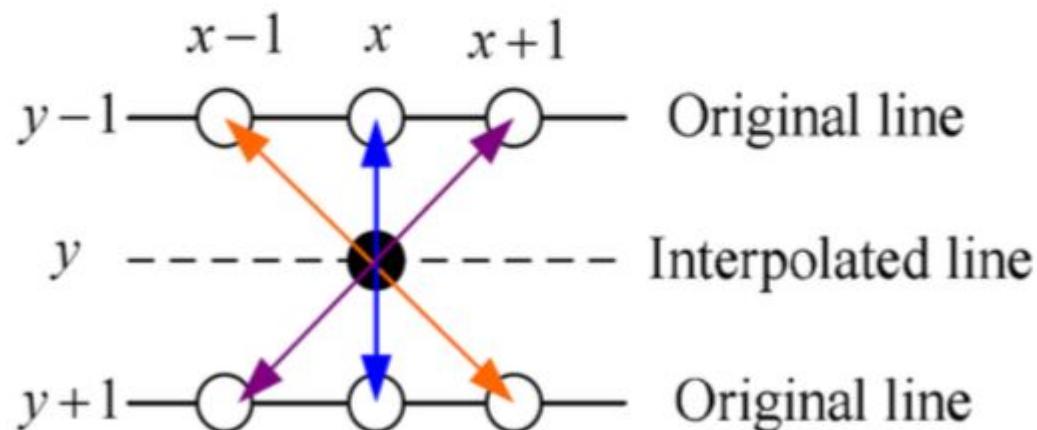


Edge-based algorithms

Edge-based Line Averaging (ELA)

EL Averaging (ELA) = EL Interpolation (ELI) =
 = Edge-based De-Interlacing (EDI)

$$f_n(x, y) = \frac{\hat{f}(x+k, y-1) + \hat{f}(x-k, y+1)}{2} \quad \hat{k} = \arg \min_{-1 \leq k \leq 1} \{D(k)\}$$

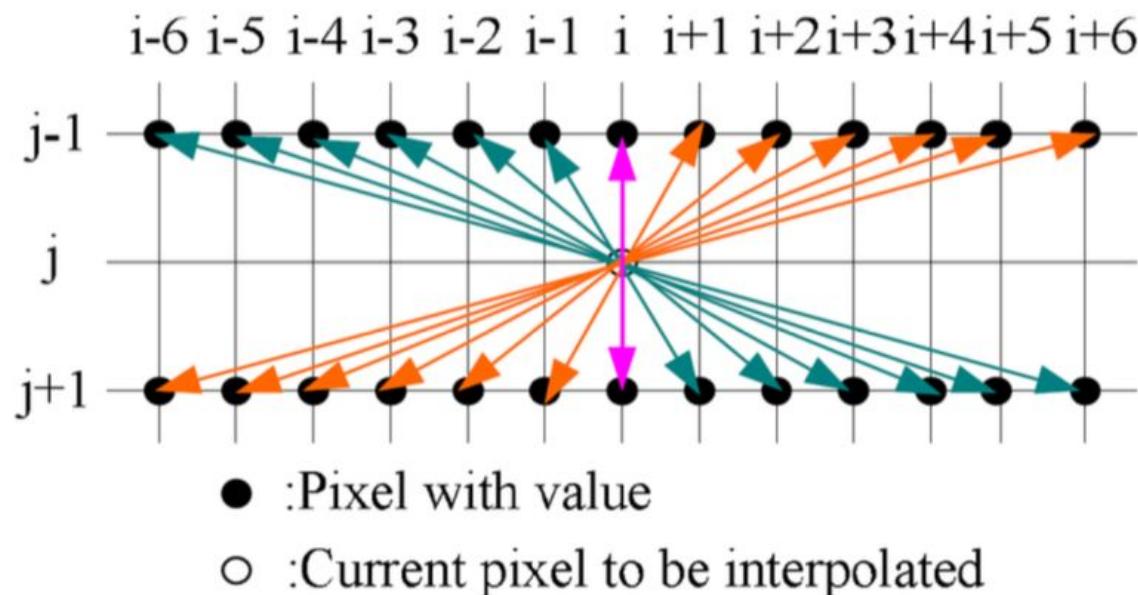


Edge-based algorithms

Weighted Edge-based Line Averaging (WELA)

Улучшение: интерполяция по 12 направлениям

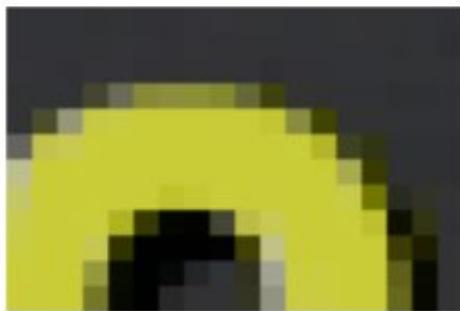
$$f_n(x, y) = \sum_{k=-1,0,1} W_k * \left(\frac{f(x+k, y-1) + f(x-k, y+1)}{2} \right), \quad W_1 + W_2 + W_3 = 1$$



