



---

# Деинтерлейсинг

Петров Александр

*Video Group*  
*CS MSU Graphics & Media Lab*

# Содержание

- Введение
  - Понятие интерлейсинга
  - Понятие деинтерлейсинга
  - Постановка задачи
  - Визуальные дефекты
  - Примеры
- Классификация методов
- Простые методы
- Сложные методы
- Дальнейшие планы
- Список материалов

# Введение

**Интерлейсинг (Interlacing)** - метод отображения, передачи или хранения видео, при котором:

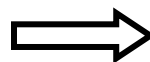
- Захват видео по строчкам в разные моменты времени: сначала четные, потом нечетные (или наоборот).
- Поле (field) – набор строчек одинаковой четности, соответствующих одному моменту времени.
- Видео-поток с удвоенной частотой.
- При построении кадра, четные и нечетные поля смешиваются. Образуется один кадр (frame).

# Введение

Поле #5



Поле #6



Кадр #3





# Введение

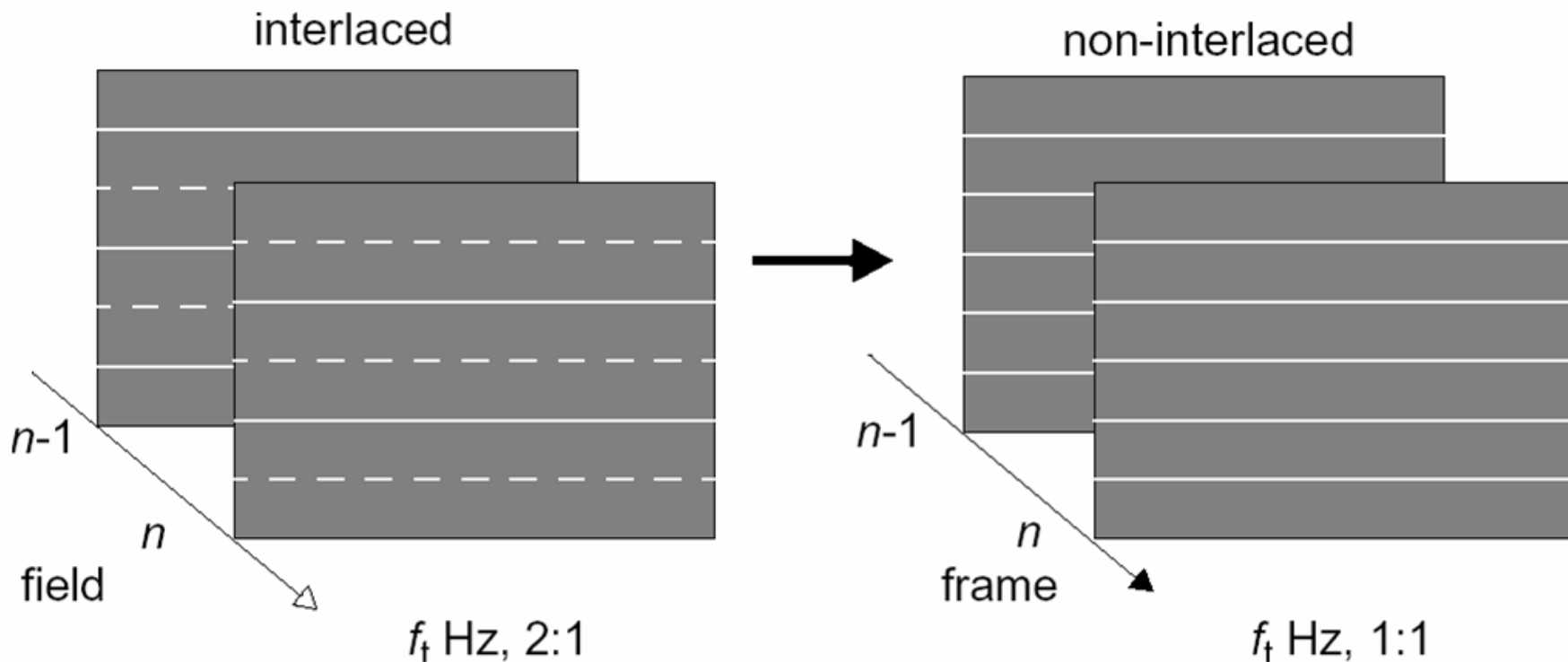
---

**Деинтерлейсинг (Deinterlacing)**— процесс создания кадров из полукадров чересстрочного формата для дальнейшего вывода на экран с прогрессивной развёрткой.

Применяется:

- В компьютерных системах обработки видео.
- В LCD и плазменных дисплеях.

# Введение



# Введение

## Постановка задачи



Задача – интерполировать недостающие пиксели и, в то же время обеспечить хорошее качество изображения

$$p_o(i, j, k) = \begin{cases} p_i(i, j, k), & (j + k) \% 2 = 0, \\ \hat{p}(i, j, k), & \text{otherwise,} \end{cases}$$

$k$  - номер поля

$i, j$  - положение пикселя

$p_i(i, j, k)$  - исходный пиксель

$\hat{p}(i, j, k)$  - интерполированный пиксель

$p_o(i, j, k)$  - результат

# Введение

## Визуальные дефекты

---



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



# Введение

## Пример



# Введение

## Пример



# Введение

## Пример

**до**



**после**



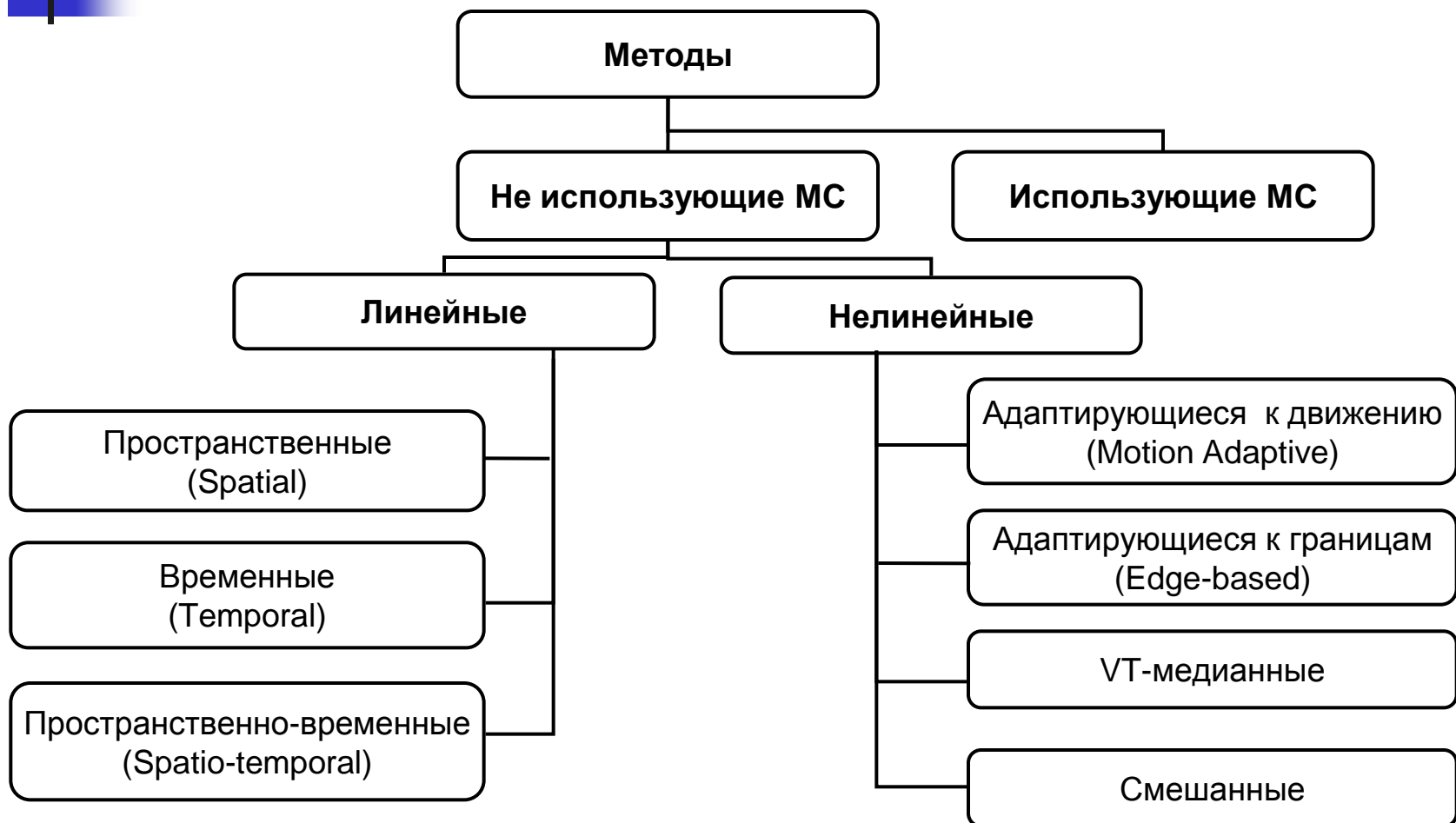


# Содержание

---

- Введение
- Классификация методов
- Простые методы
- Сложные методы
- Дальнейшие планы
- Список материалов

# Классификация методов



# Содержание

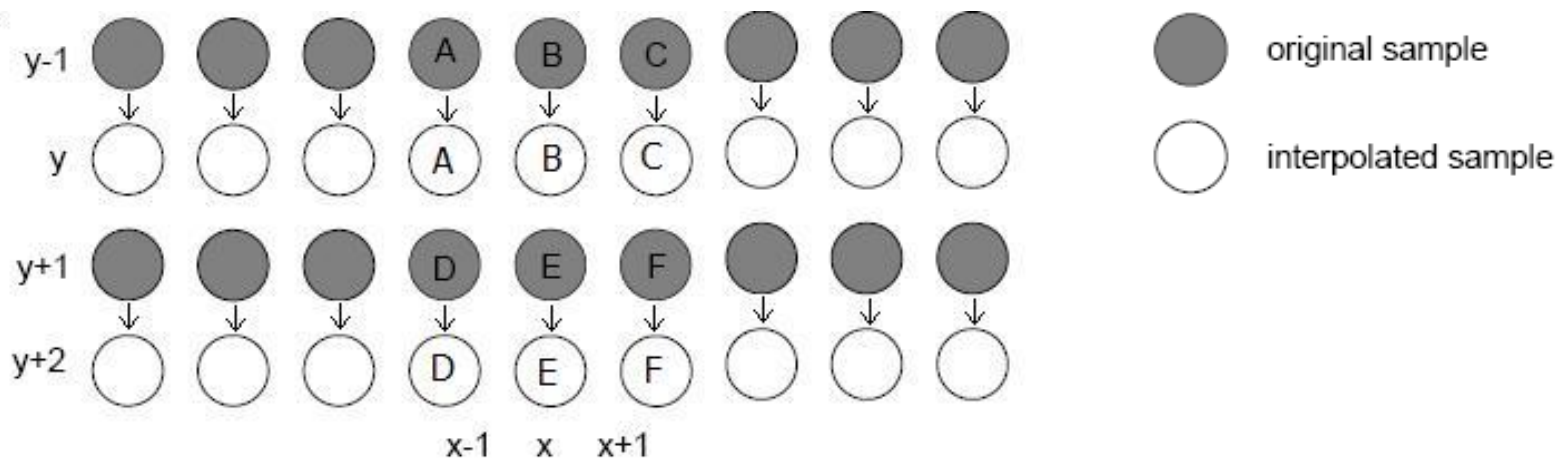
- Введение
- Классификация методов
- Простые методы
  - Пространственные (Spatial)
  - Временные (Temporal)
  - Пространственно-временные (Spatio-temporal)
  - Адаптирующиеся к границам (Edge-based)
- Сложные методы
- Дальнейшие планы
- Список материалов

# Простые методы

## Spatial

- Line Repetition (LR)

