



Современные методы деинтерлейсинга

Петров Александр

Video Group
CS MSU Graphics & Media Lab



Содержание

- Введение
- Классификация методов
- Пространственные методы
- Motion Adaptive
- Motion Compensation based
- Сравнение
- Текущие результаты и дальнейшие планы

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).

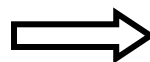


Введение

Поле #5



Поле #6



Кадр #3



Введение

Пример



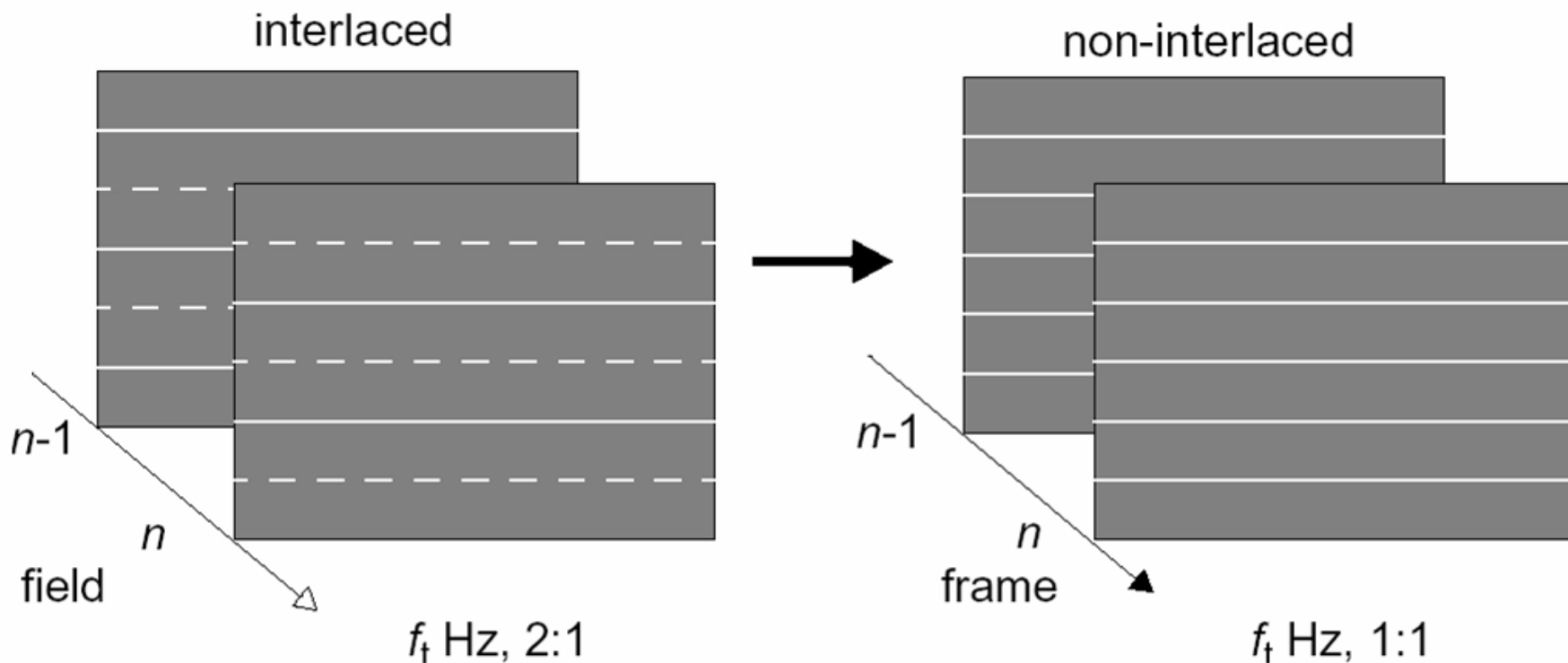
Введение

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

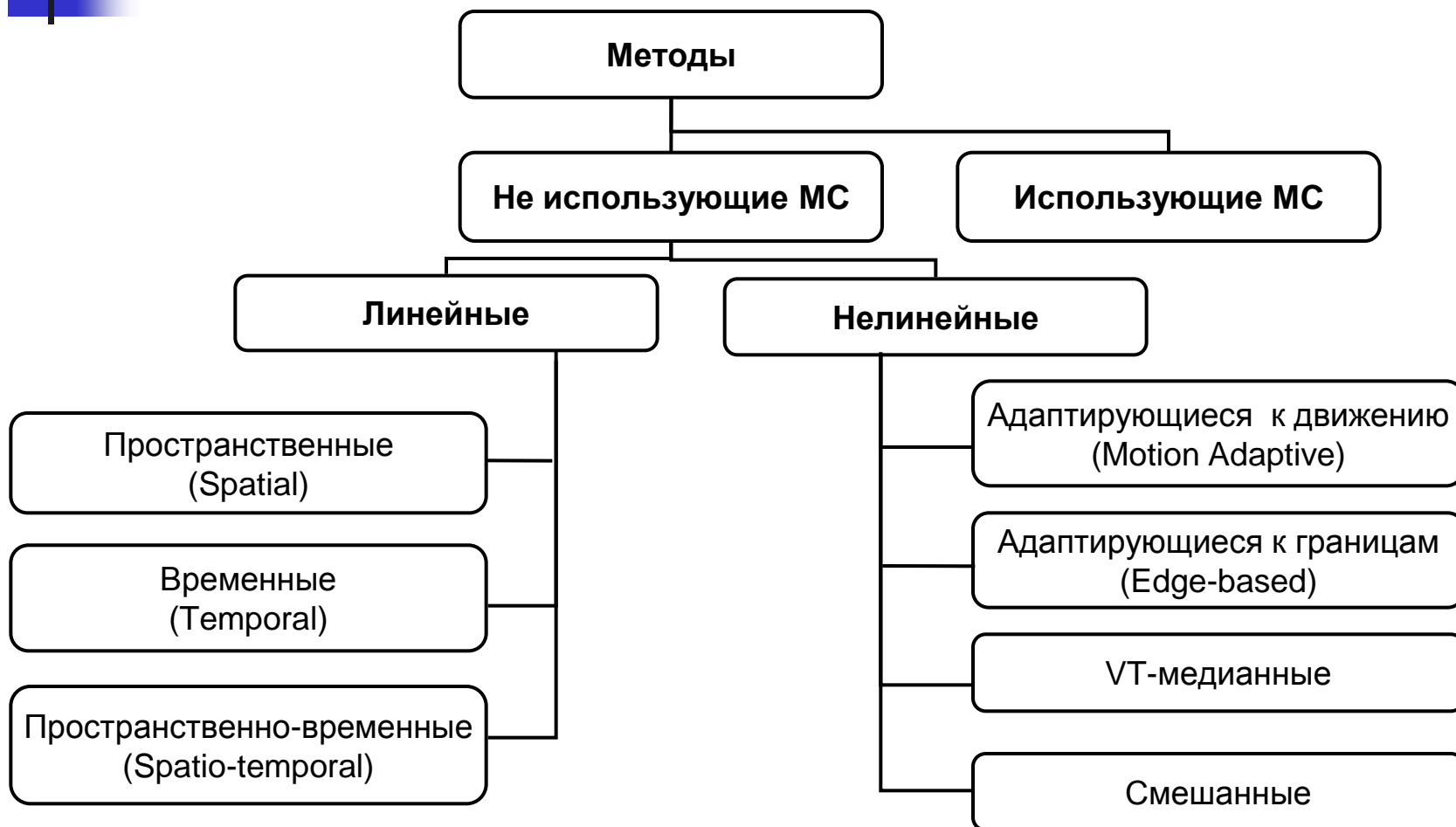
$p_o(i, j, k)$ - результат



Содержание

- Введение
- Классификация методов
- Пространственные методы
- Motion Adaptive
- Motion Compensation based
- Сравнение
- Текущие результаты и дальнейшие планы