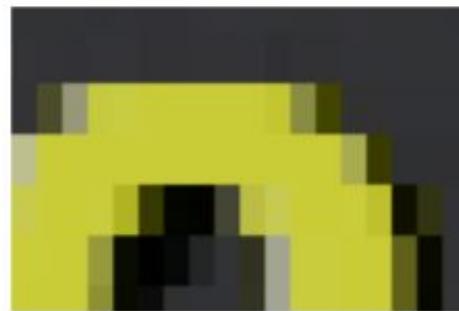
Сравнение базовых методов

Неподвижные объекты

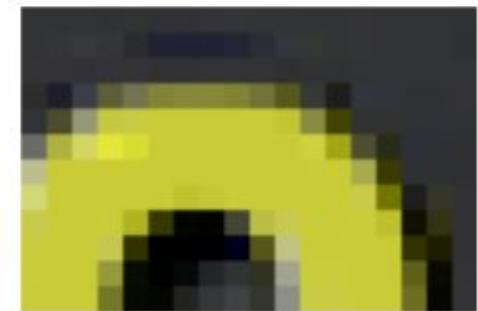
Field Insertion



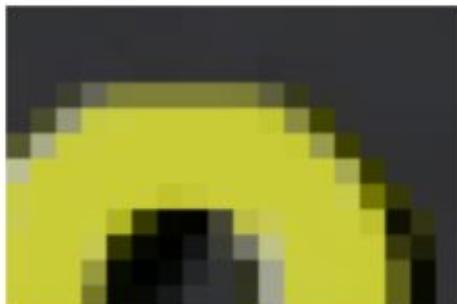
Line Repetition



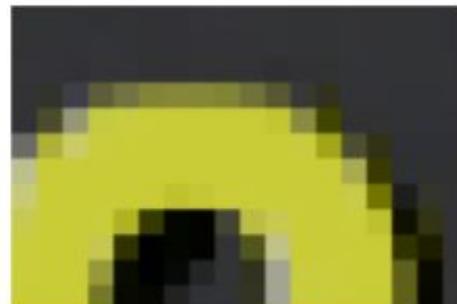
Linear VT



Edge-dependent



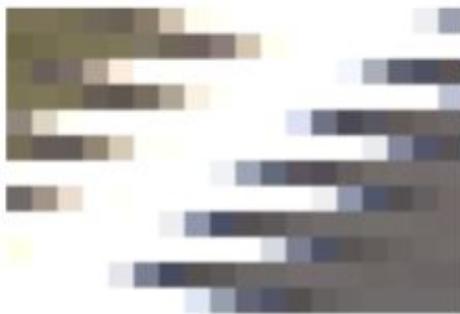
VT median



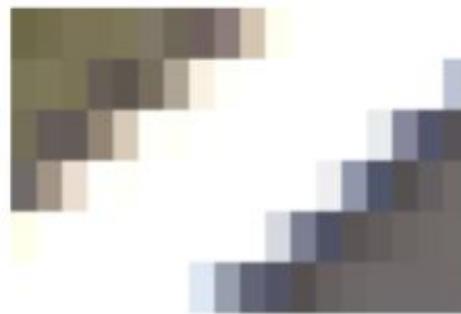
Сравнение базовых методов

Движущиеся объекты

Field Insertion



Line Repetition



Linear VT



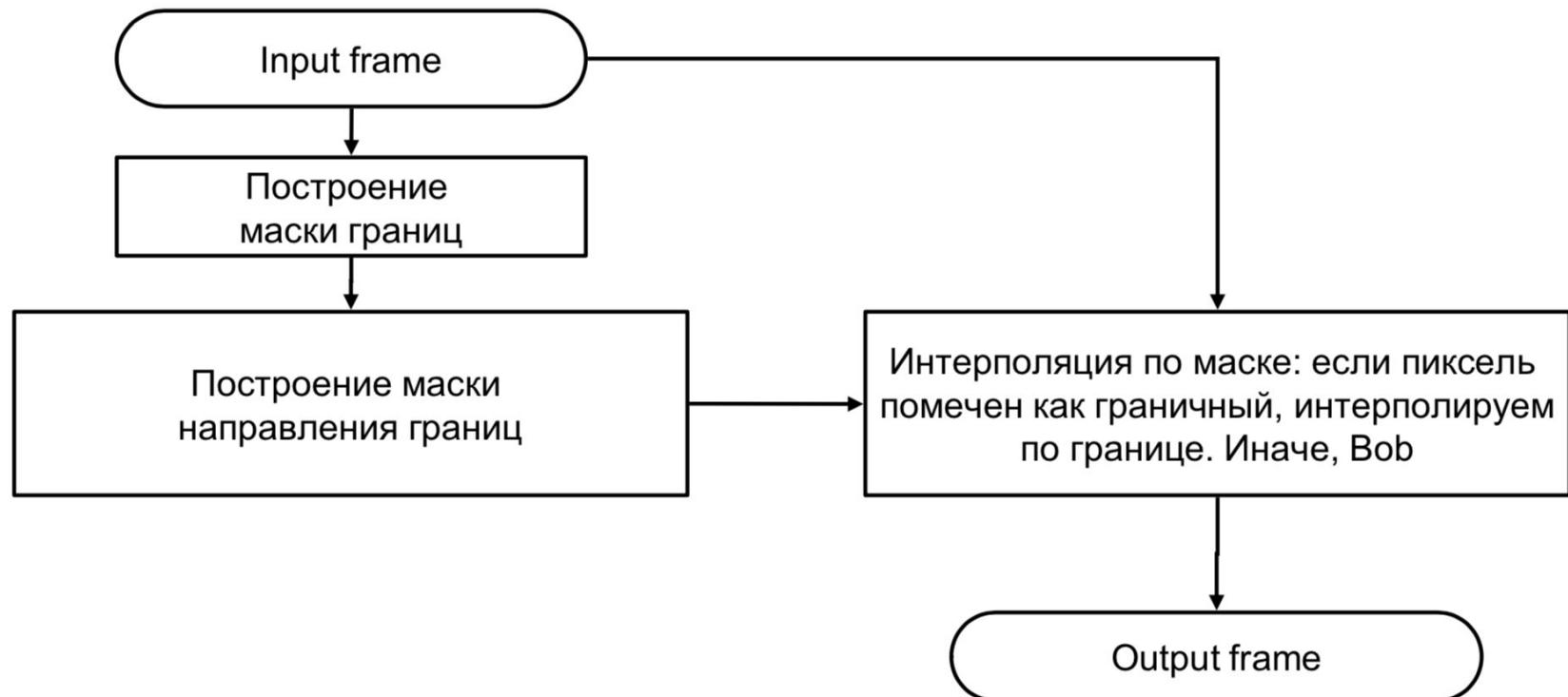
Edge-dependent



VT median



Enhanced Edge Directed Interpolation (EEDI2)



Построение маски границ

$$\text{if } (|c-b| < 10 \text{ and } |a-b| < 10 \text{ and } |a-c| < 10) \text{ OR}$$

$$(|e-f| < 10 \text{ and } |d-e| < 10 \text{ and } |d-f| < 10 \text{ and}$$

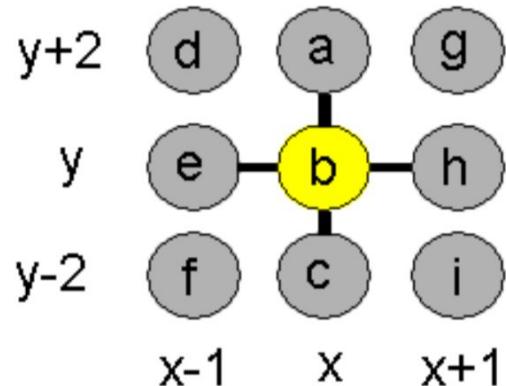
$$|h-i| < 10 \text{ and } |g-h| < 10 \text{ and } |g-i| < 10)$$

 then $b \notin \text{Edge}$

$$\begin{cases} \text{disp} > vthresh \\ Ix^2 + Iy^2 \geq mthresh \\ |Ix| + |Iy| \geq lthresh \end{cases} \Rightarrow b \in \text{Edge}$$

$$Ix = h - e$$

$$Iy = \max \left(\begin{array}{l} |a - c|, \\ |b - c|, \\ |a - b| \end{array} \right)$$



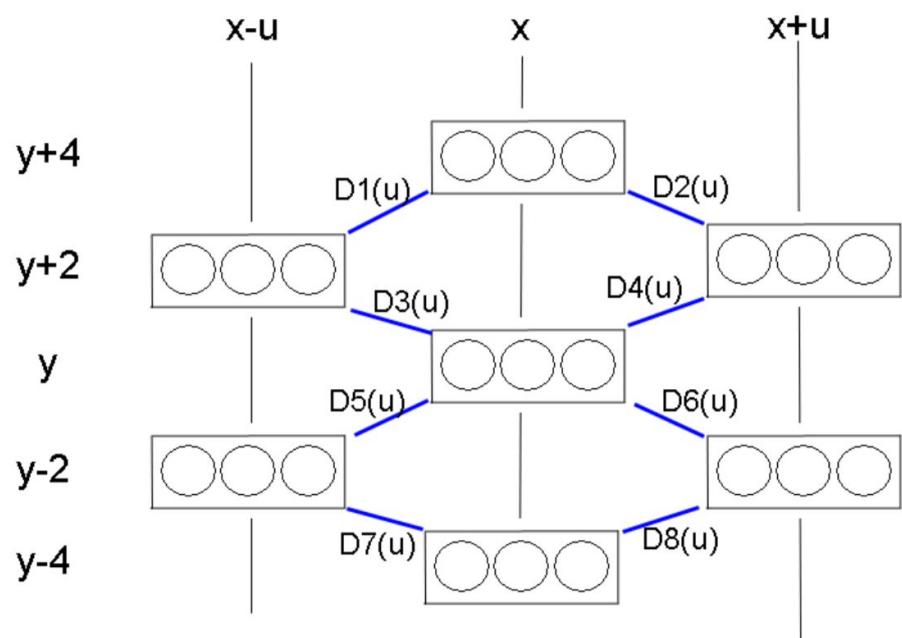
$$Ix = (e - b) + (h - b)$$

$$Iy = (c - b) + (a - b)$$

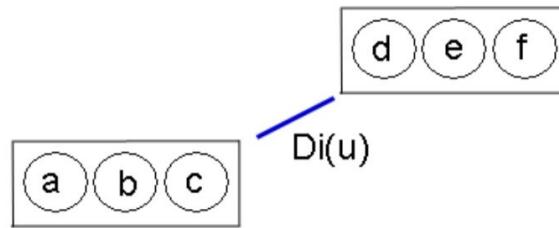
$disp$ – Дисперсия

Определение направления границы (1)

Рассматриваются пиксели, входящие в маску границ



$$\text{где } Di(u) = |a - d| + |b - e| + |c - f|$$



$$D_{down}(u) = D3(u) + \dots + D8(u)$$

$$D_{up}(u) = D1(u) + \dots + D6(u)$$

$$D_{left}(u) = D1(u) + D3(u) + D5(u) + D7(u)$$

$$D_{right}(u) = D3(u) + D4(u) + D5(u) + D6(u)$$

$$D_{center}(u) = D2(u) + D4(u) + D6(u) + D8(u)$$

Определение направления границы (2)

Усредняем 3 вектора, наиболее близких к медианному по направлению

$$u_{down} : D_{down}(u_{down}) = \min_u D_{down}(u)$$

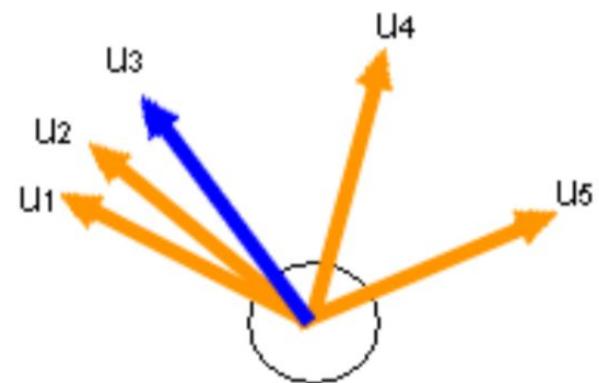
$$u_{up} : D_{up}(u_{up}) = \min_u D_{up}(u)$$

$$u_{left} : D_{left}(u_{left}) = \min_u D_{left}(u)$$

$$u_{right} : D_{right}(u_{right}) = \min_u D_{right}(u)$$

$$u_{center} : D_{center}(u_{center}) = \min_u D_{center}(u)$$

$$sort(u_{up}, u_{down}, u_{left}, u_{right}, u_{center}) \rightarrow (u_1, u_2, u_3, u_4, u_5)$$



$$\text{Edge direction} = \frac{u_1 + u_2 + u_3}{3}$$

Здесь наиболее близки к медиане векторы 1, 2 и 3

Пример маски краёв и их направлений



Edge map



Edge direction map

Выводы

Какие достоинства / недостатки этих методов?