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

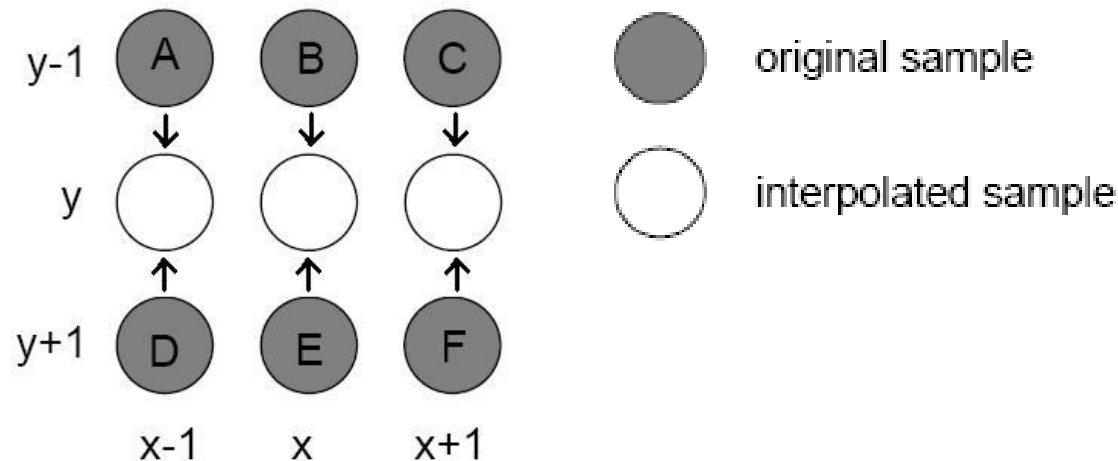


# Простые методы

## Spatial

- Line Averaging (LA, "Bob")

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





# Содержание

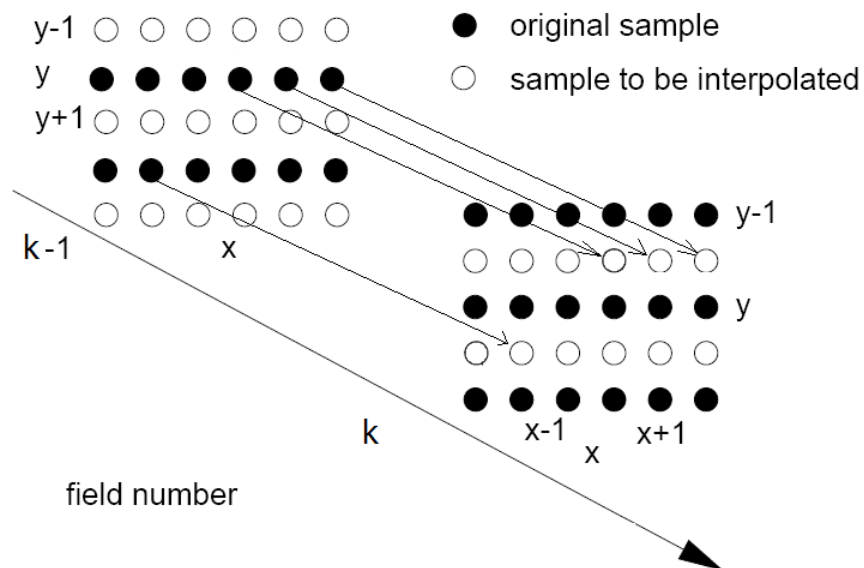
- Введение
- Классификация методов
- Простые методы
  - Пространственные (Spatial)
  - **Временные (Temporal)**
  - Пространственно-временные (Spatio-temporal)
  - Адаптирующиеся к границам (Edge-based)
- Сложные методы
- Дальнейшие планы
- Список материалов

# Простые методы

## Temporal

- Field Insertion (FI, FR, "Weave")

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

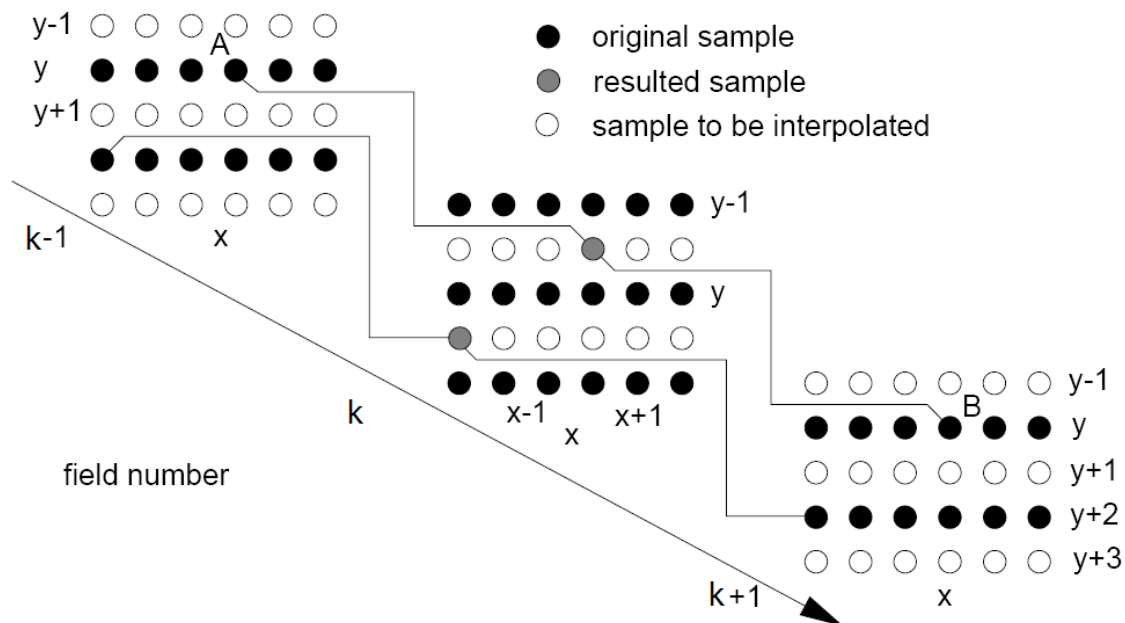


# Простые методы

## Temporal

- Inter-field Line Averaging

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



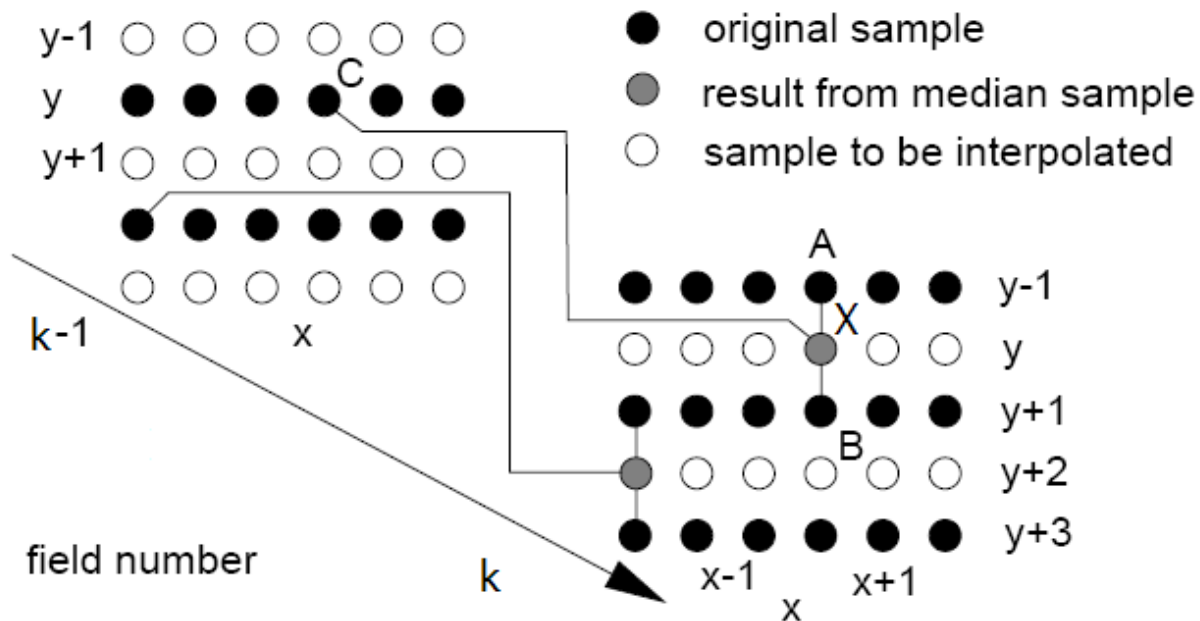
# Содержание

- Введение
- Классификация методов
- Простые методы
  - Пространственные (Spatial)
  - Временные (Temporal)
  - Пространственно-временные (Spatio-temporal)
    - VTMF
    - VT linear
  - Адаптирующиеся к границам (Edge-based)
- Сложные методы
- Дальнейшие планы
- Список материалов

# Простые методы

## Spatial-temporal

- Vertical-Temporal Median Filter



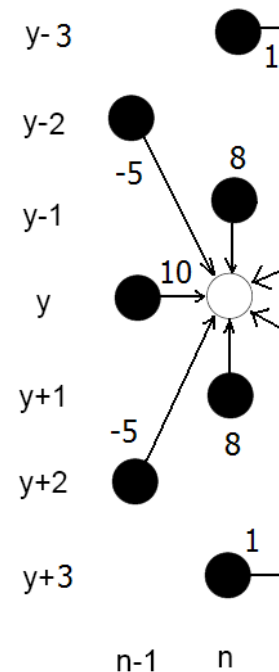
$$p_0(X) = \text{median}(A, B, C)$$

# Простые методы Spatial-temporal

## Vertical-Temporal Linear

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

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



● original sample  
○ sample to be interpolated

# Содержание

- Введение
- Классификация методов
- Простые методы
  - Пространственные (Spatial)
  - Временные (Temporal)
  - Пространственно-временные (Spatio-temporal)
  - Адаптирующиеся к границам (Edge-based)
- Сложные методы
- Дальнейшие планы
- Список материалов

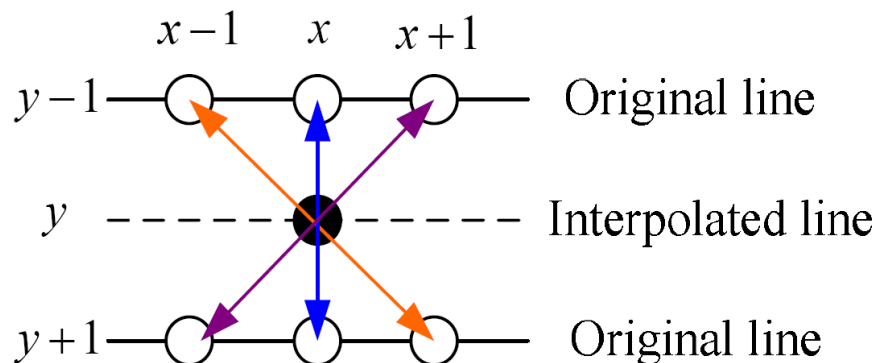
# Простые методы

## Адаптирующиеся к границам

- Edge-based Line Averaging (ELA, ELI, EDI)

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

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

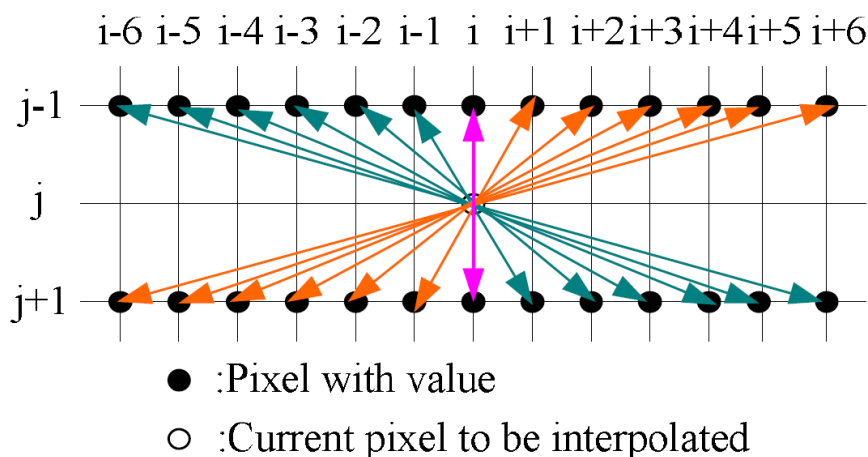




# Простые методы Адаптирующиеся к границам

## Улучшения:

- Можно рассматривать 12 направлений



- Weighted Edge-based Line Averaging (WELA)