Классификация методов



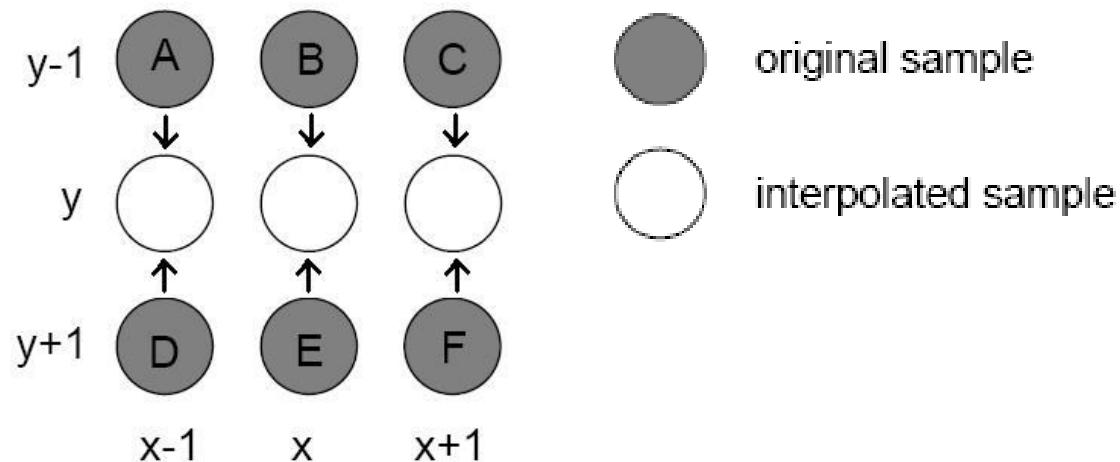
Содержание

- Введение
- Классификация методов
- Пространственные методы
 - Bob
 - EEDI2
 - Сравнение
- Motion Adaptive
- Motion Compensation based
- Сравнение
- Текущие результаты и дальнейшие планы

Spatial Bob

Усреднение строк (LA)

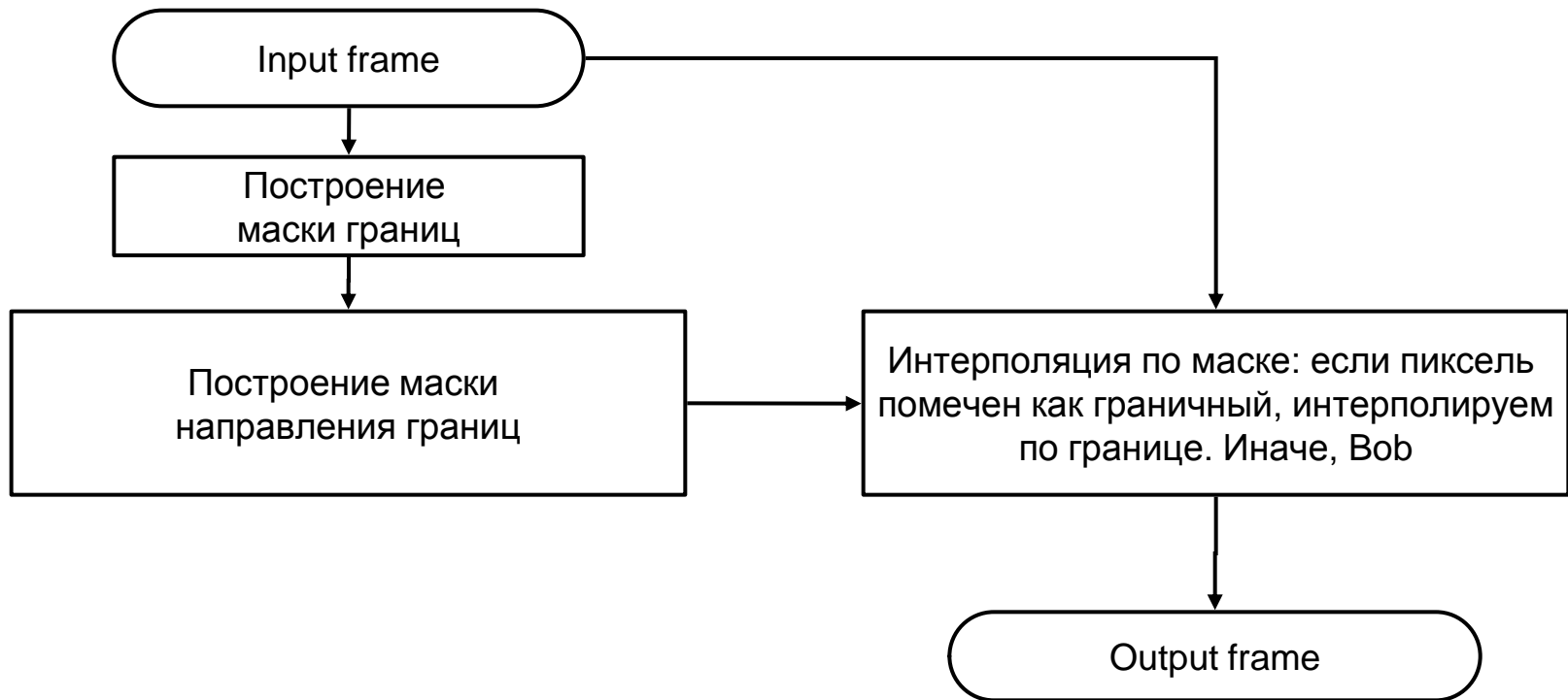
$$p_0(x, y, k) = \begin{cases} p(x, y, k), & (y + k) \% 2 = 0, \\ \frac{1}{2} [p(x, y - 1, k) + p(x, y + 1, k)], & \text{иначе,} \end{cases}$$



Spatial EEDI2



Схема работы



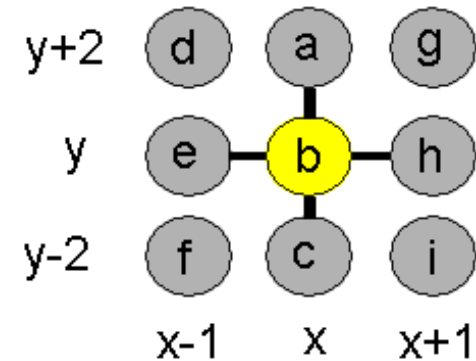
Spatial EEDI2



Построение маски границ

if ($|c-b| < 10$ and $|a-b| < 10$ and $|a-c| < 10$) OR
 $(|e-f| < 10$ and $|d-e| < 10$ and $|d-f| < 10$ and
 $|h-i| < 10$ and $|g-h| < 10$ and $|g-i| < 10$)

then $b \notin \text{Edge}$



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

$$Ix = h - e$$

$$Iy = \max \begin{pmatrix} |a - c|, \\ |b - c|, \\ |a - b| \end{pmatrix}$$

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

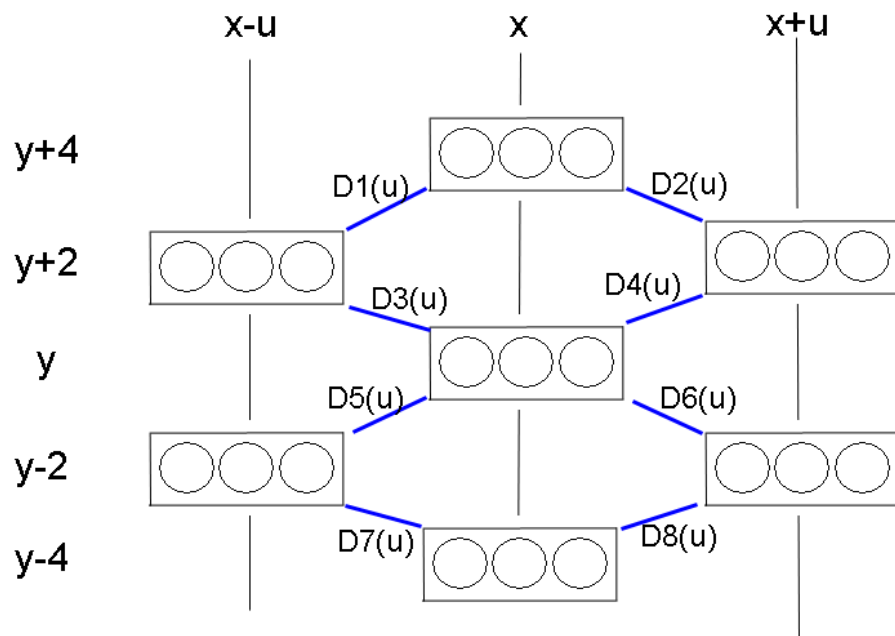
$$Iyy = (c - b) + (a - b)$$

$disp$ – Дисперсия

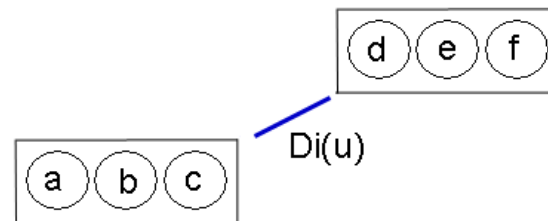
Spatial EEDI2

Определение направления границы

рассматриваются пиксели, отмеченные как граничные



где $D_i(u) = |a - d| + |b - e| + |c - f|$



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

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

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

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

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

Spatial EEDI2

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

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

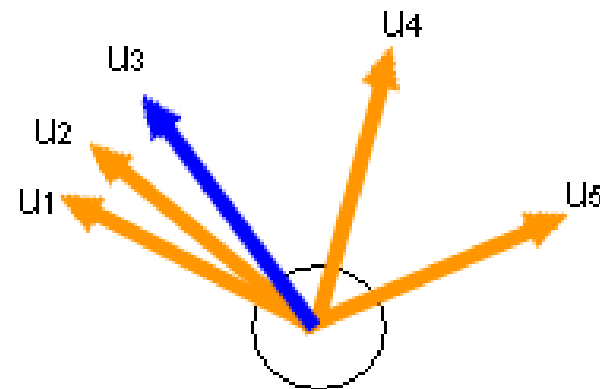
$$u_{left} : D_{left}(u_{left}) = \min_u D_{left}(u)$$

