PDE в обработке видео
Оптический поток

PDE = уравнения в частных производных

$$\Psi'(I_z^2 + \gamma(I_{xz}^2 + I_{yz}^2)) \cdot (I_x I_z + \gamma(I_{xx} I_{xz} + I_{xy} I_{yz})) - \alpha \cdot \mathrm{div}(\Psi'(|\nabla_3 u|^2 + |\nabla_3 v|^2)\nabla_3 u) = 0,$$

$$\Psi'(I_z^2 + \gamma(I_{xz}^2 + I_{yz}^2)) \cdot (I_y I_z + \gamma(I_{yy} I_{yz} + I_{xy} I_{xz})) - \alpha \cdot \mathrm{div}(\Psi'(|\nabla_3 u|^2 + |\nabla_3 v|^2)\nabla_3 v) = 0$$

Видео = ...

Компьютерное зрение

Одна из основных задач:
Получить информацию о движении в кадре

Компьютерное зрение

Одна из основных задач:
Получить информацию о движении в кадре

- Идентифицировать движение
- Определить его направление
- Определить скорость

Компьютерное зрение

Одна из основных задач:
Получить информацию о движении в кадре

- Идентифицировать движение
- Определить его направление
- Определить скорость

Положение и перемещения камеры обычно неизвестны

Компьютерное зрение

Одна из основных задач:
Получить информацию о движении в кадре

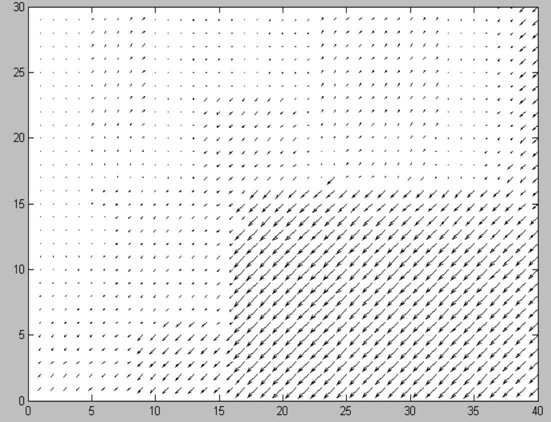
- Идентифицировать движение
- Определить его направление
- Определить скорость

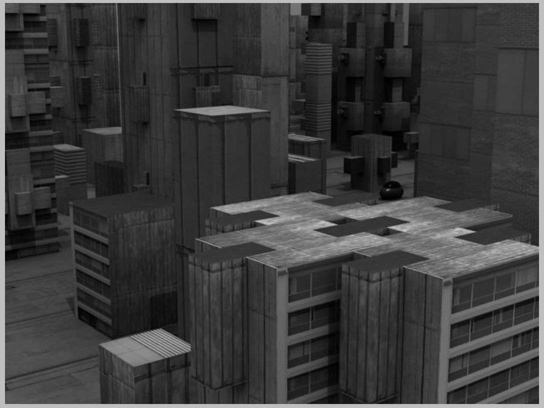
Положение и перемещения камеры обычно неизвестны

Ограничиваем задачу поиском относительного движения объектов в системе отсчета, связанной с камерой

Компьютерное зрение

Относительное движение между двумя последовательными кадрами можно представить как векторное поле – **оптический поток**