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

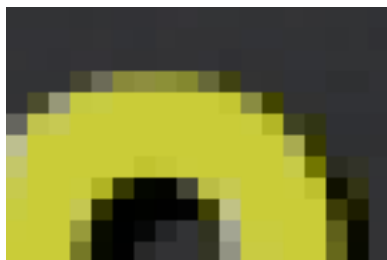
# Простые методы Сравнение



# Простые методы

## Сравнение (Неподвижные регионы)

Field Insertion



Line Repetition



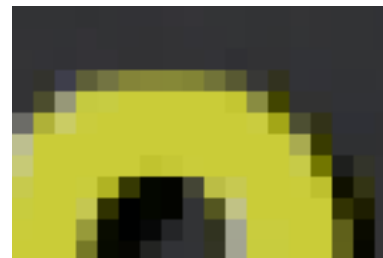
Linear VT



Edge-dependent



VT median



# Простые методы

Сравнение (регионы с движением)

Field Insertion



Line Repetition



Linear VT



Edge-dependent



VT median

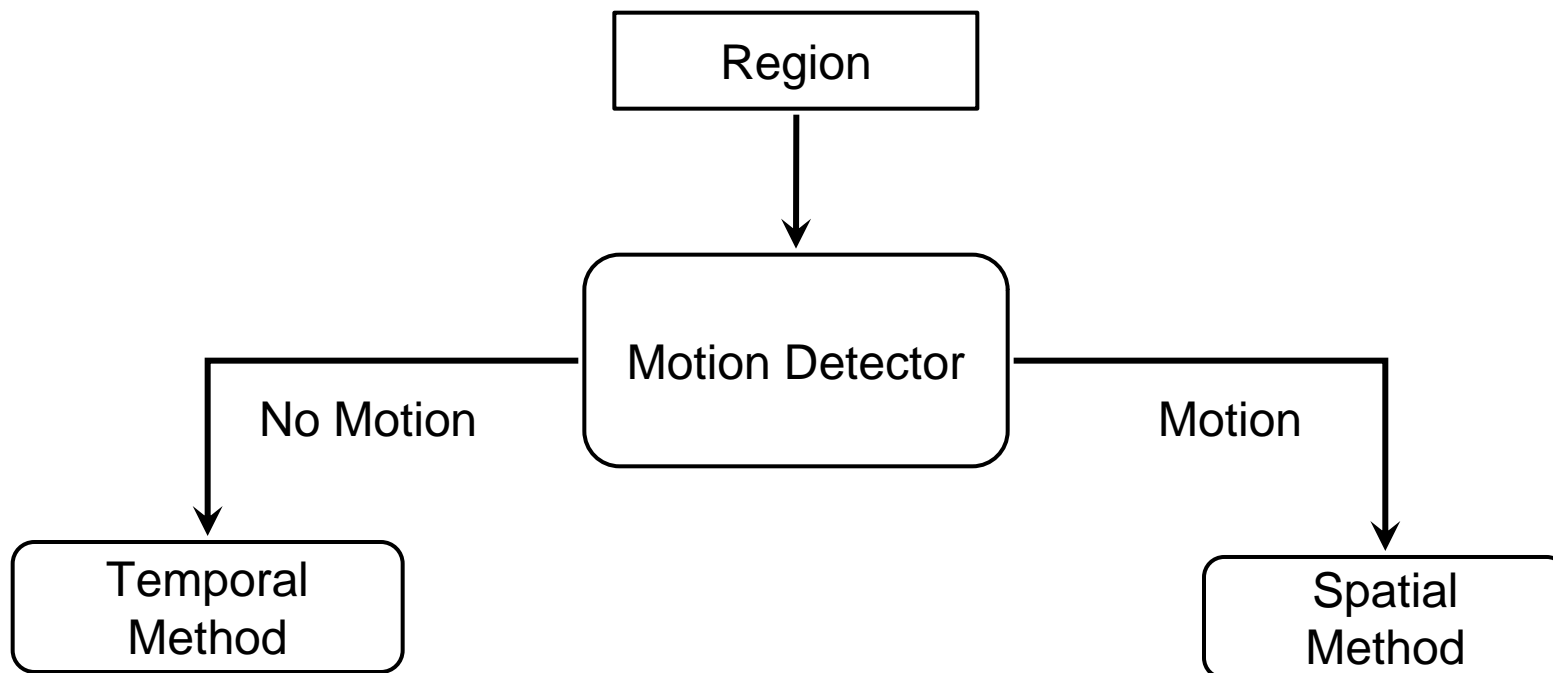


# Содержание

- Введение
- Классификация методов
- Простые методы
- Сложные методы
  - Motion Adaptive
  - Motion Compensation based
  - MSU filter
- Дальнейшие планы
- Список материалов

# Сложные методы

## Motion Adaptive



# Содержание

- Введение
- Классификация методов
- Простые методы
- Сложные методы
  - Motion Adaptive
    - Motion Detector
    - HMDERP
  - Motion Compensation based
  - MSU filter
- Дальнейшие планы
- Список материалов

# Motion Adaptive

## Motion Detector



- Сравниваем поля одной четности
- Строим маску движения
- Морфологическая операция открытия (чтобы убрать шумы в движении)



# Motion Adaptive Motion Detector

Сравниваем поля одной четности



(a)

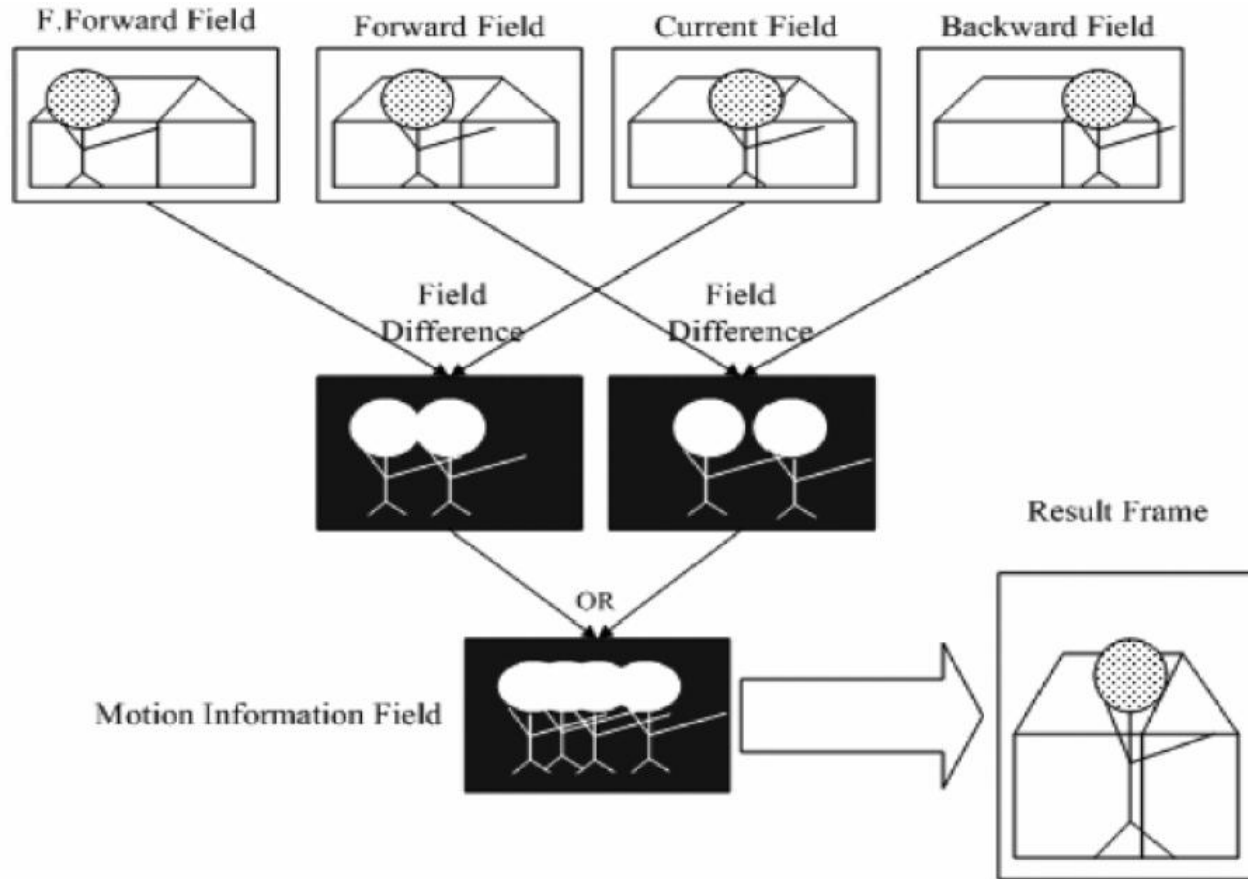


(b)

Difference between (a) the fields  $f_n$  and  $f_{n-1}$  (b) the fields  $f_n$  and  $f_{n-2}$