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

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

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

Пример:



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

По порогу выделяем наиболее сонаправленные с медианой вектора.
Для примера, это u_1 , u_2 и u_3

Spatial EEDI2



Edge map



Edge direction map

Spatial EEDI2



- Выводы
 - Неплохо интерполирует границы.
 - Может использоваться как ресайзер, хотя был разработан специально для деинтерлейсинга.
 - Медленный
 - Обычно используется как составляющая более сложных методов (Yadifmod, TDeint, MCBob, и т.д.)

Spatial Сравнение

Было



Spatial Сравнение

Bob



Spatial Сравнение

EEDI2



Spatial Сравнение

NNEDI

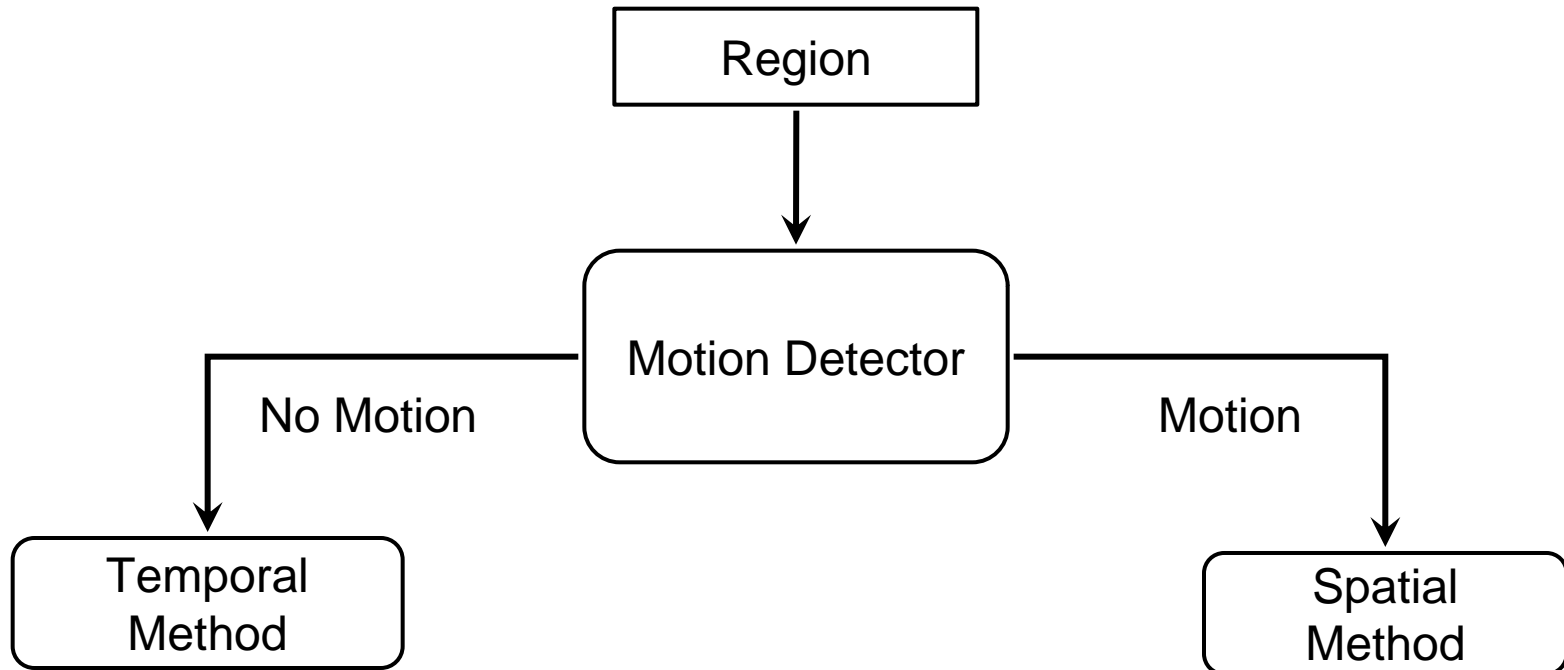


Содержание

- Введение
- Классификация методов
- Пространственные методы
- **Motion Adaptive**
 - Схема
 - Yadif
- Motion Compensation based
- Сравнение
- Текущие результаты и дальнейшие планы

Motion Adaptive

Cxema

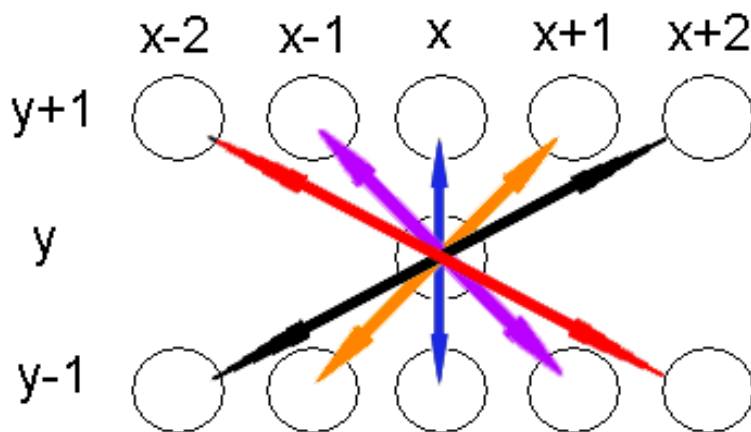


МА методы

Yadif



- Пространственная интерполяция
 - Интерполяция вдоль границ (ELA)



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

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

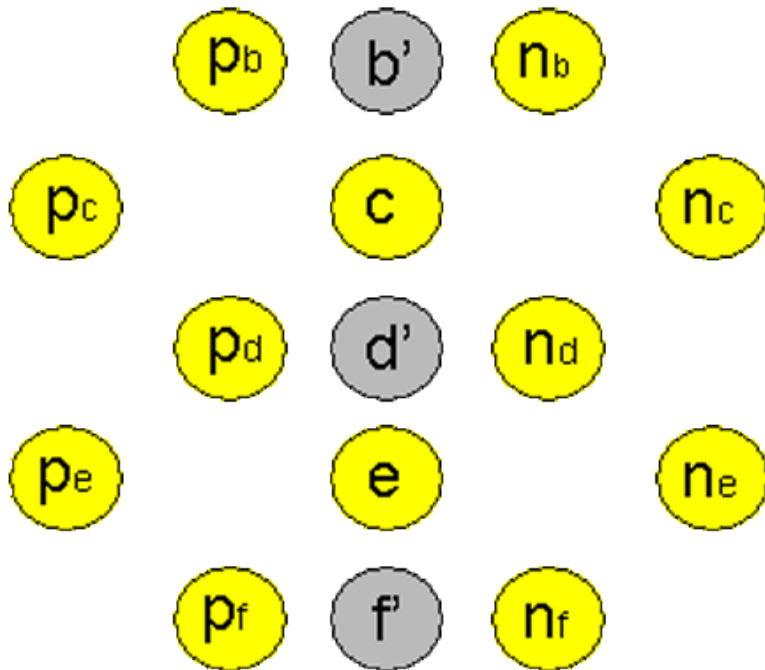
$$f_n(x, y) = \frac{f_n(x+u, y-1) + f_n(x-u, y+1)}{2}$$