Выводы

Достоинства:

- Интуитивность
- Простота реализации
- Высокая скорость работы (кроме EEDI2)

Недостатки:

- Размытость (blur)
- Зубчатость (jaggedness)
- Сползание строк (line crawling)
- Мерцание границ (edge flickering)

Недостатки

Line crawling



Недостатки

Edge flickering

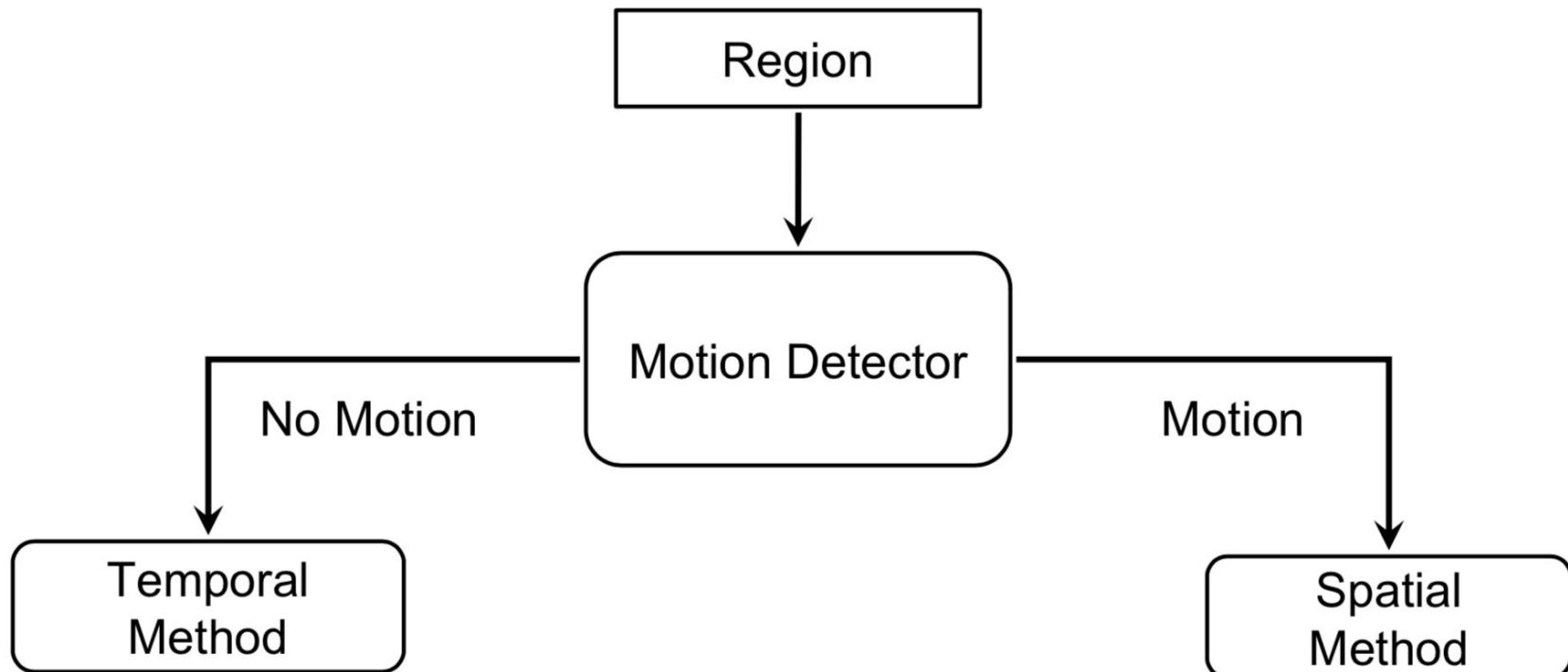


Содержание

- Мир аналогового телевидения
- Деинтерлейсинг
- Эвристические алгоритмы деинтерлейсинга
 - Spatial, Temporal, Spatio-Temporal, Edge-based
 - **Motion Adaptive**
 - Motion-Compensation-Based
- Нейросетевые алгоритмы
- Задание

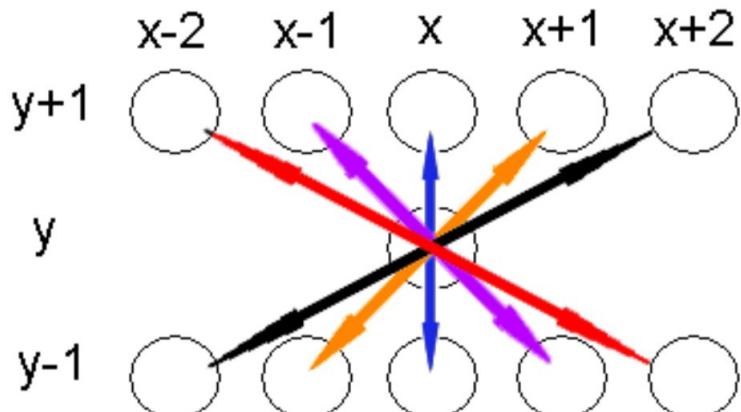
Motion-adaptive algorithms

Общая схема



Yet Another De-Interlacing Filter (Yadif)