# Motion Detector

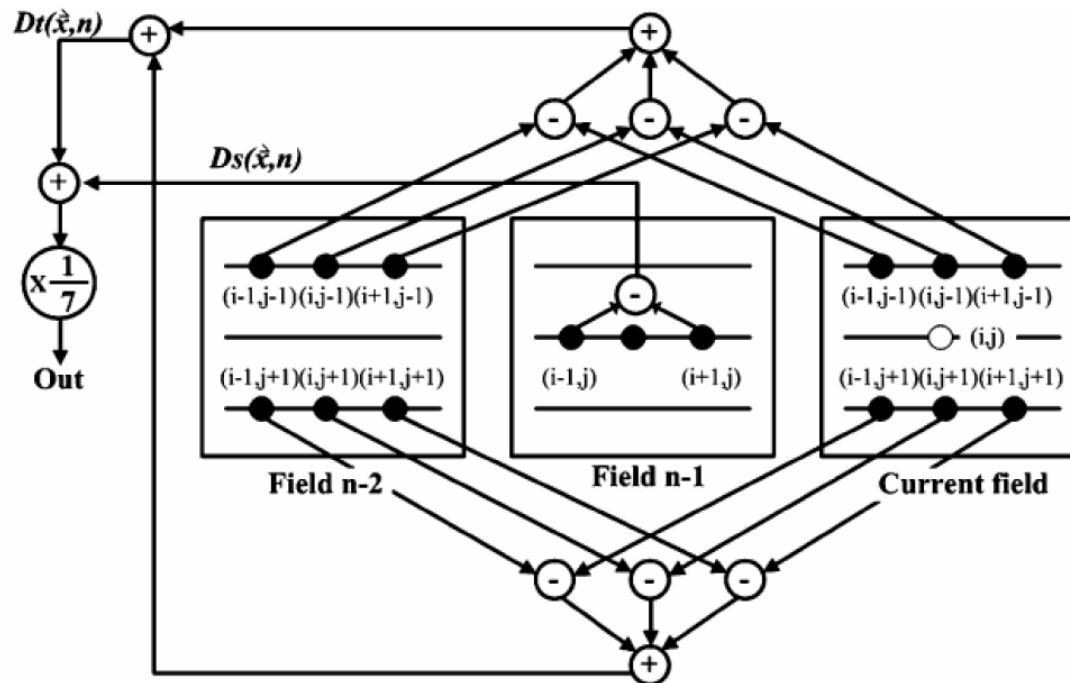
## 4-field motion detection



# Motion Detector

## Moving-Stationary Detector

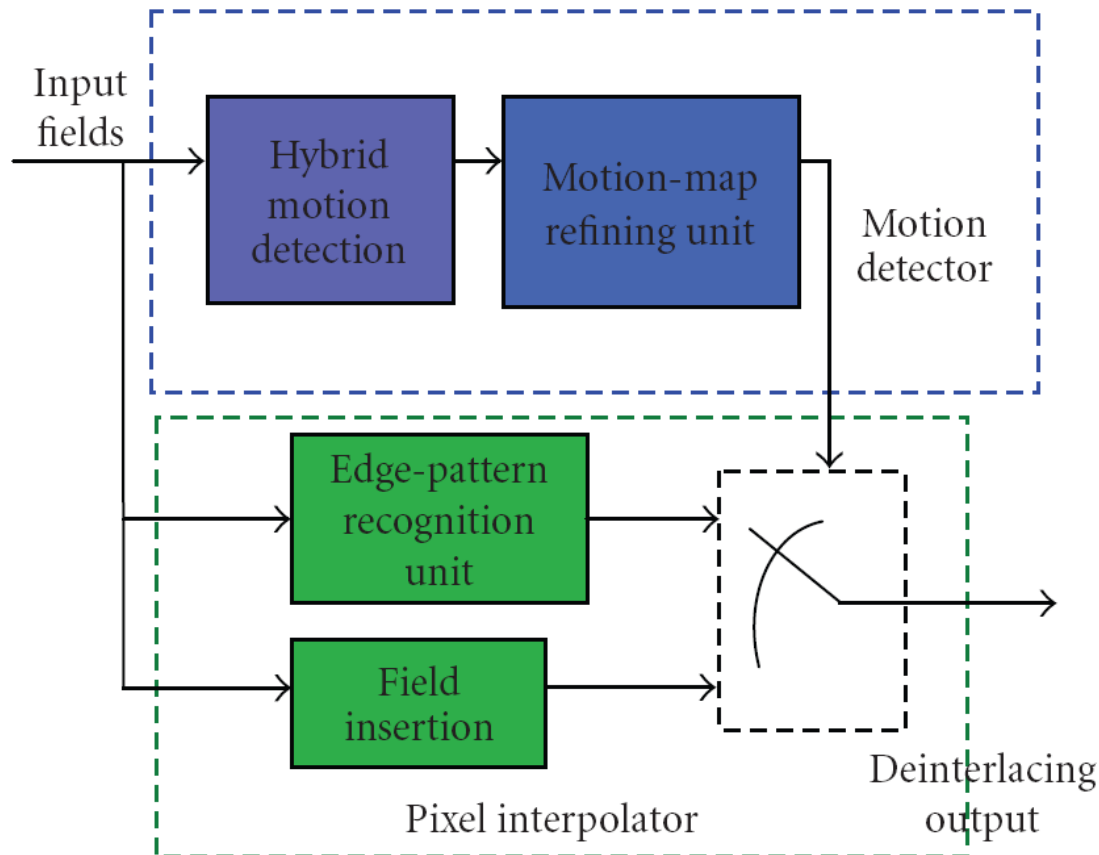
$$CIP(\vec{x}, n) = \begin{cases} S, & \frac{1}{7}(D_T(\vec{x}, n) + D_S(\vec{x}, n)) < threshold \\ M, & otherwise \end{cases}$$



# Содержание

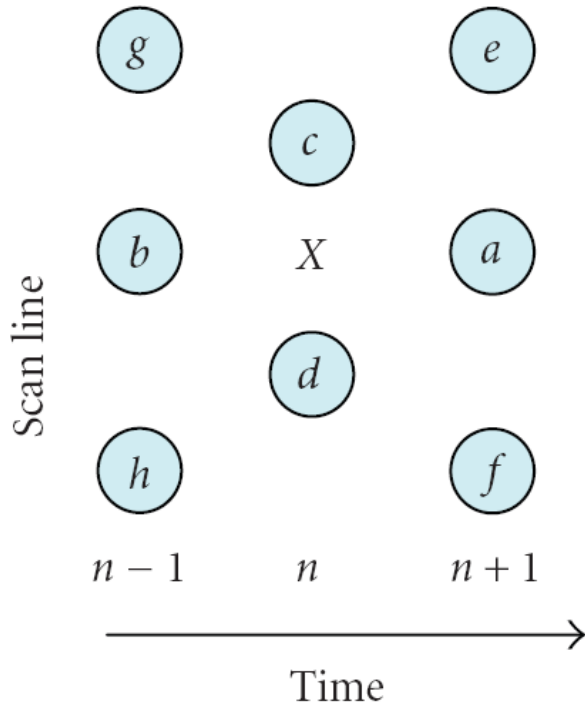
- Введение
- Классификация методов
- Простые методы
- Сложные методы
  - Motion Adaptive
    - Motion Detector
    - **HMDERP**
  - Motion Compensation based
  - MSU filter
- Дальнейшие планы
- Список материалов

# Motion Adaptive HMDEPR



# Motion Adaptive HMDEPR

Hybrid Motion Detector (HMD):



$$\text{diff1} = \text{abs}(a - b)$$

$$\text{diff2} = \text{abs}[b - (c + d)/2]$$

$$\text{diff3} = \text{abs}[b - (g + h)/2]$$

$$\text{diff4} = \text{abs}[a + (e + f)/2 - b - (g + h)/2]$$

**if**  $\text{diff1} > \text{TH1}$

▷ **1st condition**

    flag ← motion

**else if**  $\text{diff2} > \text{TH1}$  **AND**  $\text{diff3} < \text{TH2}$

▷ **2nd condition**

    flag ← motion

**else if**  $\text{diff4} > \text{TH3}$

▷ **3rd condition**

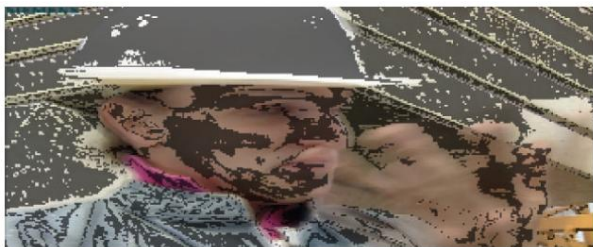
    flag ← motion

**else**

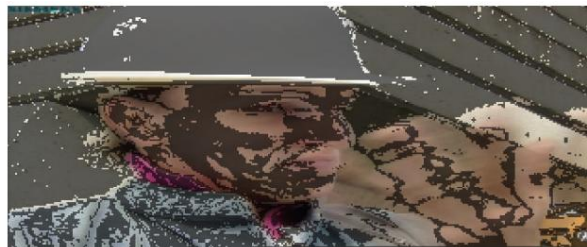
    flag ← stationary

# Motion Adaptive HMDEPR

Результат детектора движения - карта движения (motion map).



(a)



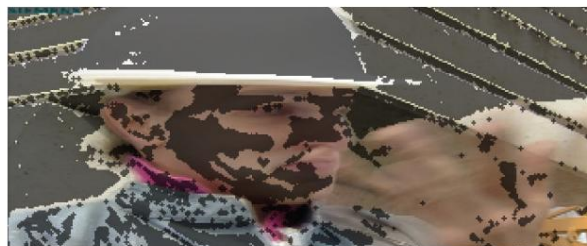
(b)



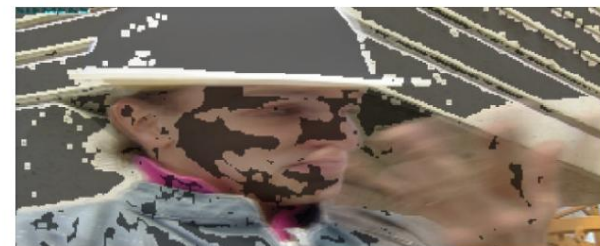
(c)



(d)



(e)



(f)

- (a) 1<sup>st</sup> condition
- (b) 2<sup>nd</sup> condition
- (c) 3<sup>rd</sup> condition

- (d) output of HMD
- (e) erosion
- (f) opening

# HMDEPR

## Тестирование



Предложенный алгоритм.

Интерполированы (FI) только статичные регионы.



3 Field Motion Adaptive (3FMA).

Интерполированы только статичные регионы.



# Motion Adaptive HMDEPR

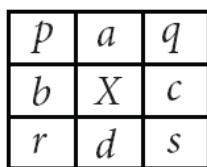
## Edge Pattern Recognition (EPR):

- Рассматривается область 3x3
- 4 типа шаблона
  - 3H1L, 3L1H
  - 2H2L - угол
  - 2H2L - линия
- 18 различных шаблонов

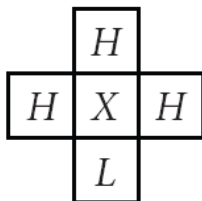
$$ave = \frac{a + b + c + d}{4}$$

$$\hat{p}_i = \begin{cases} H, & p > ave, \\ L, & p \leq ave \end{cases}$$

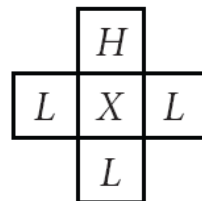
$\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{p}_4$ , для  $a, b, c, d$  соответственно



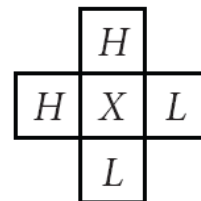
(a)



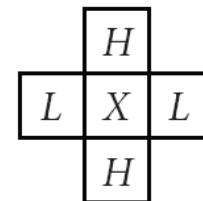
(b)



(c)



(d)



(e)

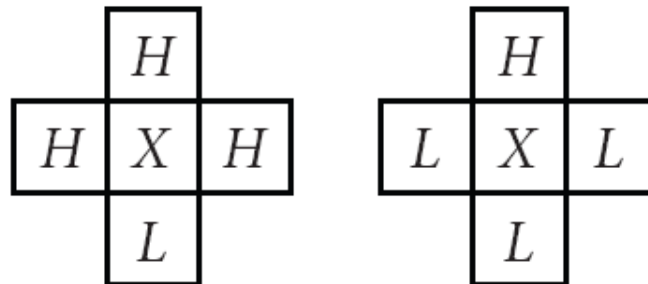
# Motion Adaptive

## HMDEPR



- В случае 3H1L, 3L1H наиболее вероятно, что центральный пиксел принадлежит к большинству (H для 3H1L и L для 3L1H)

$$p_0 = \begin{cases} \text{median}(p_{H_1}, p_{H_2}, p_{H_3}), & 3H1L, \\ \text{median}(p_{L_1}, p_{L_2}, p_{L_3}), & 3L1H, \end{cases}$$



# Motion Adaptive HMDEPR



- 2H2L – угол:

$p$	$H$	$q$
$H$	$X$	$L$
$r$	$L$	$s$

if  $|p - q| > |r - s|$   
 $X \leftarrow \min(H, H)$   
 else  
 $X \leftarrow \max(L, L)$

$p$	$H$	$q$
$H$	$H$	$L$
$r$	$L$	$s$

$p$	$H$	$q$
$H$	$L$	$L$
$r$	$L$	$s$

- 2H2L – линия:

$p$	$H$	$q$
$L$	$X$	$L$
$r$	$H$	$s$

if  $|p - q| + |r - s| > |p - r| + |q - s|$   
 $X \leftarrow \min(H, H)$   
 else  
 $X \leftarrow \max(L, L)$

$p$	$H$	$q$
$L$	$H$	$L$
$r$	$H$	$s$

$p$	$H$	$q$
$L$	$L$	$L$
$r$	$H$	$s$

# Motion Adaptive HMDEPR



- Пиксели  $b$  и  $c$ , также как и  $X$ , недостающие. Как же их получить?
  - EPR without MAP (Motion-Adaption Prediction)
  - EPR with MAP
- EPR without MAP
  - Усреднение по строкам (Line Averaging)

$p$	$a$	$q$
$b$	$X$	$c$
$r$	$d$	$s$

$$b = \frac{p+r}{2} \quad c = \frac{q+s}{2}$$

- EPR with MAP
  - Усреднение по строкам (Line Averaging), если  $b$  ( $c$ ) принадлежат области движения (Motion Area)
  - Иначе, берем пиксель с предыдущего поля

# HMDEPR

## Тестирование



FIGURE 9: Visual quality of intra-interpolation methods. (a) LA, (b) ELA, (c) EELA, (d) proposed EPR without MAP, (e) proposed EPR with MAP.

# HMDEPR

## Тестирование (текстуры)



	LA	ELA [3]	EELA [5]	EPR w/o MAP	EPR with MAP
Waves	32.32	31.29	32.05	31.85	32.72
Grass	34.45	31.49	33.31	32.86	33.45
Trees	26.33	25.73	25.95	25.91	25.90
Bricks & T.	29.18	28.64	28.96	28.67	28.94
Average	30.57	29.29	30.07	29.82	30.25

PSNR in dB

# HMDEPR

## Тестирование



# HMDEPR

## Тестирование



TABLE 4: PSNR of the deinterlacing algorithms in dB.