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

МА методы Yadif



- Временная проверка



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

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

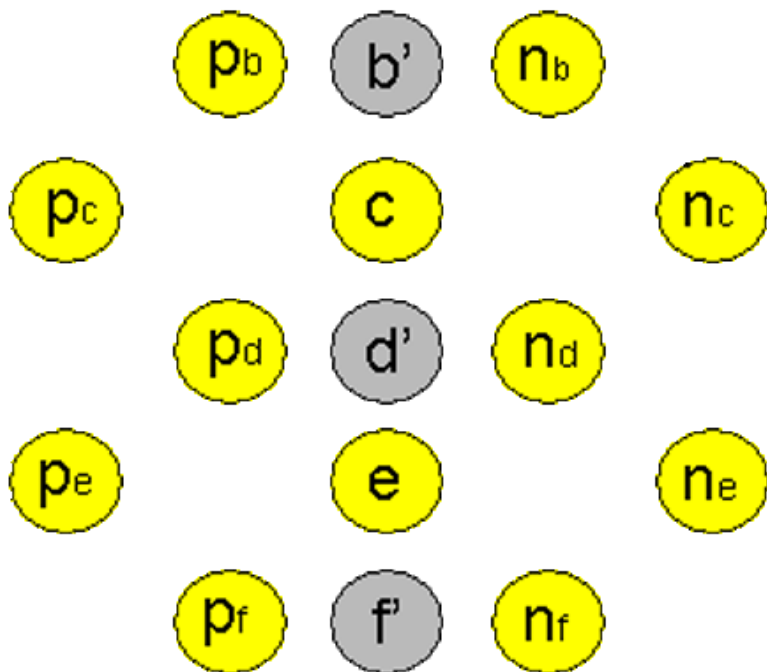
$$\Delta T_1 = |p - n|$$

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

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

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

- Пространственная проверка



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

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

$$\Delta S = \max(\Delta S \min, -\Delta S \max)$$

МА МЕТОДЫ

Yadif

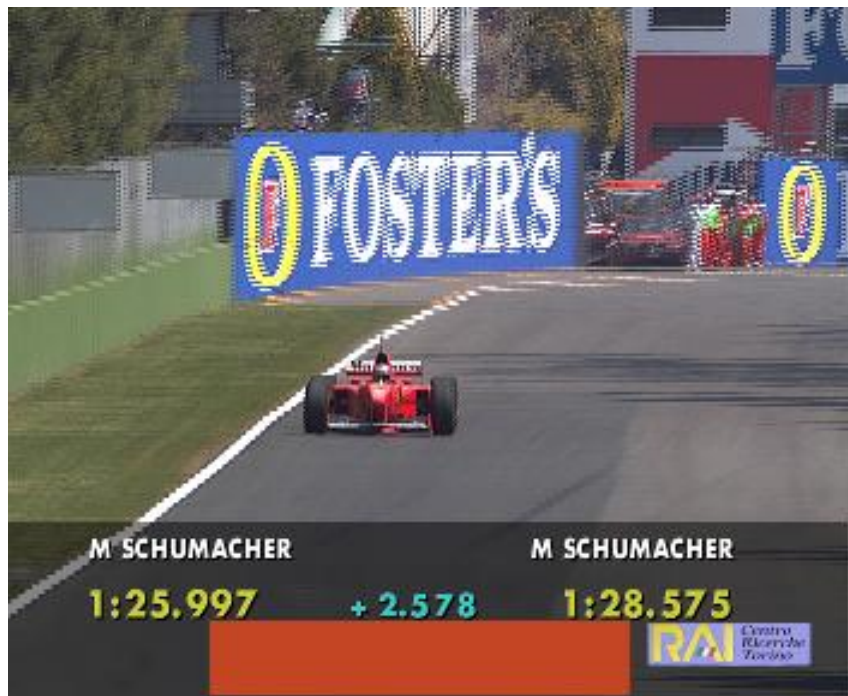


$$diff = \max(\Delta T, \Delta S)$$

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

МА методы

Yadif



Было



Стало

МА методы

Yadif



- Выводы

- Очень быстрый. Может использоваться Real-time.
- Плохо строится маска детекта движение.
- Не интерполирует границы, близкие к горизонтальным.

МА методы

Другие



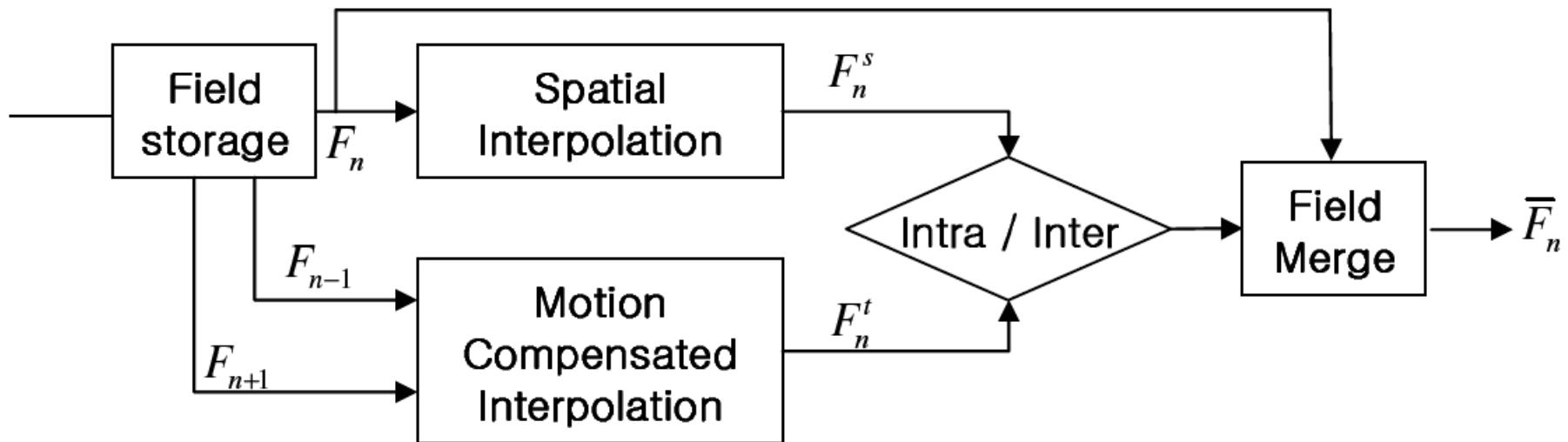
Другие популярные фильтры

- Yadifmod
 - Работает также как Yadif, но для пространственной интерполяции используется EEDI2 или NNEDI.
 - Существенно лучший результат, но также и большая потеря в скорости.
- TDeint
 - Для пространственной интерполяции может использоваться как EEDI2, NNEDI, так один из внутренних.
 - В зависимости от характера движения определяется способ временной интерполяции.

Содержание

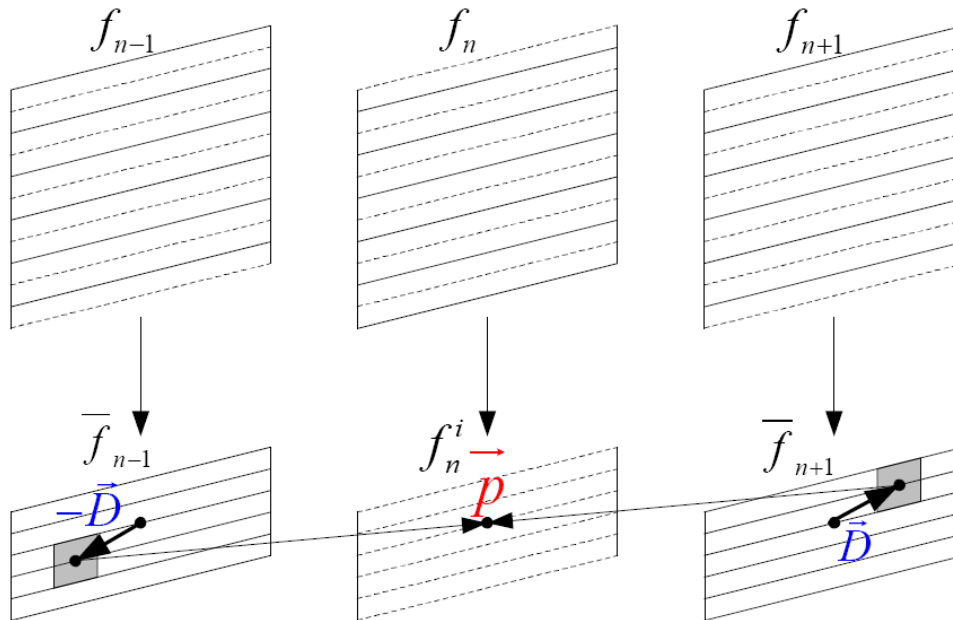
- Введение
- Классификация методов
- Пространственные методы
- Motion Adaptive
- Motion Compensation based
 - Схема
 - MCBob
 - MSU Deinterlacer
- Сравнение
- Текущие результаты и дальнейшие планы