Для пространственной интерполяции используется ELA

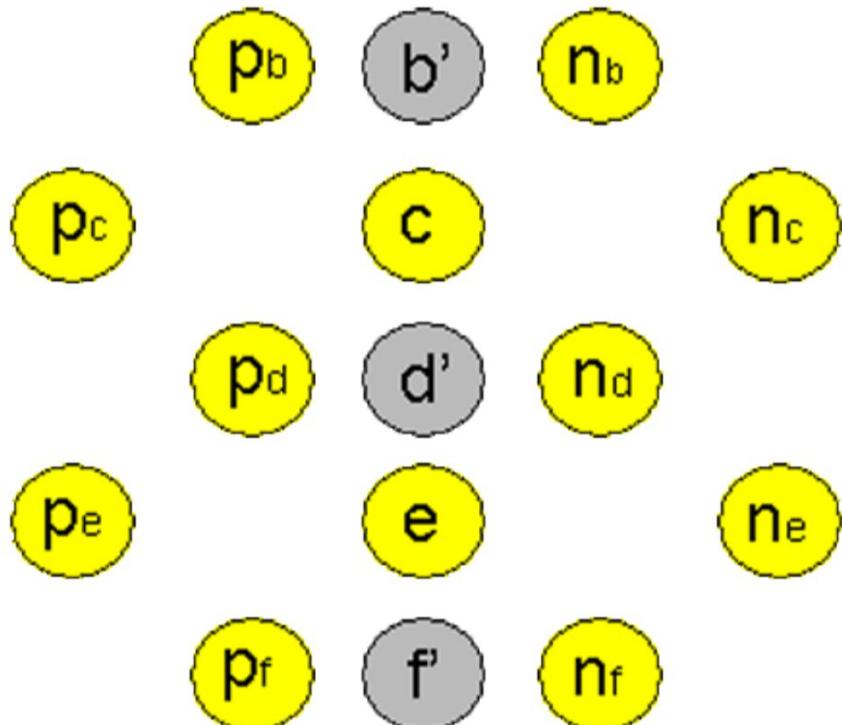


$$D(k) = |f_n(x+k, y-1) - f_n(x-k, y+1)|$$

$$u = \arg \min_{-2 \leq k \leq 2} \{D(k)\}$$

$$f_n(x, y) = \frac{f_n(x+u, y-1) + f_n(x-u, y+1)}{2}$$

$$d_{spat} = f_n(x, y)$$

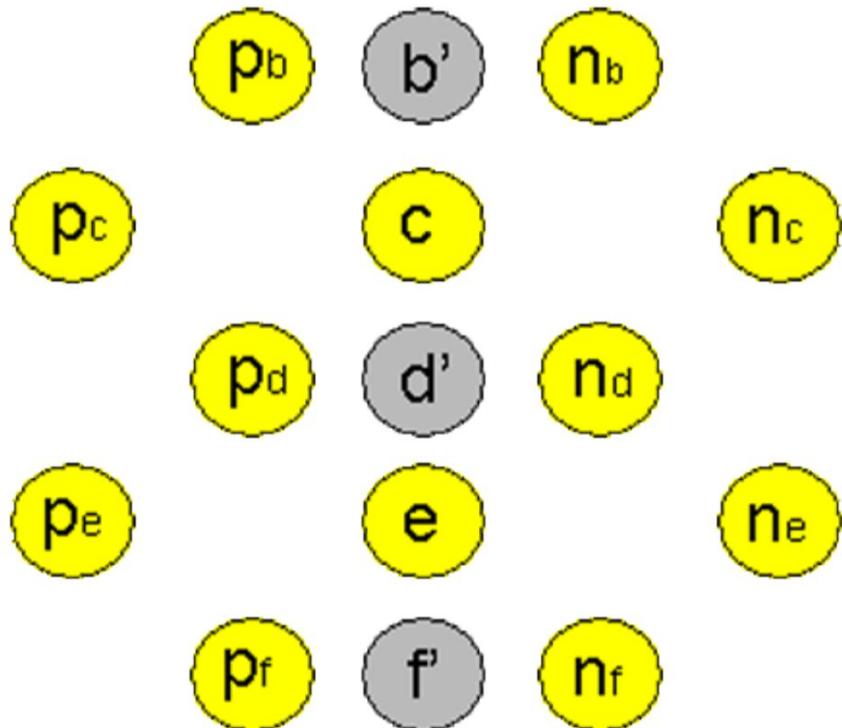


$$\Delta S_{\min} = \min \left(\begin{array}{l} d' - e, \\ d' - c, \\ \max(b' - c, f' - e) \end{array} \right)$$

$$\Delta S_{\max} = \max \left(\begin{array}{l} d' - e, \\ d' - c, \\ \min(b' - c, f' - e) \end{array} \right)$$

$$\Delta S = \max(\Delta S_{\min}, -\Delta S_{\max})$$

Temporal check



$$b' = \frac{p_b + n_b}{2} \quad f' = \frac{p_f + n_f}{2}$$

$$d' = d_{temp} = \frac{p_d + n_d}{2}$$

$$\Delta T_1 = |p - n|$$

$$\Delta T_2 = \frac{|p_e - e| + |p_c - c|}{2}$$

$$\Delta T_3 = \frac{|n_e - e| + |n_c - c|}{2}$$

$$\Delta T = \max(2 * \Delta T_1, \Delta T_2, \Delta T_3)$$

$$diff = \max(\Delta T, \Delta S)$$

$$f_n(x, y) = \begin{cases} d_{spat}, & \text{if } |d_{spat} - d_{temp}| < diff \\ d_{temp} + diff, & \text{if } (d_{spat} - d_{temp}) > diff \\ d_{temp} - diff, & \text{if } (d_{spat} - d_{temp}) < -diff \end{cases}$$



Interlaced



Yadif

Yadifmod:

- Для пространственной интерполяции — EEDI2
- Лучше качество, медленнее алгоритм

TDeint:

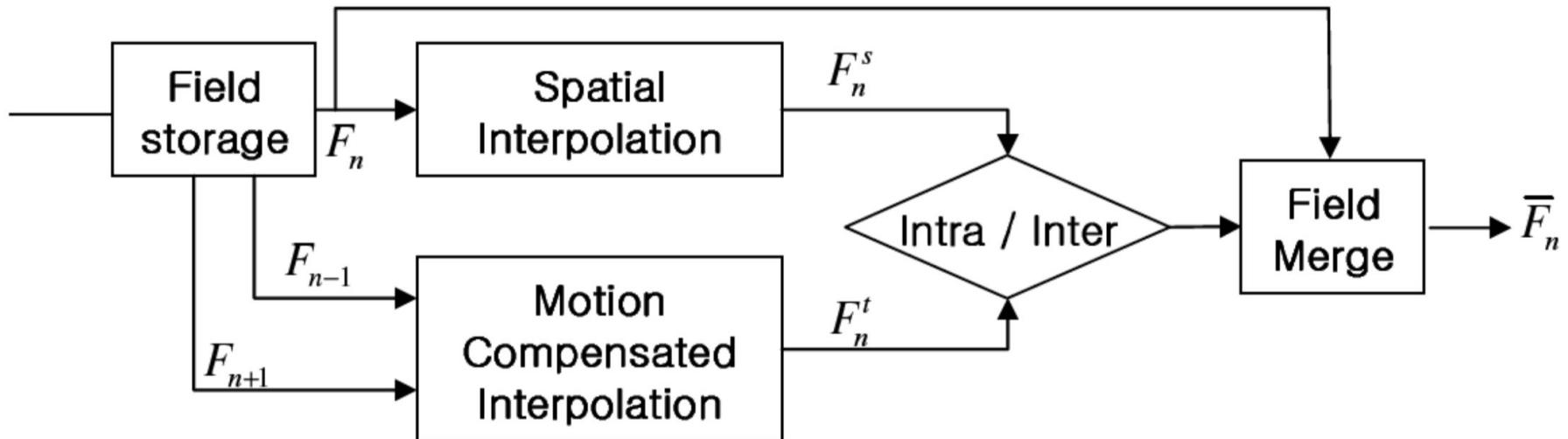
- Для пространственной интерполяции — EEDI2
- Способ временной интерполяции зависит от характера движения

Содержание

- Мир аналогового телевидения
- Деинтерлейсинг
- Эвристические алгоритмы деинтерлейсинга
 - Spatial, Temporal, Spatio-Temporal, Edge-based
 - Motion Adaptive
 - **Motion-Compensation-Based**
- Нейросетевые алгоритмы
- Задание