	Total picture number	LA	FI	2FMA	3FMA	4FTD [21]	4FHMD [5]	HMDEPR
Bus	150	28.19	20.60	26.83	25.37	26.86	25.24	26.87
Coastguard	300	28.67	27.27	28.43	29.91	29.63	30.41	29.13
Foreman	300	32.39	28.74	31.34	32.42	32.50	32.84	33.93
Hall monitor	300	31.76	36.70	32.49	39.58	35.07	36.69	38.79
Mobile	300	25.47	23.49	25.21	26.70	25.99	24.55	24.67
M&D	300	39.20	40.54	39.02	42.59	41.90	40.91	43.23
Silent	300	33.80	34.42	35.93	39.31	37.95	38.68	41.10
Stefan	300	27.46	21.01	26.59	25.82	25.83	25.56	26.88
Vectra-color	142	26.63	22.28	26.42	25.69	27.05	26.75	27.31
Average		30.40	28.34	30.25	31.93	31.42	31.29	32.43



# HMDEPR

## Выводы



- Достоинства

- Хорошо интерполирует границы
- Неплохо строит маску движения

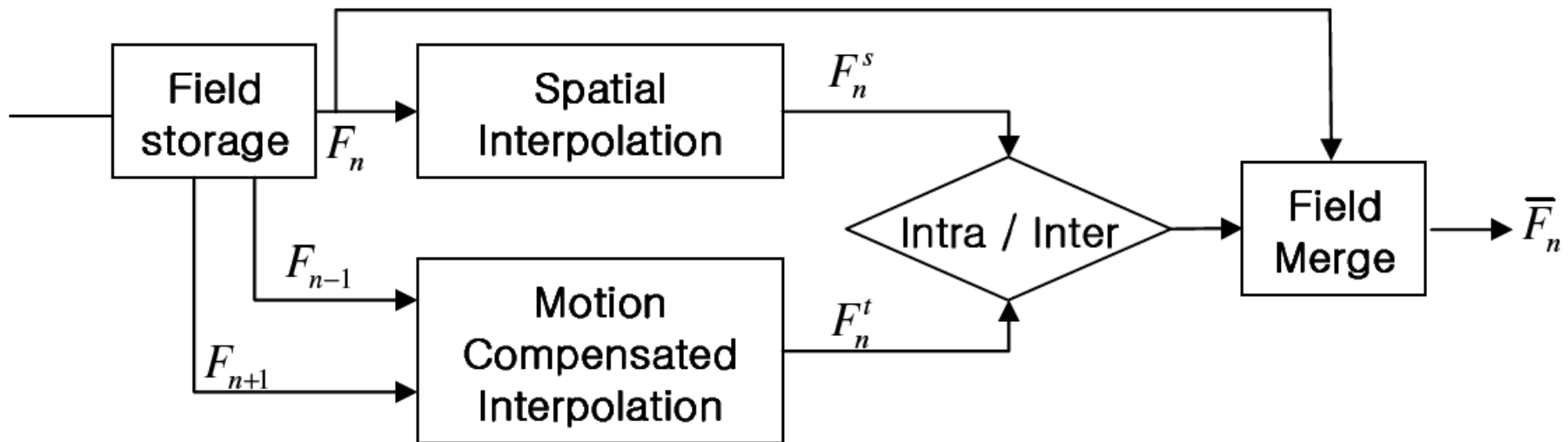
- Недостатки

- Проигрывает на текстурах

# Содержание

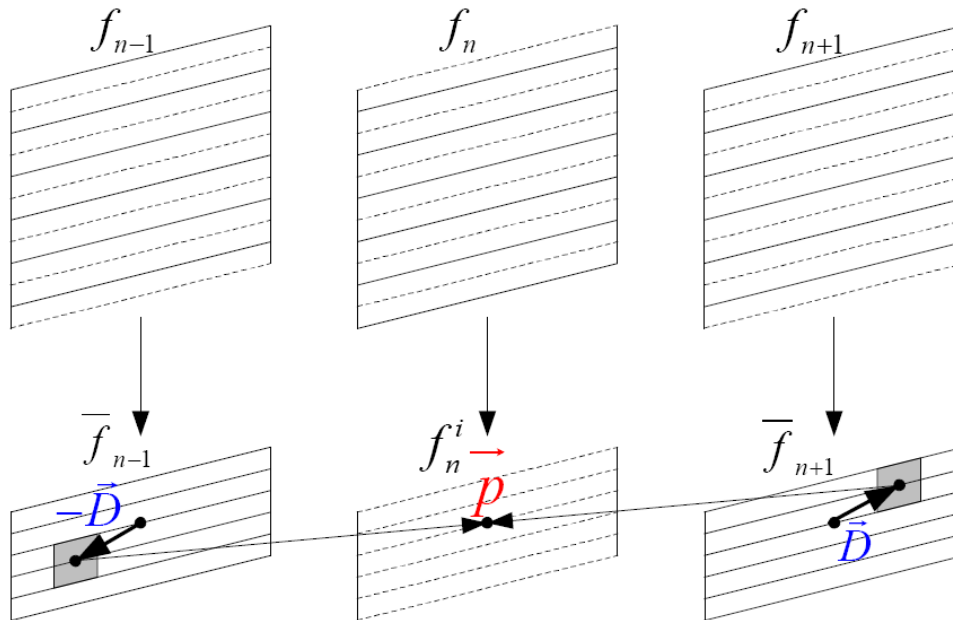
- Введение
- Классификация методов
- Простые методы
- Сложные методы
  - Motion Adaptive
  - Motion Compensation based
  - MSU filter
- Дальнейшие планы
- Список материалов

# Сложные методы MC-based Deinterlacing



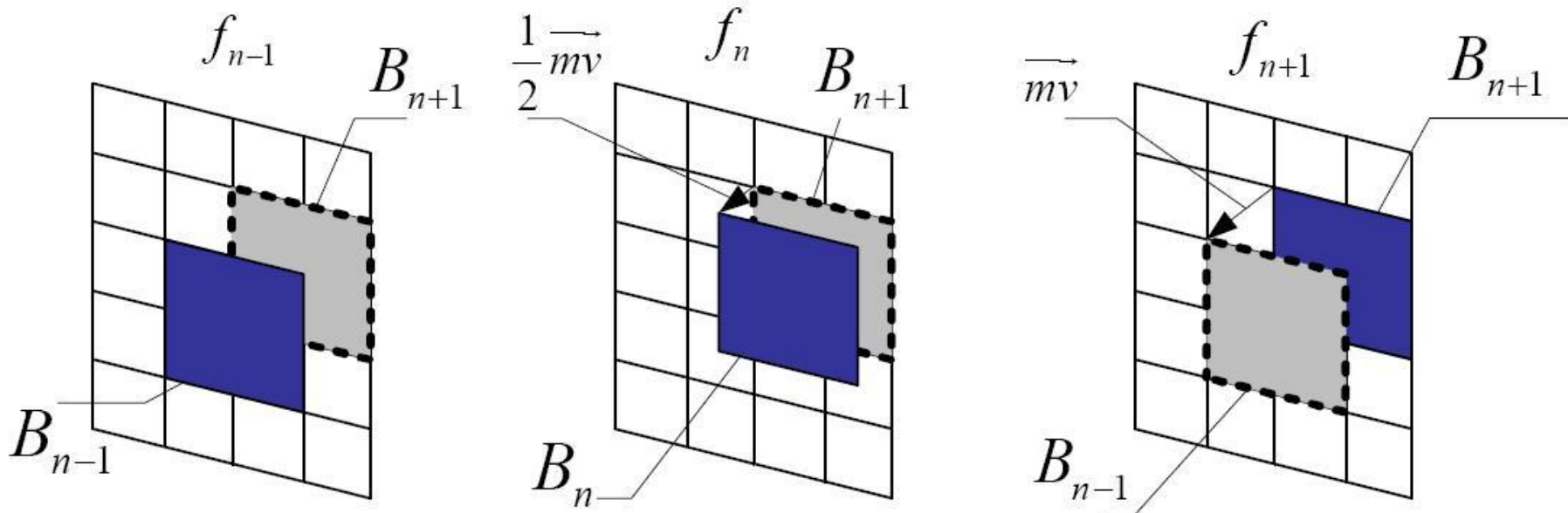
# Сложные методы MC-based Deinterlacing

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



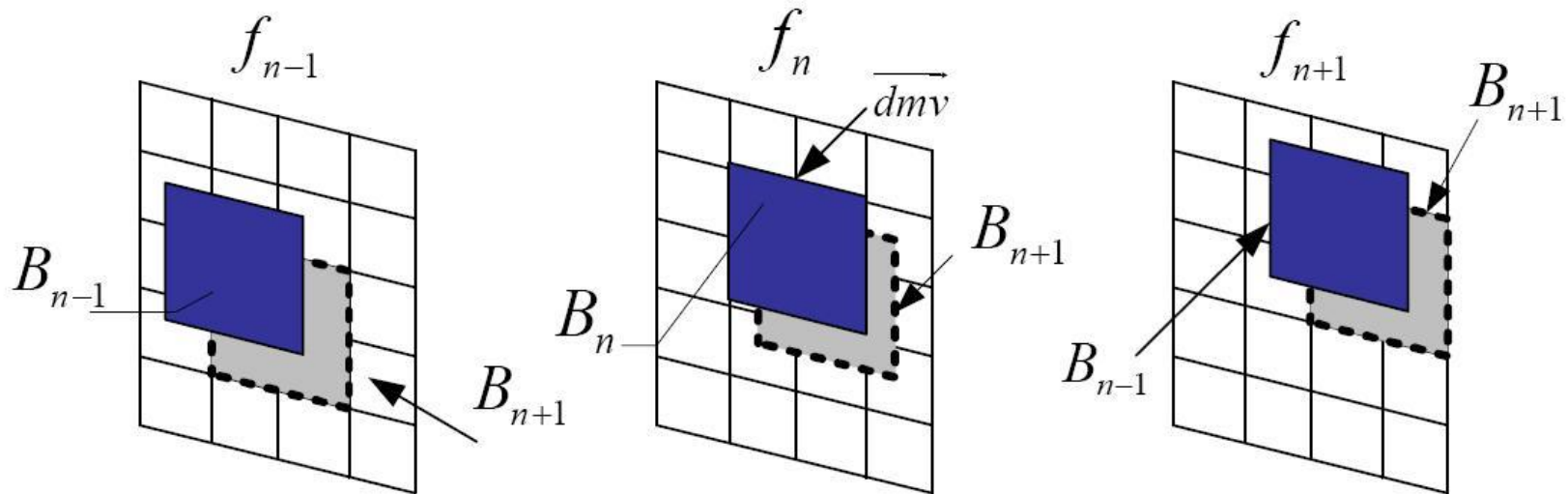
# Сложные методы MC-based Deinterlacing

Поиск похожих блоков



# Сложные методы MC-based Deinterlacing

Подгоним найденный блок  $B_n$  под сетку



$$\overline{f_n}(\overline{p} + \frac{1}{2}\overline{mv} + \overline{dmv}) = \frac{1}{2} \left[ \overline{f_{n-1}}(\overline{p} + \overline{mv} + \overline{dmv}) + \overline{f_{n+1}}(\overline{p} + \overline{dmv}) \right]$$

# MC-based Deinterlacing

## Пример работы

Field Insertion



MC-based



# Содержание

- Введение
- Классификация методов
- Простые методы
- Сложные методы
  - Motion Adaptive
  - Motion Compensation based
  - MSU filter
- Дальнейшие планы
- Список материалов