Сложные методы MC-based Deinterlacing



Сложные методы MC-based Deinterlacing

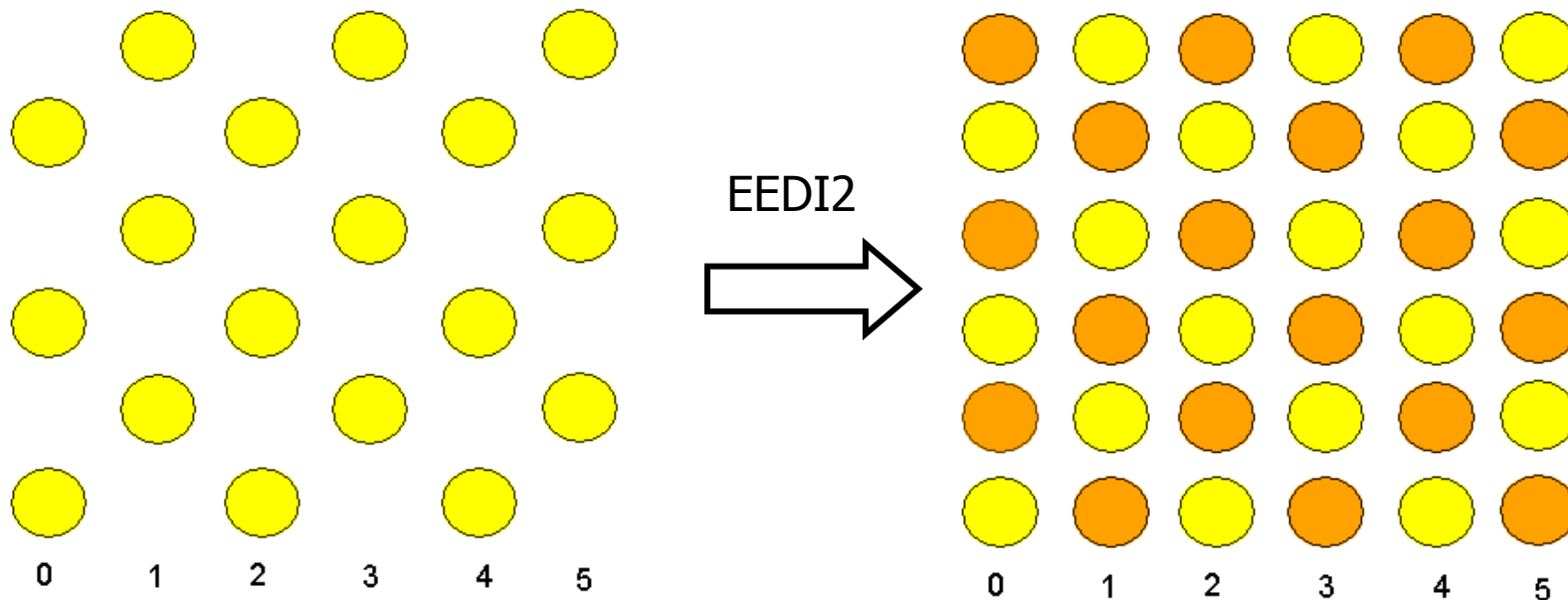
$$f_n^i(\vec{p}) = \frac{1}{2}(\hat{f}_{n-1}(\vec{p} - \vec{D}) + \hat{f}_{n+1}(\vec{p} + \vec{D}))$$



МС методы

МСBob

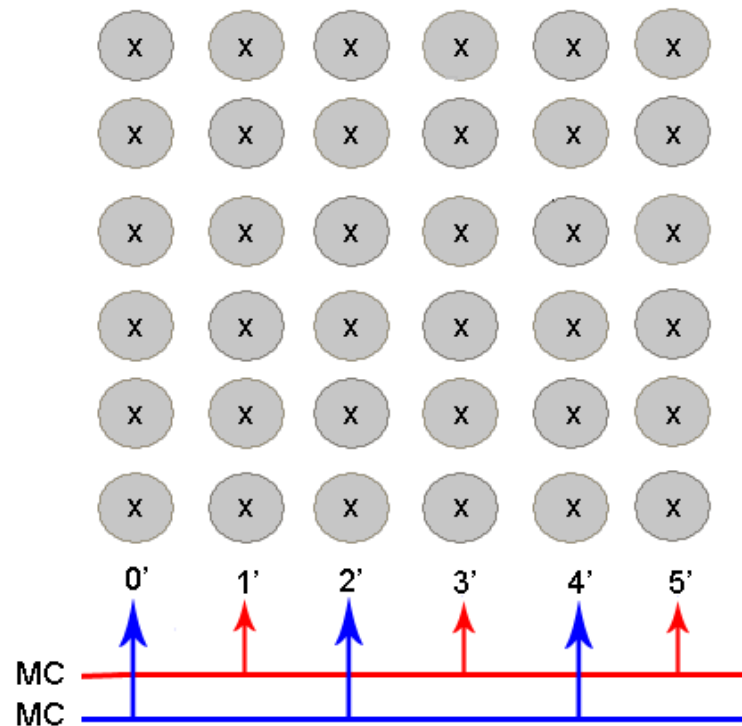
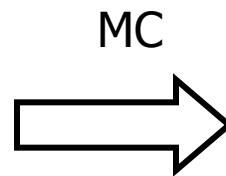
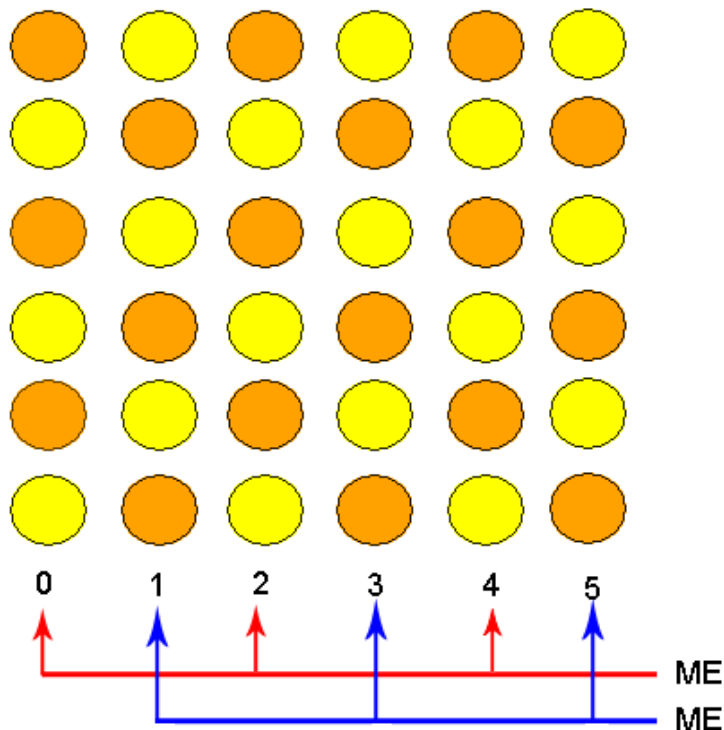
Интерполяция пространственным методом (например, EEDI2 или NNEDI)