MC-based algorithms

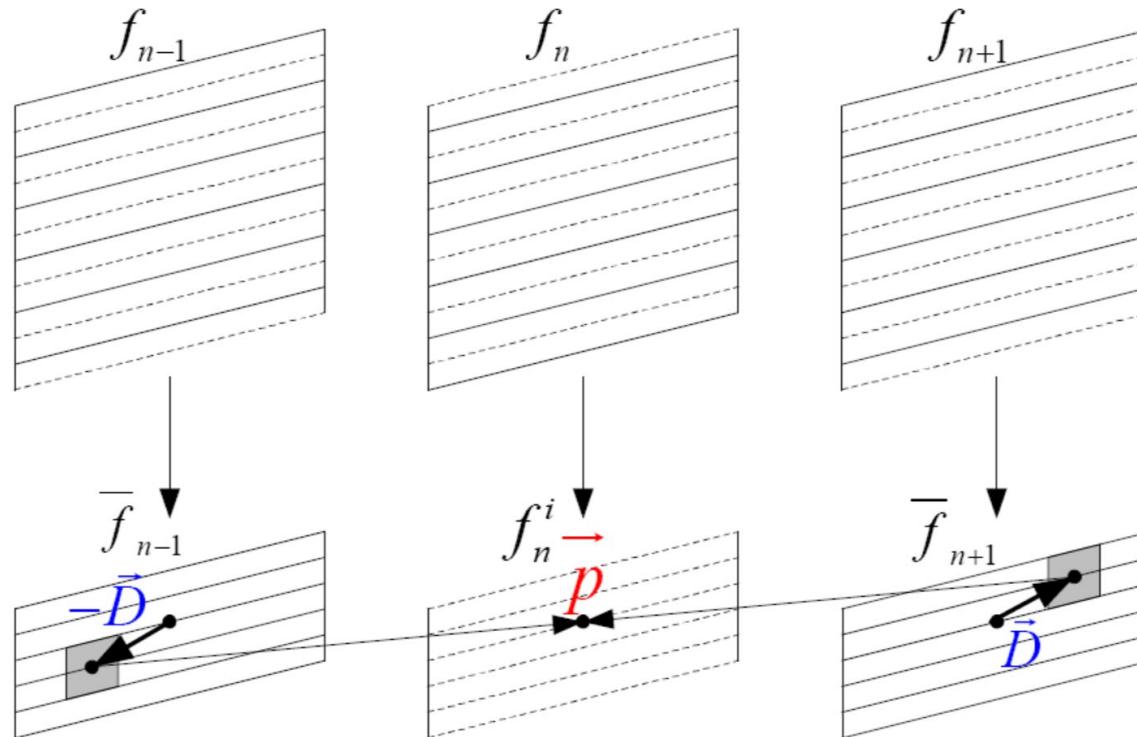
Общая схема



MC-based algorithms

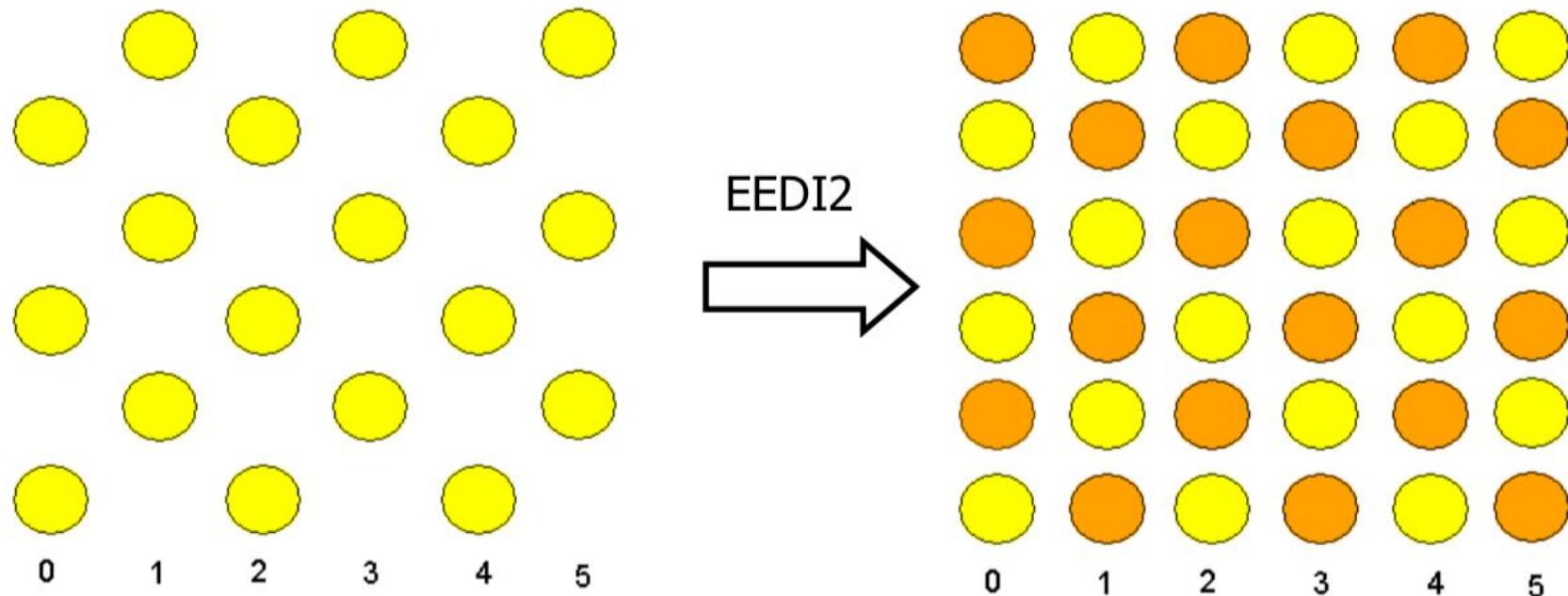
ME между полями одной чётности

$$f_n^i(\vec{p}) = \frac{1}{2}(\hat{f}_{n-1}(\vec{p} - \vec{D}) + \hat{f}_{n+1}(\vec{p} + \vec{D}))$$

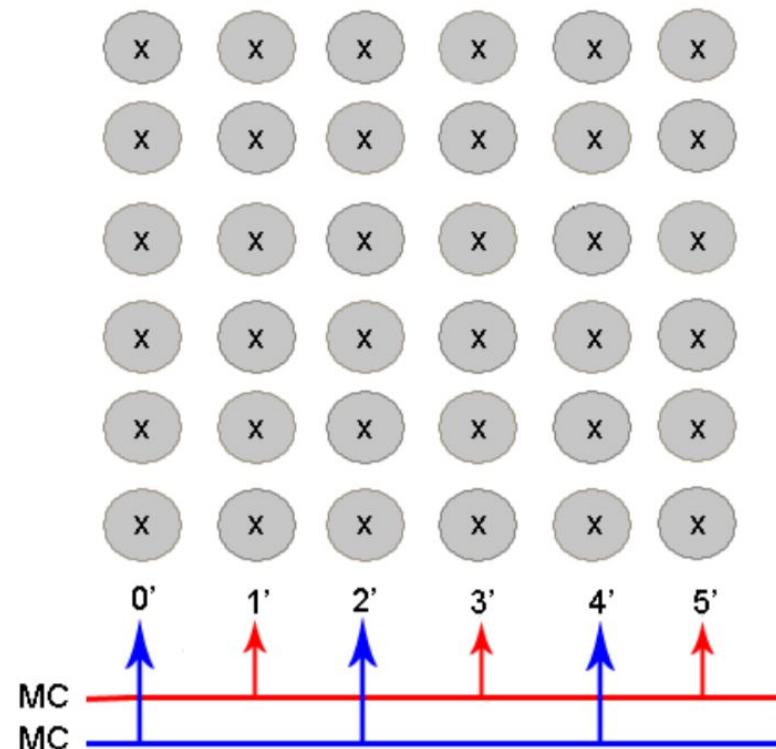
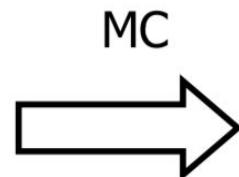
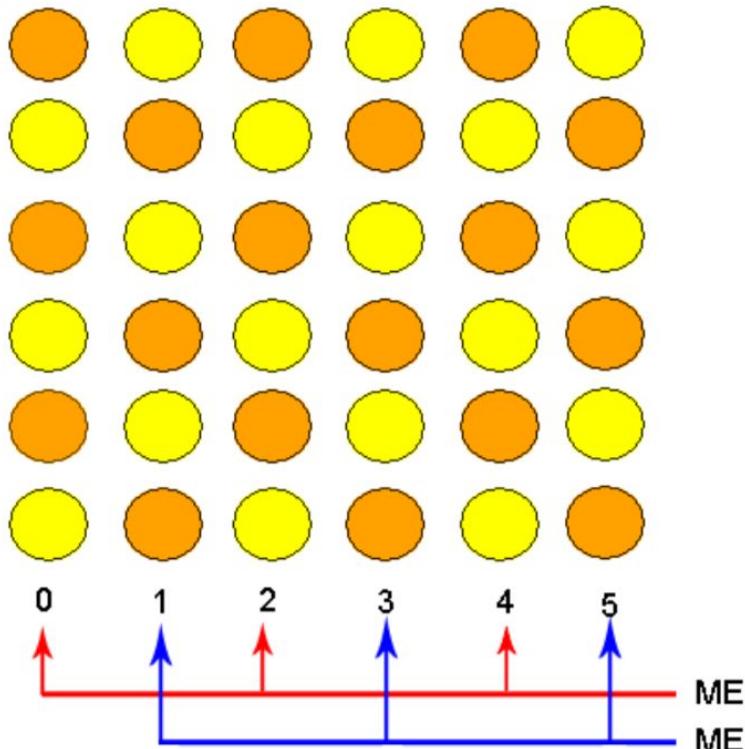


Step 1: Spatial Interpolation

Интерполяция качественным алгоритмом (EEDI2, NNEDI)

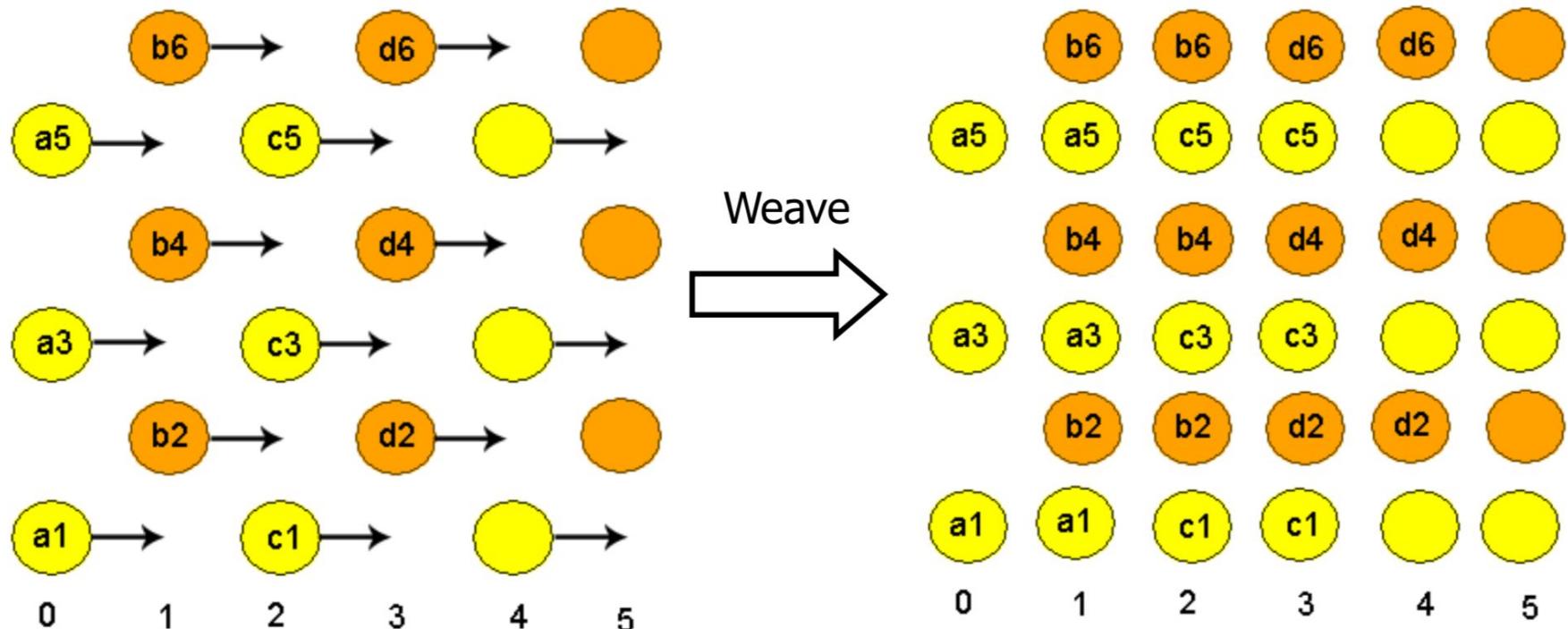


Компенсация движения по полям одной чётности
в обоих направлениях



Step 3: Temporal Interpolation

Для интерполяции используется простой Weave



MCBob

Дальнейший план:

1. Понять, насколько каждый из алгоритмов лучше для каждого пикселя
2. Взвесить значения, полученные разными алгоритмами
3. Получить наилучшую интерполяцию



CorrMask — маска весов для **пространственной** интерполяциями

Строится на основе:

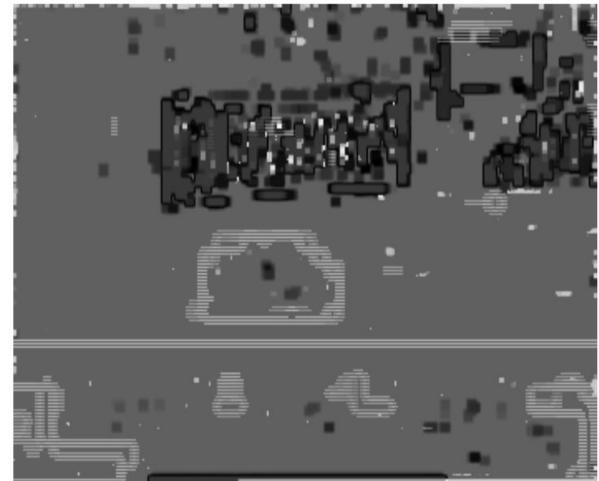
- MC-frame
- Маски ошибки компенсации (расхождение между MC-frame и Spatial-interpolated frame)
- Маски вертикальных границ
- Анализа векторов ME



Ошибка компенсации



Маска вертикальных
границ



CorrMask

MCBob

NotStatic Mask

NotStatic Mask — маска весов для **временной** интерполяции

\tilde{f}_n — кадр, интерполированный пространственным способом

$$d_n(x, y) = \left| \tilde{f}_{n+2}(x, y) - \tilde{f}_n(x, y) \right|$$

$$Motn_n(x, y) = \max(d_n(x, y), d_{n+1}(x, y), d_{n+2}(x, y))$$

$$MM_n(x, y) = \max_{\substack{-1 \leq \Delta x \leq 1 \\ -1 \leq \Delta y \leq 1}} f(x + \Delta x, y + \Delta y) - \min_{\substack{-1 \leq \Delta x \leq 1 \\ -1 \leq \Delta y \leq 1}} f(x + \Delta x, y + \Delta y)$$

$$NotStatic_n(x, y) = \frac{\frac{Motn_n(x, y) - 1}{MM_n(x, y) - 1} - MThr1}{MThr2 - MThr1} * 255$$

$MThr1, MThr2 - const$

MCBob

Пример NotStatic Mask



$Motn_n$



MM_n



$NotStatic_n$

$$256 * \hat{f}_n(x, y) = \text{NotStatic}_n(x, y) * [(256 - \text{CorrMask}_n(x, y)) * MC_n(x, y) + \text{CorrMask}_n(x, y) * Spat(x, y)] + \\ + (256 - \text{NotStatic}_n(x, y)) * Temp_n(x, y) + 128$$

$\text{Temp}_n(x, y)$ – пиксель, интерполированный временным способом

$\text{Spat}_n(x, y)$ – пиксель, интерполированный пространственным способом

$MC_n(x, y)$ – скомпенсированный пиксель

- Высокая точность интерполяции
- Очень медленный (компенсация движения и EEDI2)

Сравнение

Yadif



Сравнение

Yadifmod



Сравнение TDeint + EEDI2



Сравнение MCBob

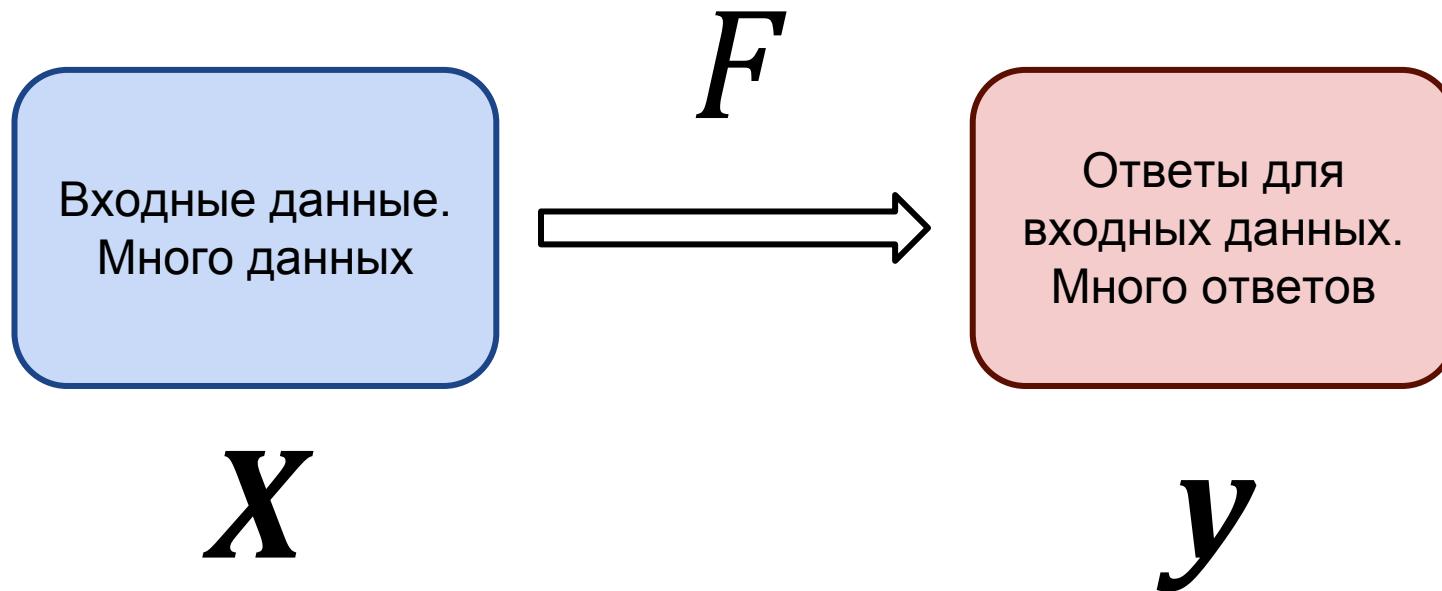


Содержание

- Мир аналогового телевидения
- Деинтерлейсинг
- Эвристические алгоритмы деинтерлейсинга
 - Spatial, Temporal, Spatio-Temporal, Edge-based
 - Motion Adaptive
 - Motion-Compensation-Based
- **Нейросетевые алгоритмы**
- Задание