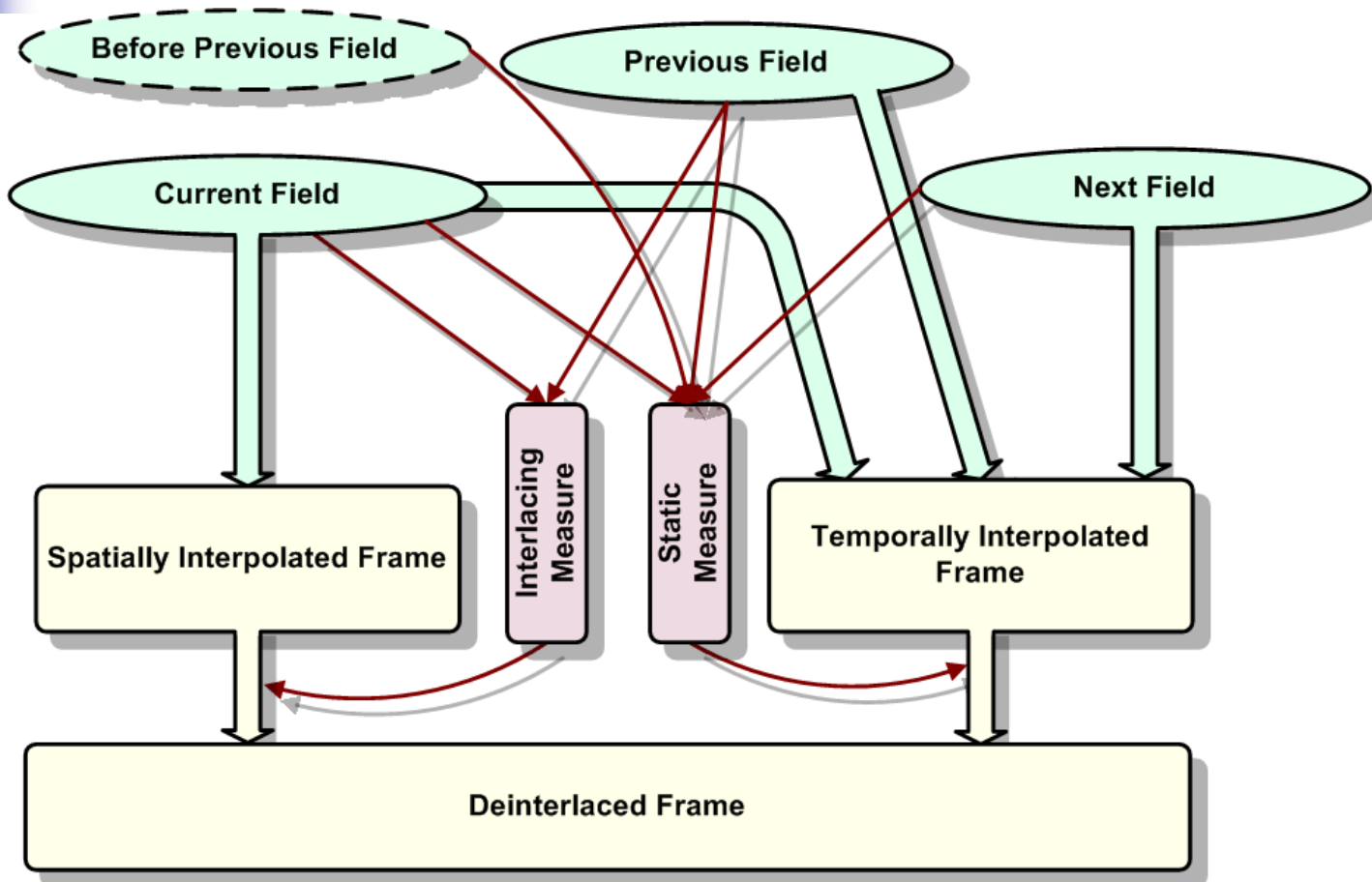


# MSU Filter

---

- MSU MA
  - Temporal Interpolation
  - Spatial Interpolation
- MSU MC
- Тестирование

# MSU MA



# MSU MA

- Пиксель интерполируется двумя способами
  - Пространственная интерполяция (Spatial)
  - Временная интерполяция (Temporal)
- Каждый из двух получившихся пикселей вносит свой вклад в конечный результат с определенными весами, зависящим от характера движения

$$p_0(x, y, k) = \begin{cases} p(x, y, k), & (y + k) \% 2 = 0, \\ \hat{p}_s(x, y, k) * SpatCoef + \hat{p}_t(x, y, k) * TempCoef, & \text{иначе,} \end{cases}$$

*SpatCoef* – пространственный весовой коэффициент

*TempCoef* – временной весовой коэффициент

$\hat{p}_s(x, y, k)$  – интерполированный пространственным способом пиксел

$\hat{p}_t(x, y, k)$  – интерполированный временным способом пиксел

## Temporal Interpolation

- В качестве временного приближения используется **Field Insertion** или **Field Averaging**
- Весовая функция для временного приближения ("Static" weighting function) задается так:

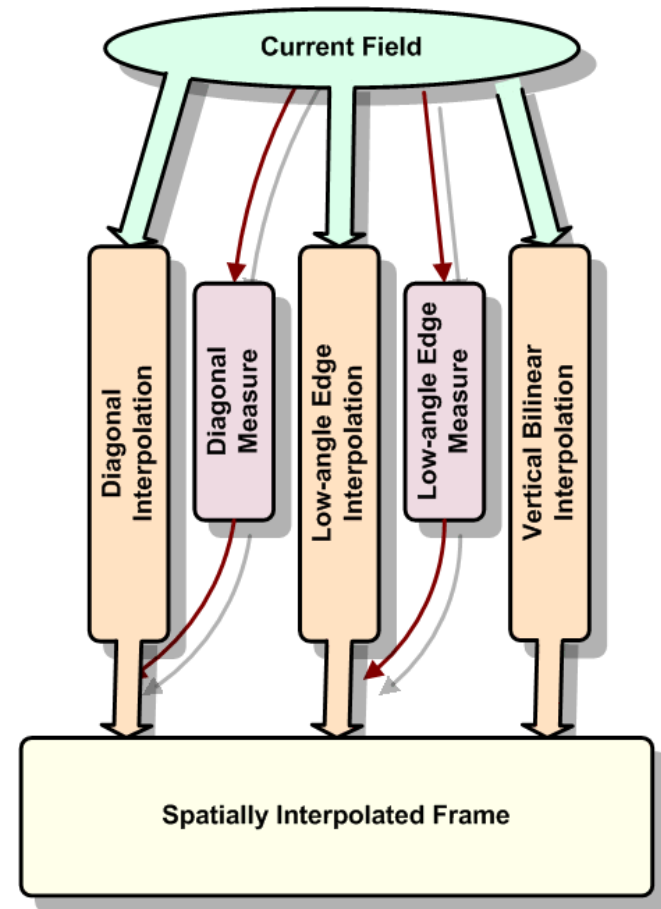
$$St = \frac{5}{3E + 5} * \frac{D}{D + 150} = f(E) * h(D)$$

E - so called SAD (Sum of Absolute Differences). Так называемая сумма абсолютных разностей, рассчитанная для полей n+1, n, n-1, n-2.

D - дисперсия текущего поля. Считается по восьми соседним пикселям.

- *St* влияет на *TempCoef*

- Интерполяция
  - **Vertical Bilinear** – позволяет избежать эффект Гиббса (рингинг)
  - **Diagonal** – позволяет избежать ступенчатые границы
  - **Low-angle Edge** – интерполирует границы с маленькими углами наклона
- Пиксели, получившиеся разными способами берем с определенным весом



$$X = \begin{cases} X, & \text{четность поля и строки совпадают,} \\ W_{diag} * X_{diag} + W_{low} * X_{low} + W_{bilinear} * X_{bilinear}, & \text{иначе,} \end{cases}$$

$X_{diag}$  - пиксел, интерполированный Diagonal Interpolation

$X_{low}$  - пиксел, интерполированный Low - angle Edge Interpolation

$X_{bilinear}$  - пиксел, интерполированный Bilinear Interpolation

$W_{diag}$  - вес, соответствующий  $X_{diag}$

$W_{low}$  - вес, соответствующий  $X_{low}$

$W_{bilinear}$  - вес, соответствующий  $X_{bilinear}$

- Весовая функция, влияющая на *SpatCoef*
- Вес больше, когда присутствует «зубчатость»

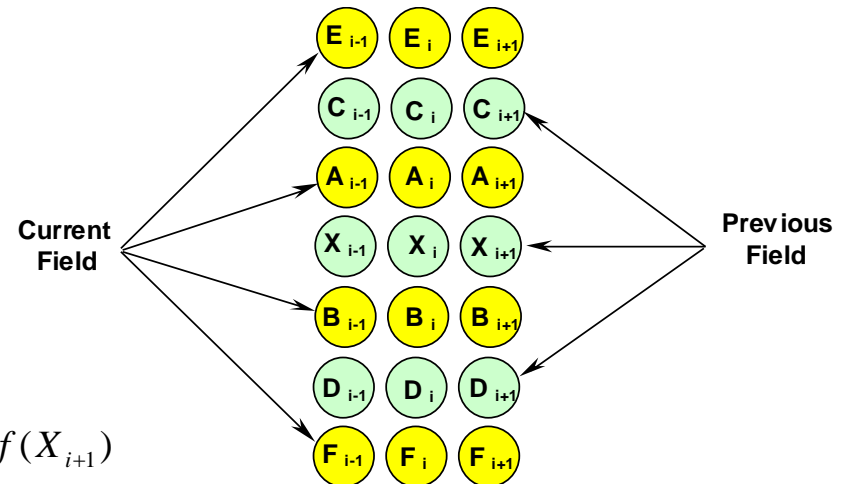
$$m_1 = (A_i - \frac{C_i + X_i}{2})$$

$$m_2 = (X_i - \frac{A_i + B_i}{2})$$

$$m_3 = (B_i - \frac{X_i + D_i}{2})$$

$$Dif(X_i) = |m_1 - m_2 + m_3|$$

$$Interlacing\_measure(X_i) = Dif(X_{i-1}) + 2Dif(X_i) + Dif(X_{i+1})$$





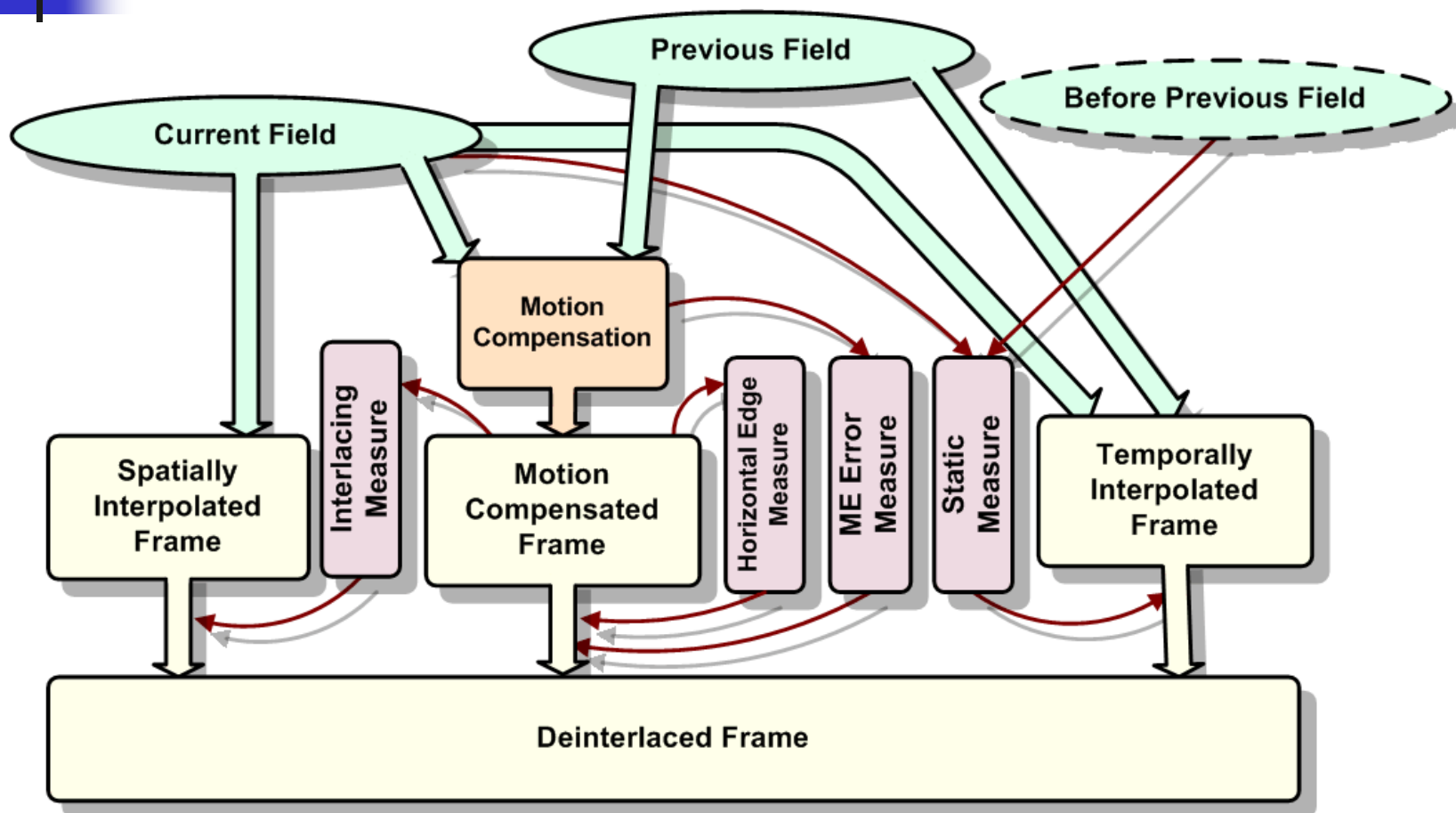
# MSU Filter

---

- MSU MA
- MSU MC
- Тестирование



# MSU MC



# MSU MC

- Пиксель интерполируется тремя способами
  - Пространственная интерполяция (Spatial)
  - Временная интерполяция (Temporal)
  - MC-интерполяция
- Веса *SpatCoef* и *TempCoef* определяются похожим (как в MSU MA) образом