МС методы

МСBob

Компенсация движения по полям одинаковой четности в
обоих направлениях

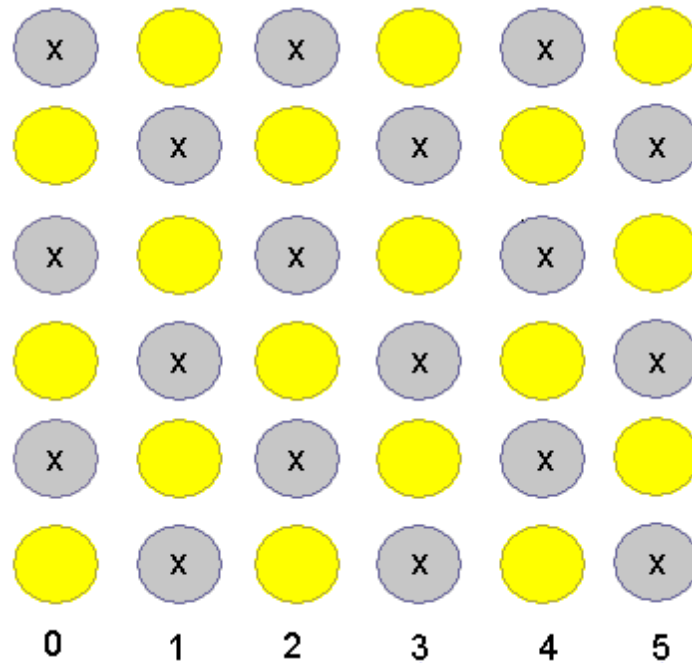


МС методы

МСBob



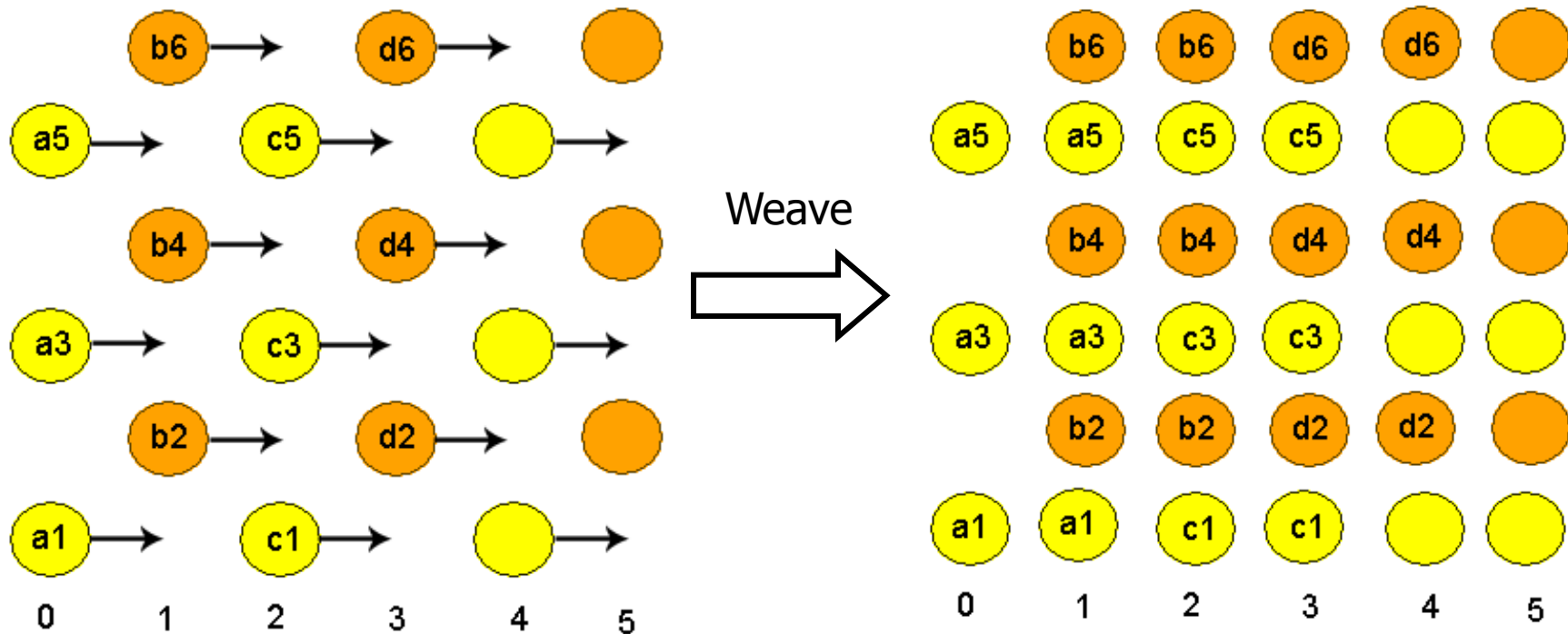
В итоге получаем тс-интерполяцию



МС методы

МСBob

Временная интерполяция (Weave)



МС методы

МСBob



Построение маски ошибок для МС- и пространственной интерполяции - *CorrMask*

CorrMask строится на основе

- Маски ошибки компенсации (МС frame и Spatial-interpolated frame)
- Самого скомпенсированного кадра
- Маски вертикальных границ
- Анализа векторов движения

МС методы

МСBob

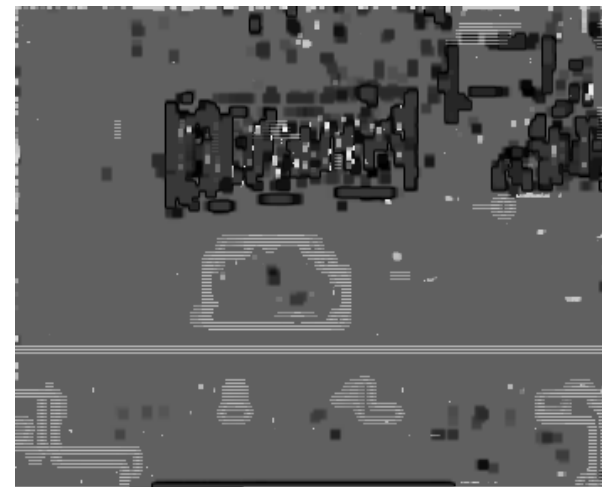
Маска ошибок для МС- и пространственной интерполяции



Ошибка компенсации



Маска вертикальных
границ



CorrMask

МС методы

МСBob



Построение маски весов для временной интерполяции

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

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

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

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

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

$MThr1, MThr2 - const$

МС методы

МСBob

Маска NotStatic весов для временной интерполяции



$Motn_n$



MM_n



$NotStatic_n$

МС методы

МСBob



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

$Temp_n(x, y)$ – пиксель, интерполированный временным способом

$Spat_n(x, y)$ – пиксель, интерполированный пространственным способом

$MC_n(x, y)$ – скомпенсированный пиксель

МС методы

МСBob



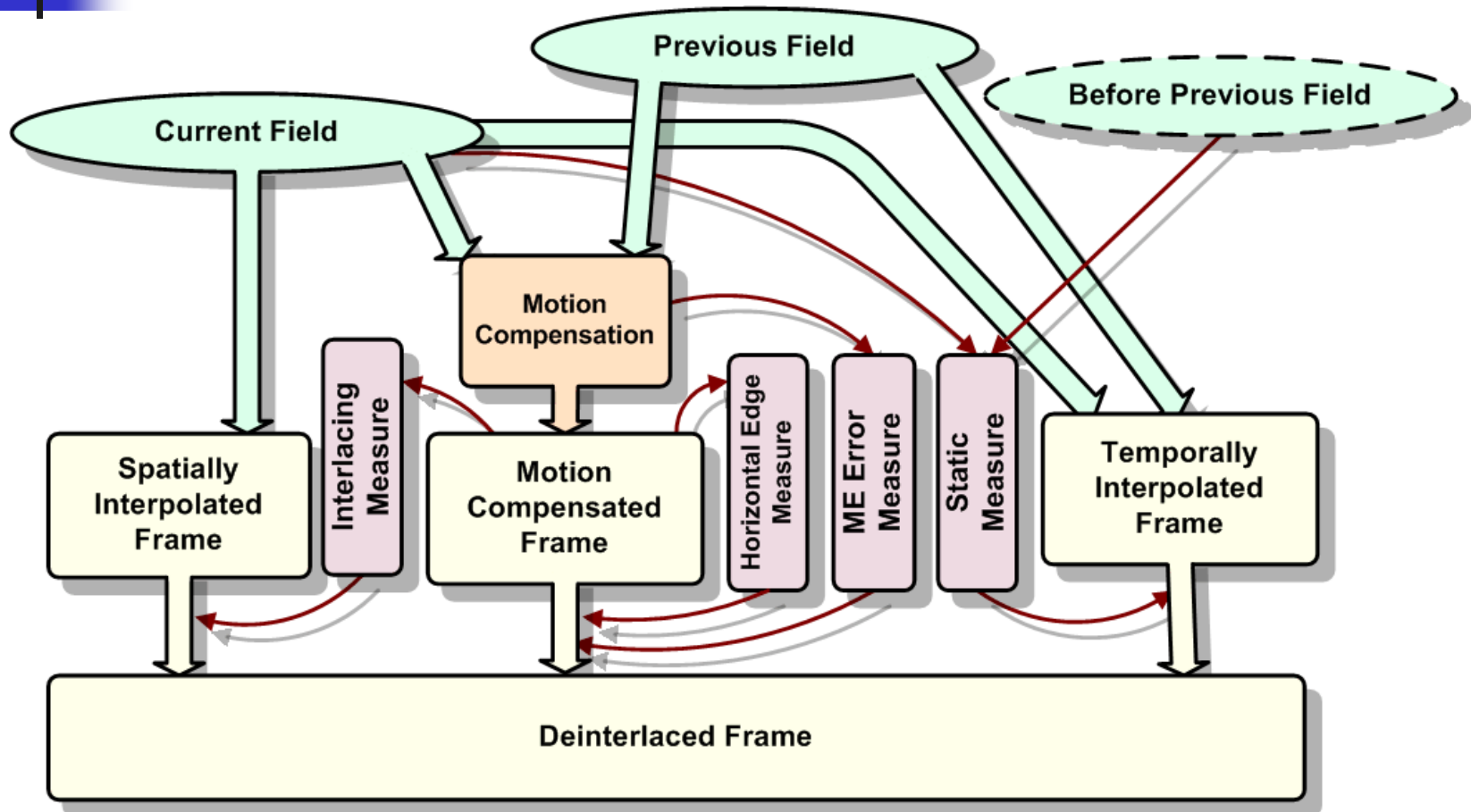
- Выводы

- Хорошая интерполяция (лучший, среди рассмотренных).
- Очень медленный, за счет компенсации и пространственной интерполяции.