Нейросетевые алгоритмы

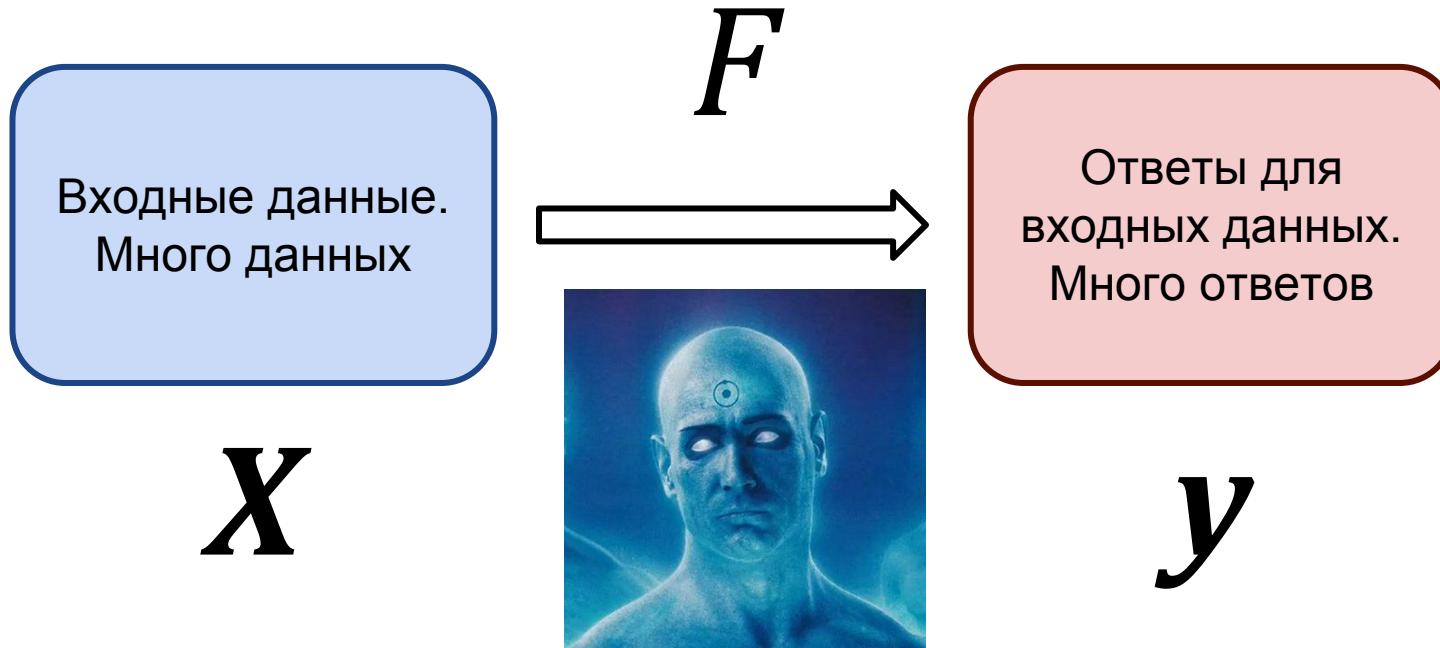
Общая идея (1)



F — сложно определить аналитически

Нейросетевые алгоритмы

Общая идея (2)

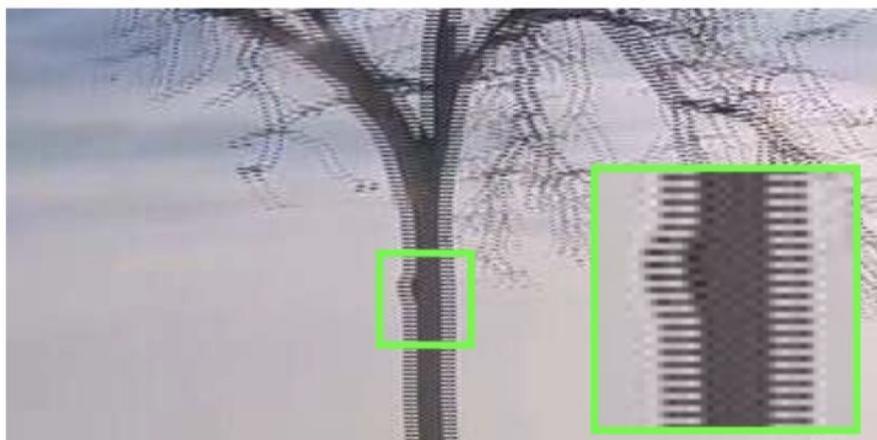


$F \approx \text{Neural Network}$

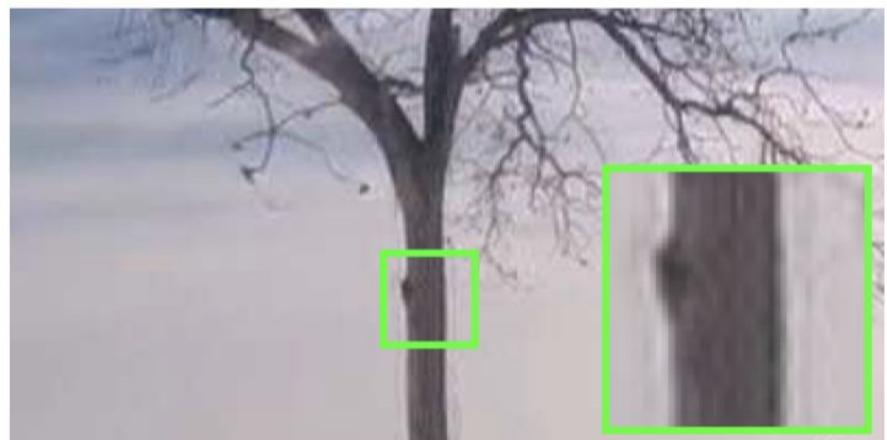
Deep Video Deinterlacing

Более общая задача — Superresolution (SR,
увеличение разрешения)

Но применение сети для SR напрямую даёт
некачественный результат:



(a) Input



(b) SRCNN

Deep Video Deinterlacing

Архитектура сети



- Обычно нейросети зависят от размера входа
- В деинтерлейсинге важно не терять информацию

Нужна архитектура, **не зависящая** от размера входа

Deep Video Deinterlacing

Архитектура сети



- Обычно нейросети зависят от размера входа
- В деинтерлейсинге важно не терять информацию

Нужна архитектура, **не зависящая** от размера входа

Какие идеи?

Deep Video Deinterlacing

Архитектура сети



- Обычно нейросети зависят от размера входа
- В деинтерлейсинге важно не терять информацию

Нужна архитектура, **не зависящая** от размера входа

Решение: Fully Convolutional Network (FCN)

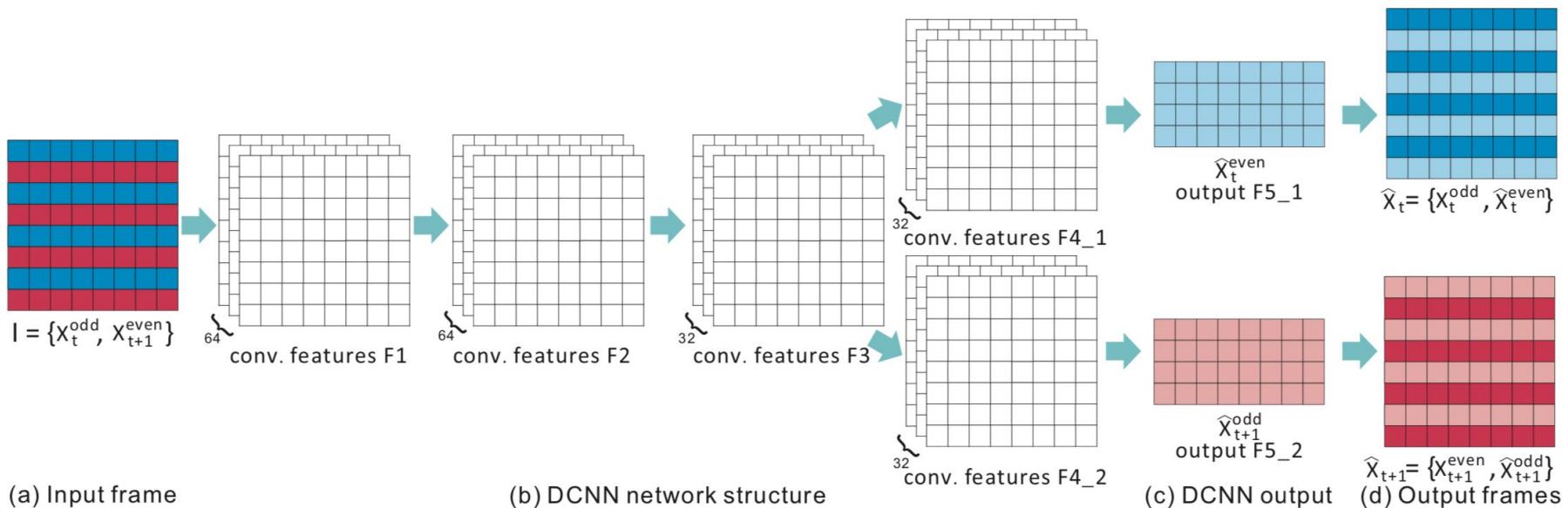
- Семантическая сегментация
- Superresolution