$$p_0(x, y, k) = \begin{cases} p(x, y, k), & (y + k) \% 2 = 0, \\ \hat{p}_s(x, y, k) * \text{SpatCoef} + \hat{p}_t(x, y, k) * \text{TempCoef} + \hat{p}_{mc}(x, y, k) * \text{MCCoef}, & \text{иначе,} \end{cases}$$

*SpatCoef* – пространственный весовой коэффициент

*TempCoef* – временной весовой коэффициент

*MCCoef* – весовой коэффициент MC

$\hat{p}_s(x, y, k)$  – интерполированный пространственным способом пиксел

$\hat{p}_t(x, y, k)$  – интерполированный временным способом пиксел

$\hat{p}_i(x, y, k)$  – интерполированный MC- способом пиксел

# MSU MC

- Мера ошибки ME (ME Error Measure)
  - Влияет на вес MC кадра.
  - Определяет отсутствие движения и слабое движение.
  - Может обнаружить только объекты с большой дисперсией.
  - Зависит от:
    - различия блоков (SAD) текущего и предыдущего полей – B.
    - различия 6 соседей (SAD) предыдущего и текущего полей – L.
    - дисперсии - D. Чем больше дисперсия, тем меньше вероятность движения.

$$W = \frac{1}{\text{probability\_of\_motion}} = \frac{D}{L^2 + 1} * \frac{B}{B + L + 1}$$

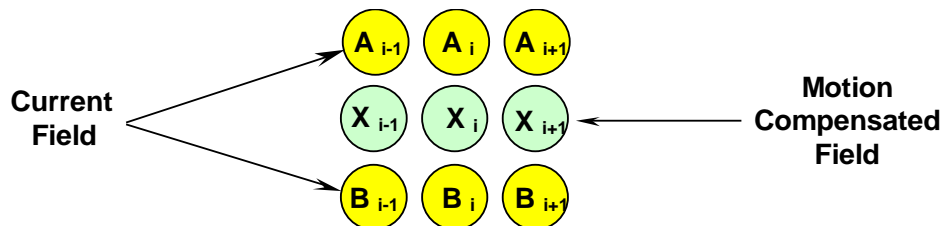
# MSU MC

- Весовая функция горизонтальных границ (Horizontal Edge Detection weighting function)
  - Влияет на вес MC-кадра
  - Нужна для более точно интерполяции границ, близких к горизонтальным.
  - Пиксель принадлежит горизонтальной границе, если выполнено два условия:
    - Градиент по вертикали больше определенного порога (для области этого пикселя).
    - Значение пикселя лежит между значениями соседей.

$$Gradient(X_i) = |(A_{i-1} + 2A_i + A_{i+1}) - (B_{i-1} + 2B_i + B_{i+1})|$$

$$Match(X_i) = \min \{ \max \{ A_i, B_i \} - X_i, X_i - \min \{ A_i, B_i \} \}$$

$$W(X_i) = Gradient(X_i) * \left( 1 + \frac{2Match(X_i)}{\max \{ A_i, B_i \} - \min \{ A_i, B_i \} + 1} \right)$$





# MSU Filter

---

- MSU MA
- MSU MC
- Тестирование

# MSU Filter

## Тестирование



Original frame

# MSU Filter

## Тестирование

VirtualDub (internal)



AlparySoft Deinterlace



Smart Deinterlace

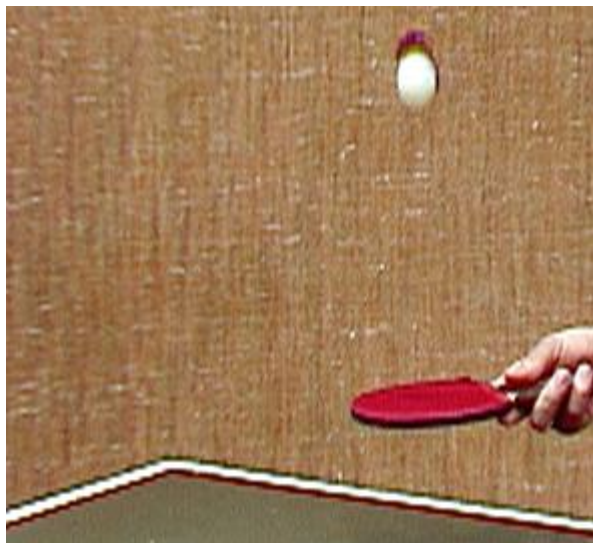


De Haan Deinterlacer



# MSU Filter

## Тестирование



Smart Deinterlace



MSU Deinterlacer (motion adaptive)



De Haan Deinterlacer



# MSU Filter

## Тестирование



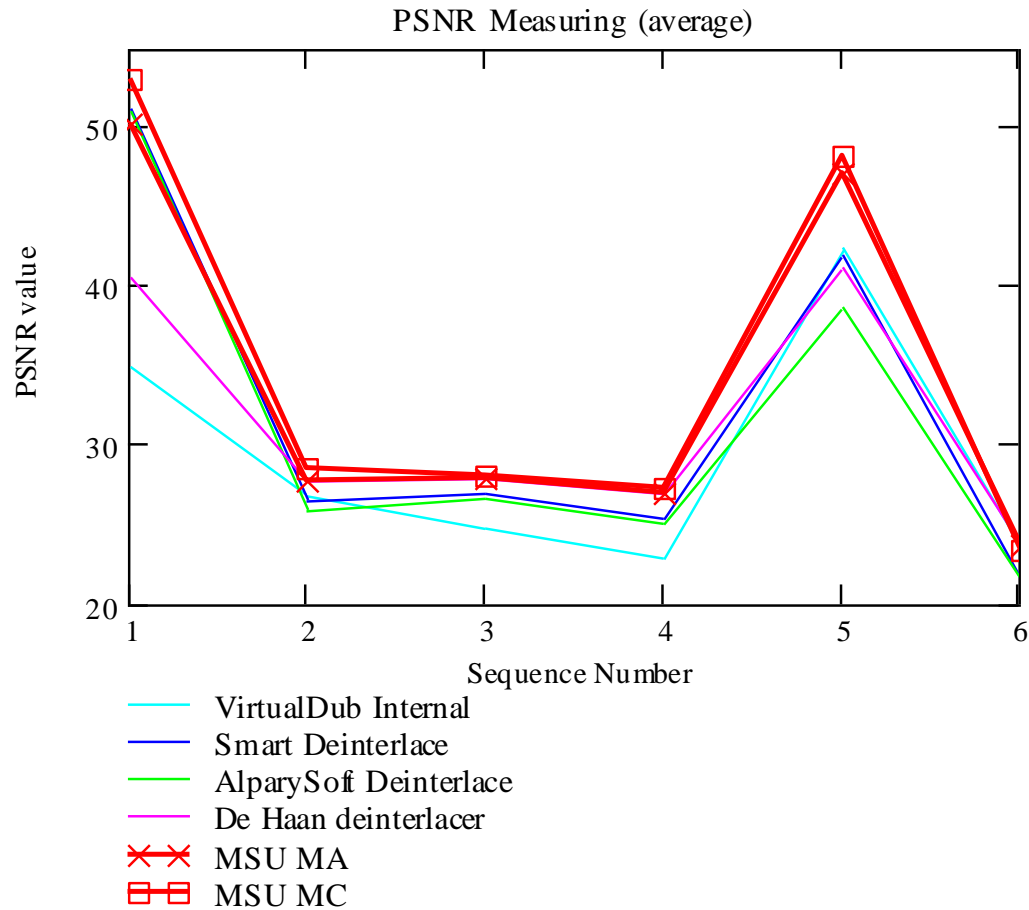
MSU Deinterlacer (motion adaptive)



MSU Deinterlacer (motion compensation)

# MSU Filter

## Тестирование





# Содержание

---

- Введение
- Классификация методов
- Простые методы
- Сложные методы
- Дальнейшие планы
- Список материалов

# Дальнейшие планы

- Опробовать изложенные методы
- Совместить следующие подходы
  - Motion Adaptive
    - Spatial
    - Temporal
    - Edge-based
  - Motion Compensation based
  - Возможно, вероятностный подход
  - Анализ области и весовые функции



# Содержание

---

- Введение
- Классификация методов
- Простые методы
- Сложные методы
- Дальнейшие планы
- Список материалов

# Список материалов

1. A De-interlacing Algorithm Based on True Motion Vectors. *Yi-Shan Shiu*, Institute of Computer & Communication, 2006 .
2. Video De-Interlacing by Adaptive 4-Field Global/Local Motion Compensated Approach. *Yu-Lin Chang, Shyh-Feng Lin, Ching-Yeh Chen, Liang-Gee Chen*, IEEE Transactions on Circuits and Systems for Video Technology, Volume 15, Issue 12, pp.1569 - 1582, Dec. 2005.
3. High Performance De-Interlacing Algorithm for Digital Television Displays. *Li G.-L., Chen M.-J.* Journal of Display Technology, Volume 2, Issue 1, pp.85 - 90, March 2006.
4. Motion compensated frame interpolation by new block-based motion estimation algorithm. *Taehyeun Ha, Seongjoo Lee, Jaeseok Kim*. IEEE Transactions on Consumer Electronics, Volume 50, Issue 2, pp.752 - 759, May 2004.
5. Advanced de-interlacing techniques. *E.B. Bellers and G. de Haan*.
6. Direction-Oriented Interpolation and Its Application to De-interlacing. *Hoon Yoo and Jechang Jeong*. IEEE Transactions on Consumer Electronics, Volume 48, Issue 4, pp.954 – 962, Nov. 2002.
7. A recursive approach for de-interlacing using improved ELA and motion compensation based on bi-directional BMA. *Seungchan Byun, Jeongmonn Byun, Gyeonghwan Kim*. 2004 International Conference on Image Processing, Volume 3, 24-27, pp.1679 – 1682, Oct. 2004.
8. A Motion-Adaptive Deinterlacer via Hybrid Motion Detection and Edge-Pattern Recognition. *Gwo Giun Lee, Ming-Jiun Wang, Hsin-Te Li, He-Yuan Lin*. EURASIP Journal on Image and Video Processing, 2008.
9. Презентации Видеогруппы. *Сергей Путилин, Денис Кубасов*. 2003, 2004.
10. Отчет по MSU Deinterlacer.