Содержание

- Введение
- Классификация методов
- Пространственные методы
- Motion Adaptive
- Motion Compensation based
 - Схема
 - MCBob
 - MSU Deinterlacer
- Сравнение
- Текущие результаты и дальнейшие планы

MC методы MSU Deinterlacer



МС методы

MSU Deinterlacer



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

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

SpatCoef – пространственный весовой коэффициент

TempCoef – временной весовой коэффициент

MCCoef – весовой коэффициент МС

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

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

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

МС методы

MSU Deinterlacer



- Временная интерполяция
 - Используется пиксель предыдущего поля.
 - Весовая функция зависит от дисперсии локальной области пикселя и разности полей n и $n-2$ в локальной области.
- Пространственная интерполяция
 - Используются 4 способа: диагональная интерполяция, 2 вертикальных, и low-edge интерполяция.
 - Весовая функция (interlacing measure) принимает большие значения в тех местах, где присутствует «зубчатость»
- Компенсация движения
 - Ищется соответствующее поле в предыдущем обработанном кадре
 - Весовая функция зависит от дисперсии и ошибки компенсации.

Содержание

- Введение
- Классификация методов
- Пространственные методы
- Motion Adaptive
- Motion Compensation based
- Сравнение
- Текущие результаты и дальнейшие планы