Deep Video Deinterlacing



Deep Video Deinterlacing

Предложенная архитектура



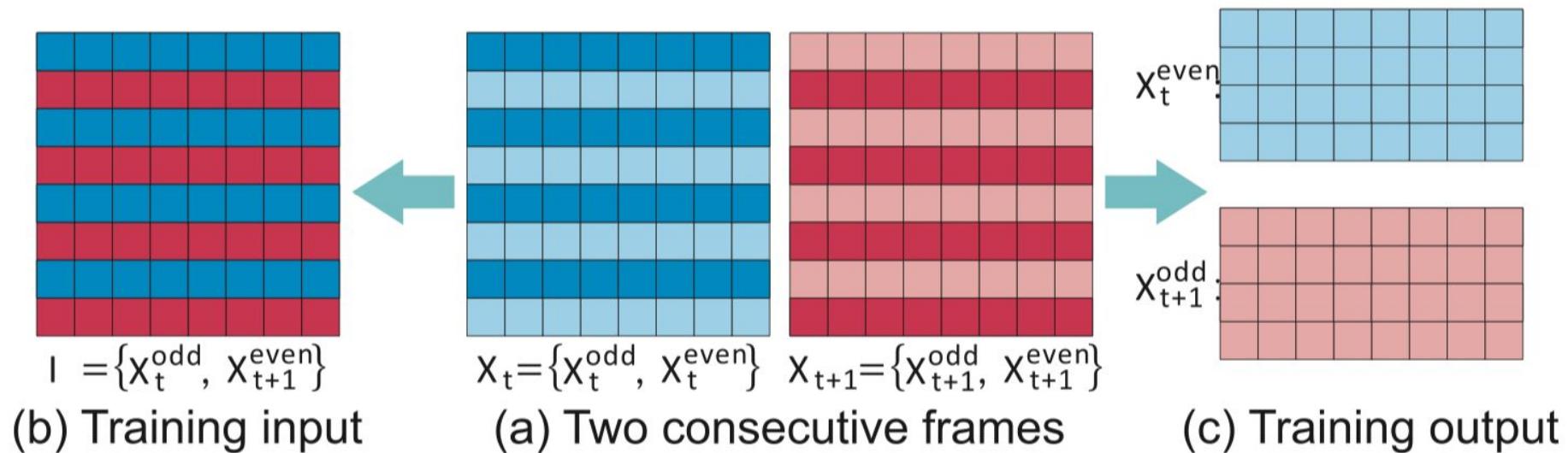
Deep Video Deinterlacing

Обучающий датасет

- 51 видео
- Из каждого видео по 3 пары последовательных кадров
- Ресайз в 512×512
- Разбиение на патчи 64×64 — всего 9792 патча
- 80% — train, 20% — test

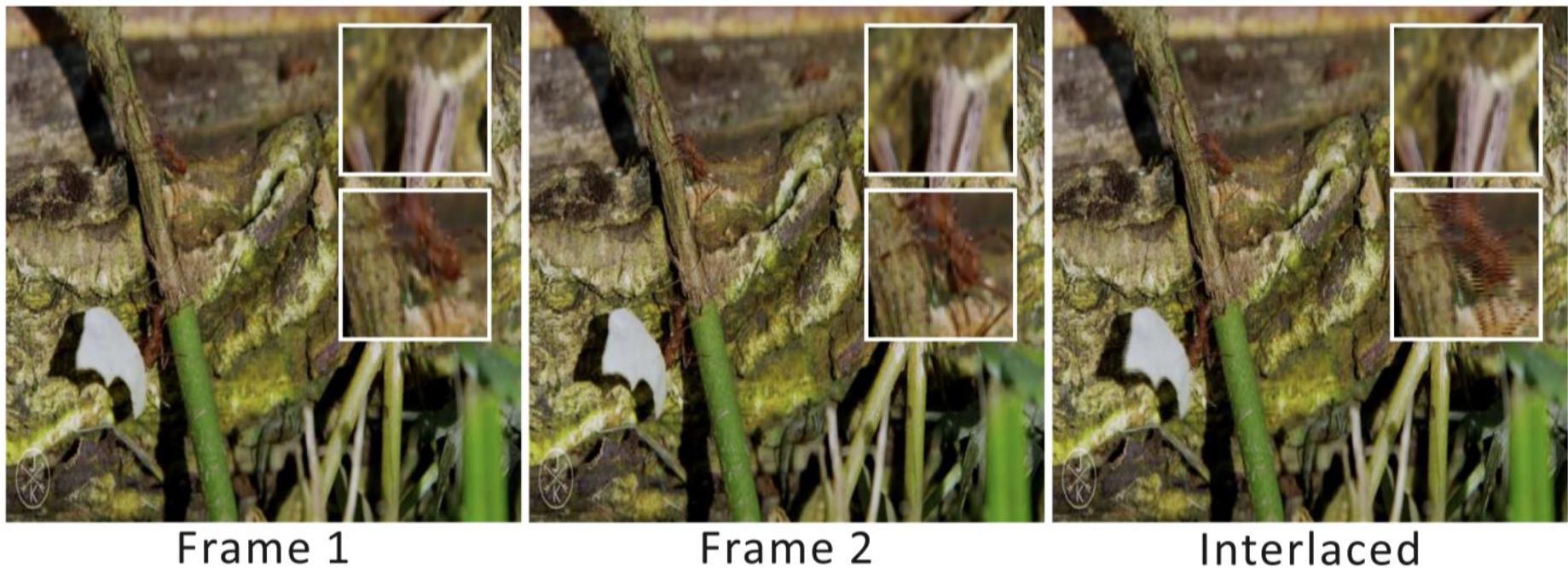
Deep Video Deinterlacing

Генерация датасета (1)



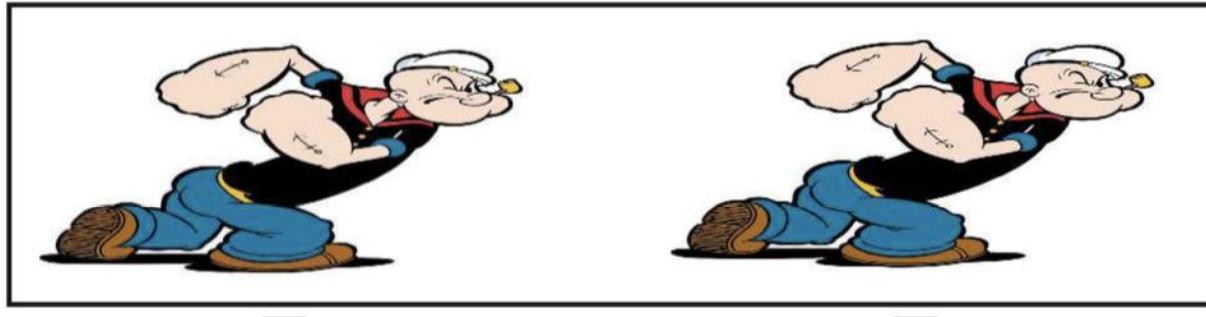
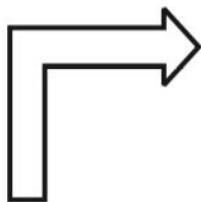
Deep Video Deinterlacing

Генерация датасета (2)



Deep Video Deinterlacing

Почему 2 поля на входе?



(a) Interlaced frame



(b) Reconstructed frame 1



(c) Reconstructed frame 2

Deep Video Deinterlacing

Обучение



Функция потерь:

$$W^* = \arg \min \frac{1}{N_p} \sum_p \left(\|\widehat{\mathbf{X}}_{t,p}^{\text{even}} - \mathbf{X}_{t,p}^{\text{even}}\|_2^2 + \|\widehat{\mathbf{X}}_{t+1,p}^{\text{odd}} - \mathbf{X}_{t+1,p}^{\text{odd}}\|_2^2 + \lambda_{TV} (TV(\widehat{\mathbf{X}}_{t,p}) + TV(\widehat{\mathbf{X}}_{t+1,p})) \right)$$

TV — Total Variation

- 200 эпох
- Adam optimizer
- Batch size — 64
- Обучение на nVidia Titan X заняло 4 часа

Deep Video Deinterlacing

Сравнение



(b) SRCNN (trained with our dataset)

(c) Blown-ups

(d) Ours

Deep Video Deinterlacing

Interlaced



Deep Video Deinterlacing

Bicubic Interpolation



Deep Video Deinterlacing

SRCNN



Deep Video Deinterlacing

Proposed



Содержание

- Мир аналогового телевидения
- Деинтерлейсинг
- Эвристические алгоритмы деинтерлейсинга
 - Spatial, Temporal, Spatio-Temporal, Edge-based
 - Motion Adaptive
 - Motion-Compensation-Based
- Нейросетевые алгоритмы
- **Задание**

Задание по деинтерлейсингу

Две части:

1. Реализовать эвристический алгоритм
2. Реализовать нейросетевой алгоритм (TF + Keras)

Требования:

- Скорость работы не ниже 1 FPS (на CPU)
- Размер весов сети не более 10 МБ
- Не списывать

Время на выполнение: **2 недели**