# Вопросы

---



# Содержание

- Введение
- Классификация методов
- Простые методы
- Сложные методы
  - Motion Adaptive
  - Motion Compensation based
  - MSU filter
  - Вероятностный подход
- Дальнейшие планы
- Список материалов



# Сложные методы Вероятностный подход



Максимальная апостериорная гипотеза (MAP):

$u_0$  – текущее поле (четные или нечетные строки)

$u$  – кадр на выходе, соответствующий  $u_0$

$\Omega_0$  – область, соответствующая  $u_0$

$\Omega$  – область, соответствующая  $u$

$P_3(u_0) - \text{const}$

$$P(u|u_0) = \frac{P_1(u_0|u) P_2(u)}{P_3(u_0)}.$$

$$P(u|u_0) \propto P_1(u_0|u) \cdot P_2(u).$$

Оценка максимального правдоподобия:

$$u = \arg \max_u \{ \ln P_1(u_0|u) + \ln P_2(u) \}.$$

# Сложные методы

## Вероятностный подход



$P_1(u_0|u)$  - «похожесть»  $u$  и  $u_0$  на области  $\Omega_0$ . По сути, плотность распределения шума.

$$P_1(u_0|u) = f_n(u - u_0) \propto \exp \left\{ -\frac{\lambda \int_{\Omega} (u - u_0)^2 dx dy}{2} \right\}, \quad \lambda = \begin{cases} 1 & \delta \in \Omega_0, \\ 0 & \text{other.} \end{cases}$$

$\delta$  - позиция каждого пикселя кадра

Существующие пиксели ( $u_0$ ) желательно оставить без изменений

# Сложные методы

## Вероятностный подход



$P_2(u)$  - априорная оценка

Модель Гиббса:

$$P_2(u) \propto \exp(-\alpha E[u]),$$

$E[u]$  - функция энергии

$\alpha$  - обратная абсолютная температура

$$E[u] = \frac{1}{2} \int_{\Omega} \varphi(|\nabla_3 u|) dx dy,$$

$\nabla_3 = (\partial_x, \partial_y, \partial_t)^T$  - пространственно-временной градиент

$\varphi: R \rightarrow R$

$$P_2(u) \propto \exp \left\{ -\frac{\alpha}{2} \left[ \int_{\Omega} \varphi(|\nabla_3 u|) dx dy \right] \right\}.$$

# Сложные методы

## Вероятностный подход



$$\varphi(t) = \frac{t^2}{1+t^2}, \quad \frac{\varphi'(t)}{2t} = \frac{1}{(1+t^2)^2}$$

Таким образом, задача сводится к минимизации функционала энергии:

$$u = \arg \min_u J_{TV}(u).$$

$$J_{TV}(u) = \frac{\lambda}{2} \int_{\Omega} (u - u_0)^2 dx dy + \frac{\alpha}{2} \int_{\Omega} \varphi(|\nabla_3 u|) dx dy.$$

# Сложные методы

## Вероятностный подход



Соответствующее уравнение Эйлера-Лагранжа:

$$\lambda(u - u_0) - \alpha \operatorname{div} \left( \frac{\varphi'(|\nabla_3 u|)}{|\nabla_3 u|} \nabla u \right) = 0,$$
$$\left. \frac{\partial u}{\partial n} \right|_{\partial\Omega} = 0,$$

$n$  - нормаль вектора границы  $\partial\Omega$  области  $\Omega$

# Сложные методы Вероятностный подход



Стандартное видео (без шума)

**Table 1** Comparison of the proposed scheme and the conventional methods ( $\alpha = 0.0001$ )

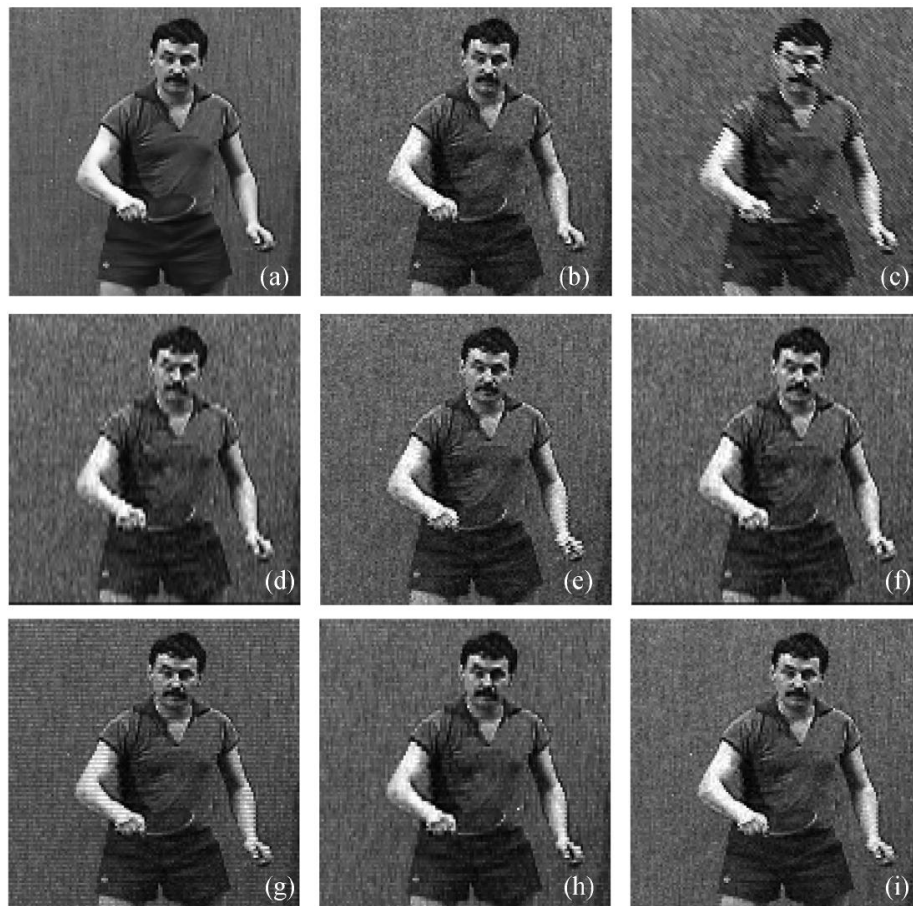
No.	Technique	Flowers sequence PSNR (dB)	TennisA sequence PSNR (dB)	Bicycle sequence PSNR (dB)	Cact sequence PSNR (dB)
1	LR	20.79	22.87	24.14	25.92
2	LA	22.27	23.69	31.38	31.19
3	FI	19.04	21.68	22.75	29.41
4	VT	21.95	23.76	28.94	31.66
5	MCmed	22.39	24.13	25.97	26.53
6	McVT	23.25	24.54	30.58	33.76
7	The proposed	25.06	27.70	28.77	35.43

# Вероятностный подход тестирование



Стандартное видео (с шумом)

- (a) original video
- (b) video with noise (zero-mean, SNR=20 dB)
- (c) LR (PSNR=22.10 dB)
- (d) LA (PSNR=24.80 dB)
- (e) FI (PSNR=21.78 dB)
- (f) VT (PSNR = 22.68)
- (g) MCmed (PSNR=24.11 dB)
- (h) McVT (PSNR=25.96 dB)
- (i) Proposed (PSNR=27.81dB)



# Вероятностный подход тестирование



Стандартное видео (с шумом)

**Table 2** Comparison of the proposed scheme and the conventional methods with noisy sequences ( $\alpha = 0.0001$ )

No.	Technique	SNR=10 dB		SNR=20 dB		SNR=30 dB		SNR=40 dB	
		Sequence 5 PSNR (dB)	Sequence 6 PSNR (dB)	Sequence 5 PSNR (dB)	Sequence 6 PSNR (dB)	Sequence 5 PSNR (dB)	Sequence 6 PSNR (dB)	Sequence 5 PSNR (dB)	Sequence 6 PSNR (dB)
1	LR	14.41	16.15	15.75	22.10	15.94	23.68	15.95	23.88
2	LA	17.02	17.70	19.32	24.80	19.69	27.25	19.72	27.59
3	FI	16.36	16.09	19.30	21.78	19.88	23.23	19.43	23.41
4	VT	16.86	16.75	20.03	23.68	20.66	25.99	20.73	26.30
5	MCmed	20.29	20.72	21.84	24.11	22.10	24.67	22.13	24.74
6	McVT	18.27	19.21	20.36	25.96	20.67	27.99	22.74	28.27
7	The proposed	22.24	22.14	25.34	27.81	25.91	29.25	25.96	30.73

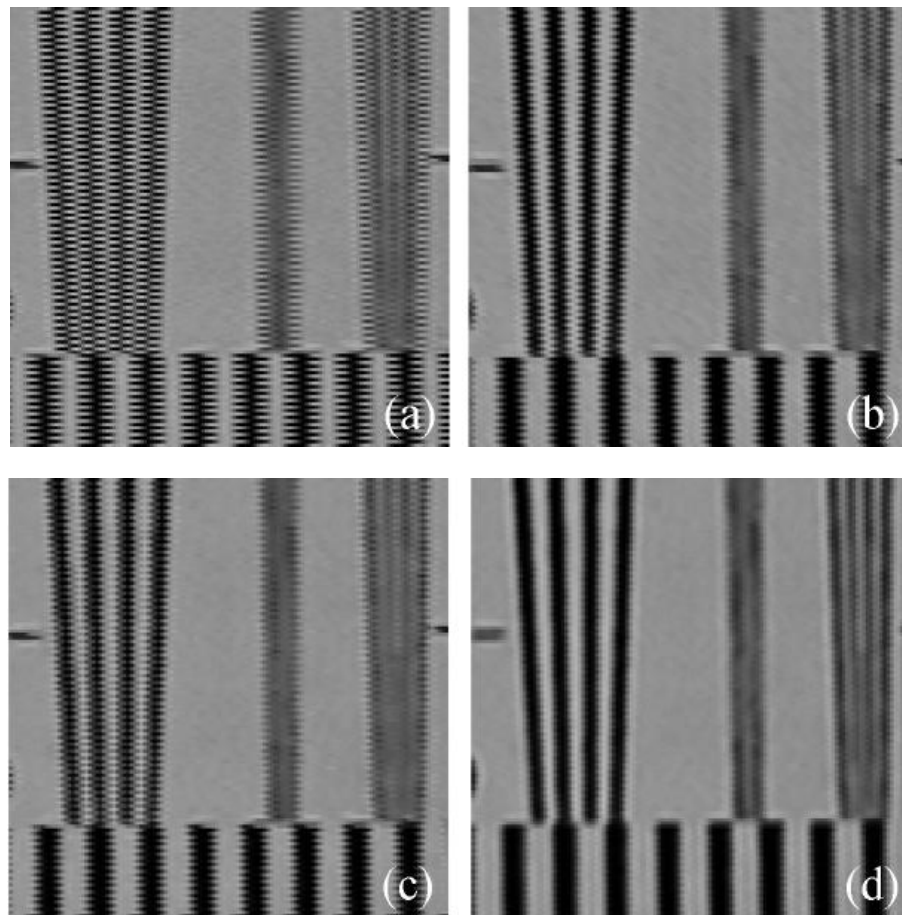


# Вероятностный подход тестирование



Реальное видео

- (a) original video
- (b) LR
- (c) LA
- (d) Proposed



# Motion Adaptive Motion Detector



$$CIP(\vec{x}, n) = \begin{cases} S, & \frac{1}{7}(D_T(\vec{x}, n) + D_S(\vec{x}, n)) < threshold \\ M, & otherwise \end{cases}$$

$$D_T(\vec{x}, n) = \sum_{k=1}^6 \left| f_n(\vec{x} + \sigma(k)) - f_{n-2}(\vec{x} + \sigma(k)) \right|$$

$$D_S(\vec{x}, n) = \left| f_{n-1}(\vec{x} + \alpha) - f_{n-1}(\vec{x} + \beta) \right|$$

$\sigma(1):(-1,-1)$     $\sigma(2):(0,-1)$

$\sigma(3):(+1,-1)$     $\sigma(4):(-1,+1)$

$\sigma(5):(0,+1)$     $\sigma(6):(+1,+1)$

$\alpha:(-1,0)$     $\beta:(+1,0)$

$D_T(\vec{x}, n)$  - различие во времени

$D_S(\vec{x}, n)$  - различие в пространстве