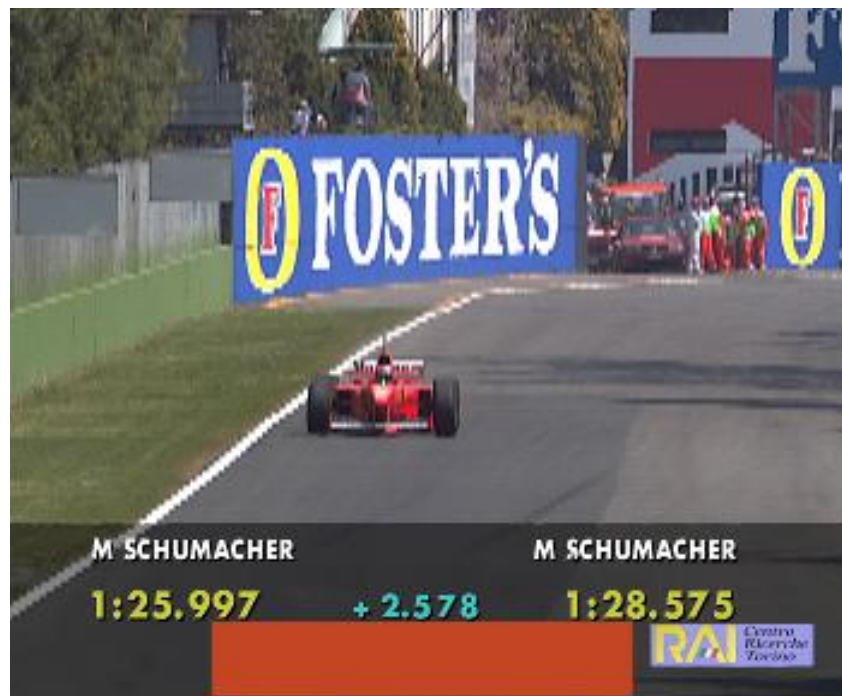
Сравнение

Yadif



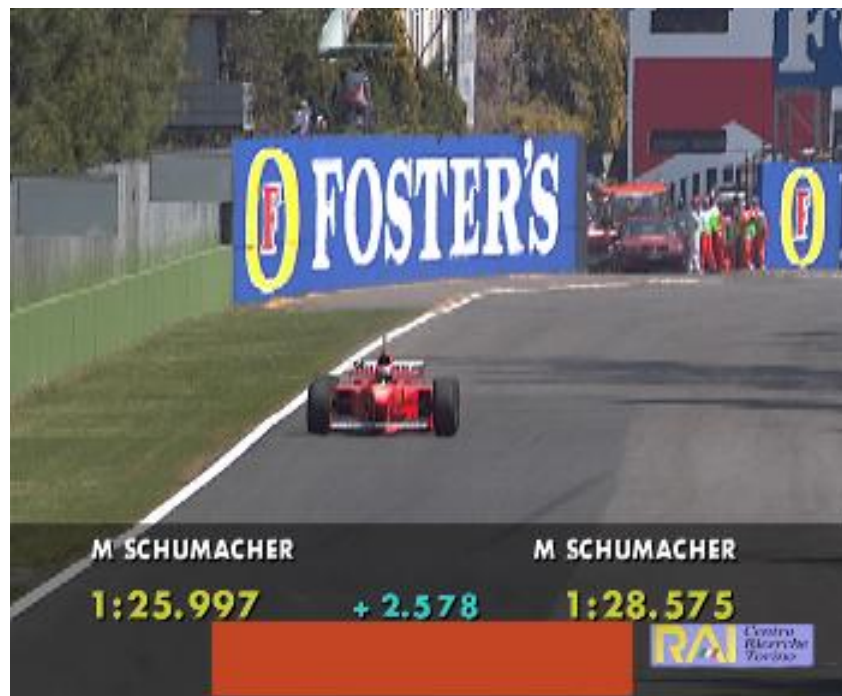
PSNR: 29,84831

Сравнение Smart



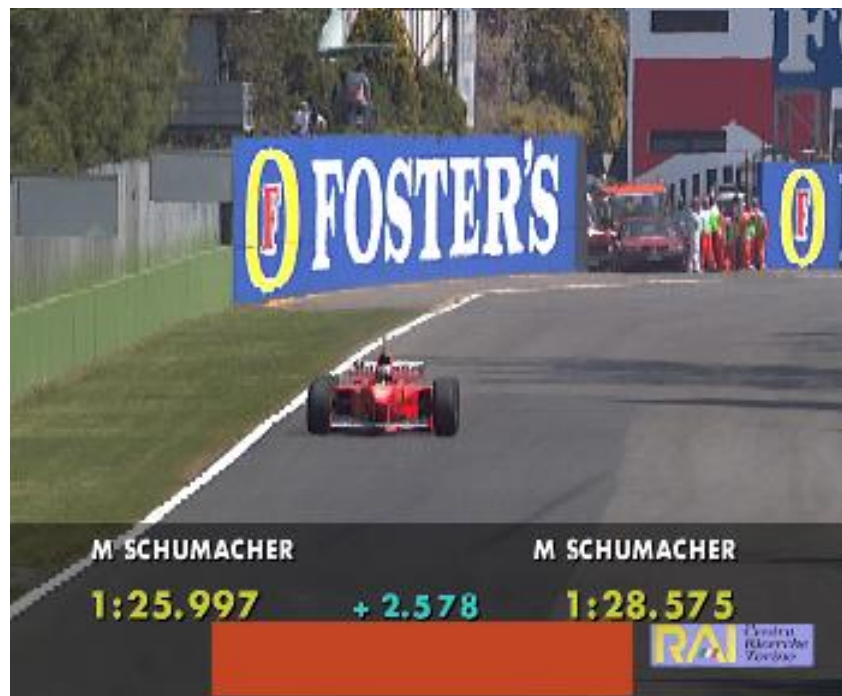
PSNR: 30,10036

Сравнение My Deinterlacer



PSNR: 30,23836

Сравнение Decomb



PSNR: 30,5194

Сравнение Yadifmod



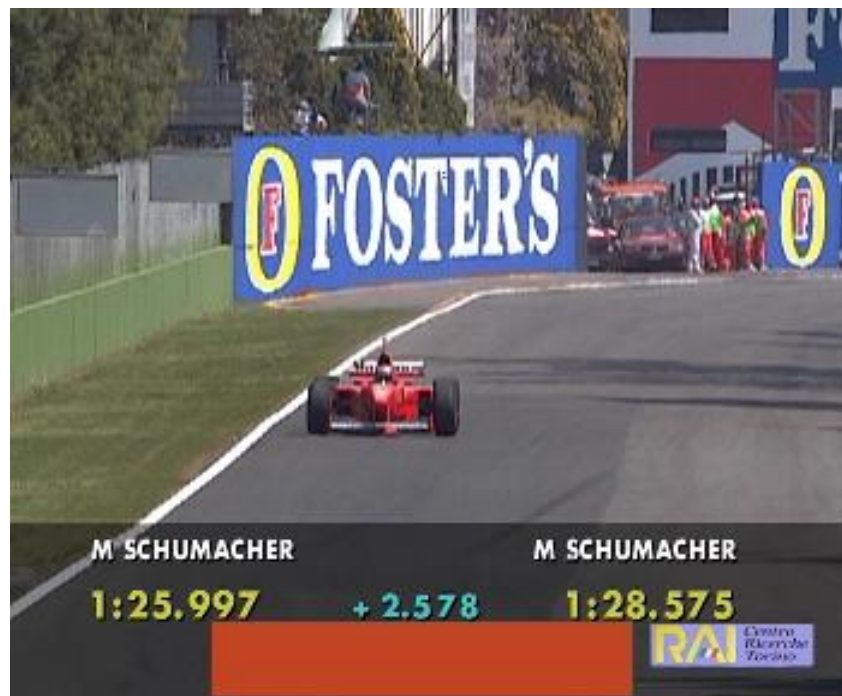
PSNR: 30,96683

Сравнение TDeint+EEDI2



PSNR: 31,08448

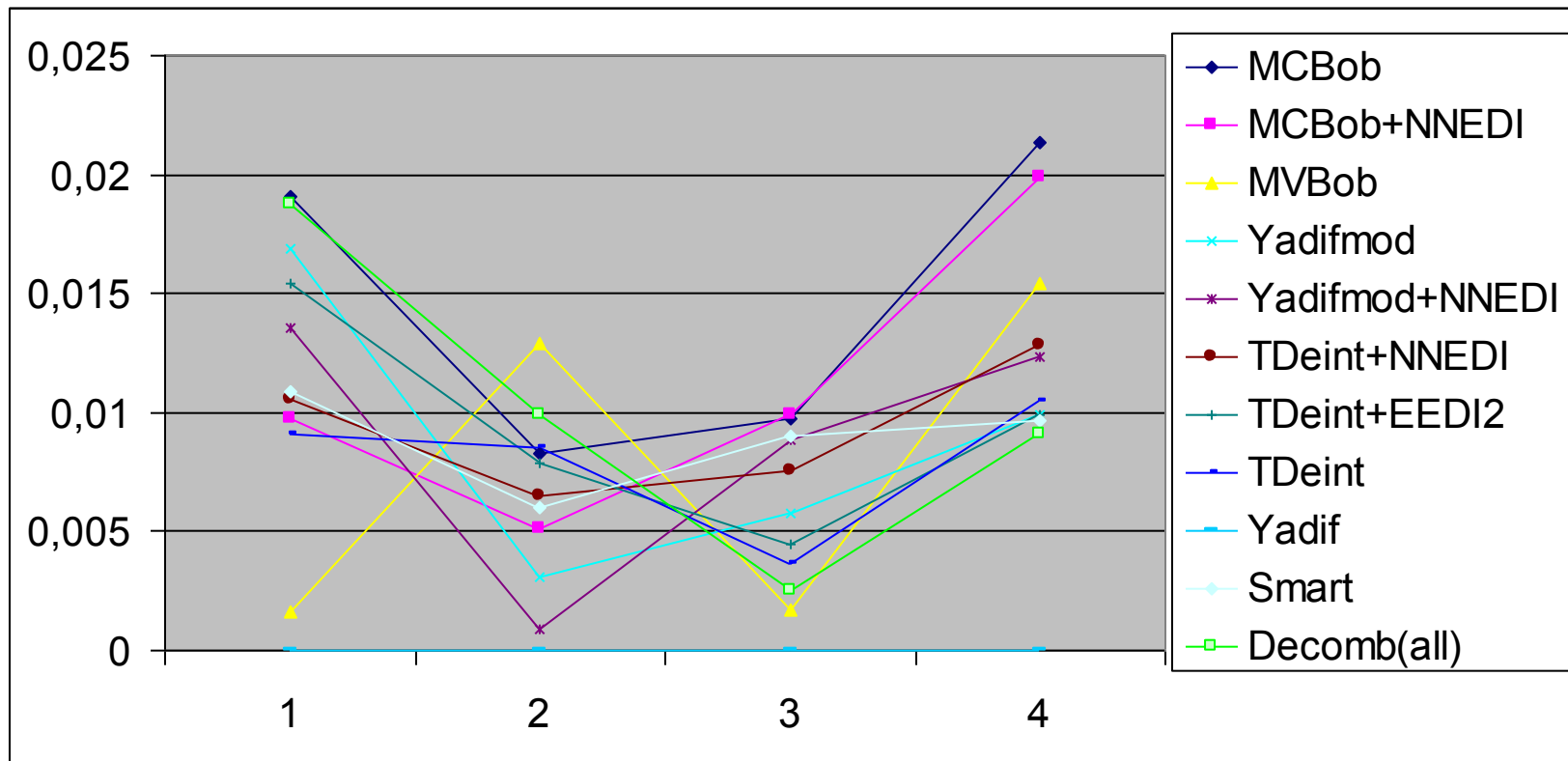
Сравнение MCVob



PSNR: 34,07132

Сравнение

Относительная шкала (PSNR)





Содержание

- Введение
- Классификация методов
- Пространственные методы
- Motion Adaptive
- Motion Compensation based
- Сравнение
- Текущие результаты и дальнейшие планы

Текущие результаты

Сравнение (PSNR)



Filter	Shumacher	Susi	Rancho	Bus
MCBob	26,98677	42,15459	43,77567	33,36838
MCBob+NNEDI	26,4564	41,45398	43,78585	32,75019
Yadifmod	26,82642	40,70768	40,97435	29,86374
Yadifmod+NNEDI	26,76849	40,55473	42,84486	30,55088
TDeint+NNEDI	26,67641	40,96761	42,84166	30,51252
TDeint+EEDI2	26,80859	41,07437	40,89787	29,72796
TDeint	26,3774	40,88614	42,06143	30,49226
TDeint(type=3)	26,59695	40,73497	39,41695	28,80793
MVBob	26,45459	42,40382	41,27823	30,7023
Yadif	26,2369	40,2328	39,38717	27,88672
Smart	26,54378	40,37844	42,28166	29,85481
Decomb(all)	26,89904	41,33203	39,46581	29,68799
My	26,75714 (6)	40,53711 (11)	40,21761 (10)	29,72072 (10)

Текущие результаты

Сравнение (SSIM)



Filter	Shumacher	Susi	Rancho	Bus
MCBob	0,83731	0,97652	0,98869	0,96803
MCBob+NNEDI	0,82791	0,97337	0,98886	0,96365
Yadifmod	0,83507	0,97129	0,9847	0,93362
Yadifmod+NNEDI	0,83177	0,9691	0,98774	0,94102
TDeint+NNEDI	0,82874	0,9747	0,9865	0,9424
TDeint+EEDI2	0,83362	0,97607	0,98338	0,93374
TDeint	0,82008	0,97673	0,9826	0,93526
TDeint(type=3)	0,8273	0,97584	0,97645	0,91405
MVBob	0,81986	0,98112	0,98065	0,95011
Yadif	0,8182	0,96823	0,97893	0,90394
Smart	0,82907	0,97427	0,98792	0,93296
Decomb(all)	0,83694	0,97812	0,98144	0,93133
My	0,83625 (3)	0,97253 (10)	0,98303 (8)	0,9282 (11)

Дальнейшие планы

- Улучшить метрики доверия к кадрам, интерполированным разными способами.
- Тем самым, добиться продвижения в лидеры среди наиболее популярных фильтров.
- Улучшение по скорости, поставив фильтр на GPU.

Список материалов

1. <http://forum.doom9.org/>
2. <http://avisynth.org.ru/>
3. <http://bengal.missouri.edu/~kes25c/> - tritical's web page
4. http://bengal.missouri.edu/~kes25c/deinterlace_comparison.txt
5. http://avisynth.org/mediawiki/External_filters#Deinterlacing



Вопросы