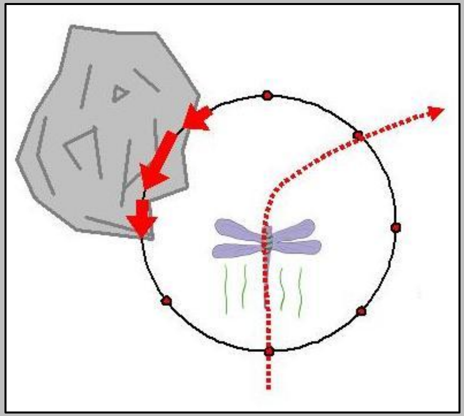




Оптический поток

Применение

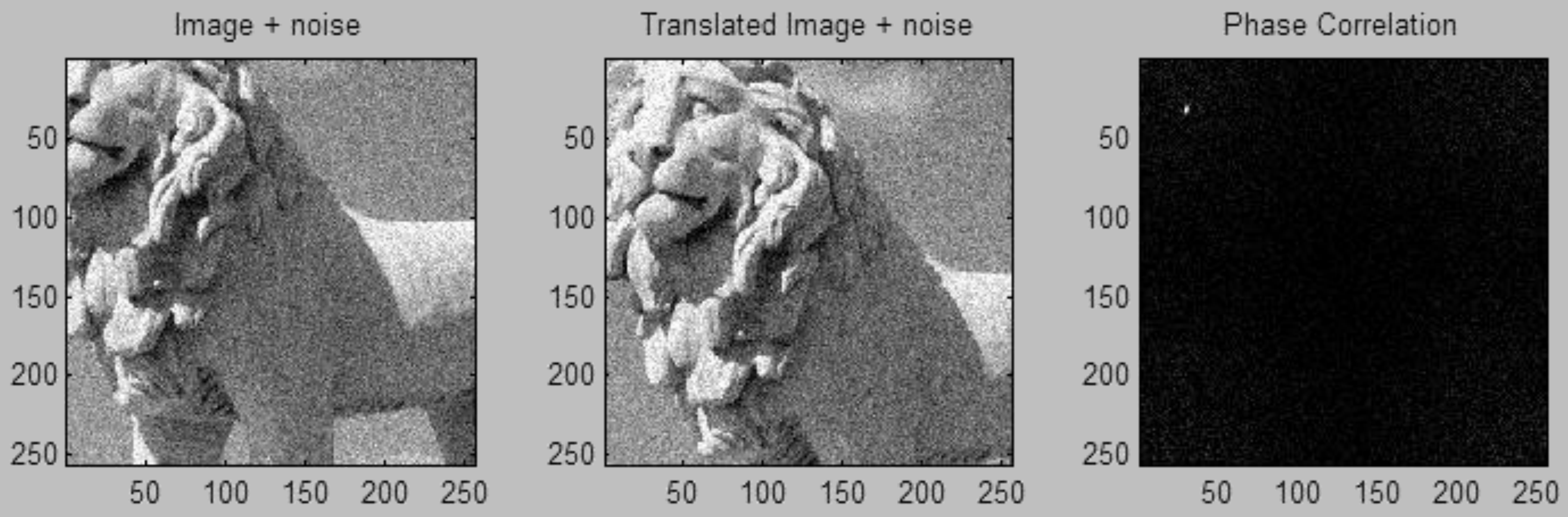
- Визуальные эффекты
- Отслеживание объектов
- Автономная навигация роботов
- Системы видеонаблюдения



Оптический поток

Методы нахождения

- **Фазовая корреляция**
    - преобразование Фурье
    - только прямолинейное движение
    - все точки перемещаются одинаково



Взаимная корреляция сигналов

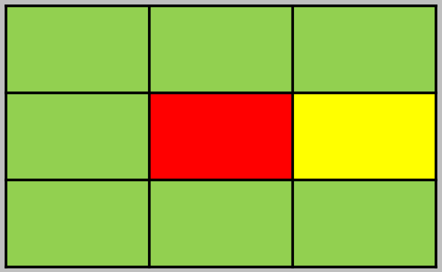
$$(f \bullet g)(t) = \int_{-\infty}^{+\infty} f^*(y) g(y+t) dy$$

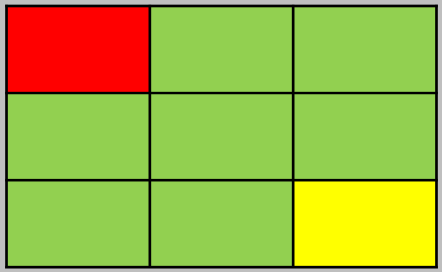
$$F\{f \bullet g\} = (F\{f\})^* \cdot F\{g\}$$

Оптический поток

Методы нахождения

- Фазовая корреляция
    - преобразование Фурье
    - только прямолинейное движение
    - все точки перемещаются одинаково
- **Блочные методы**
    - поиск похожих блоков
    - все точки блока перемещаются одинаково





Оптический поток

Методы нахождения

- Фазовая корреляция
  - преобразование Фурье
  - только прямолинейное движение
  - все точки перемещаются одинаково
- Блочные методы
  - поиск похожих блоков
  - все точки блока перемещаются одинаково
- **Вариационные методы**
  - минимизация некоторого функционала

$$E(u,v) \to \inf$$

$$x - a = b \Leftrightarrow |x - a - b| = 0$$

$$u - \text{решение} \Leftrightarrow |u - a - b| = \inf_x |x - a - b| = 0$$

$$\Rightarrow E(u) = |u - a - b| \to \inf$$

Оптический поток

Вариационные методы

- формулировка в виде задачи оптимизации
- плотное поле скоростей
  - для каждого пикселя
- высокая точность
  - смещение не обязательно кратно размеру пикселя

Оптический поток

Построение модели

•**Дано**:

•Непрерывная последовательность изображений

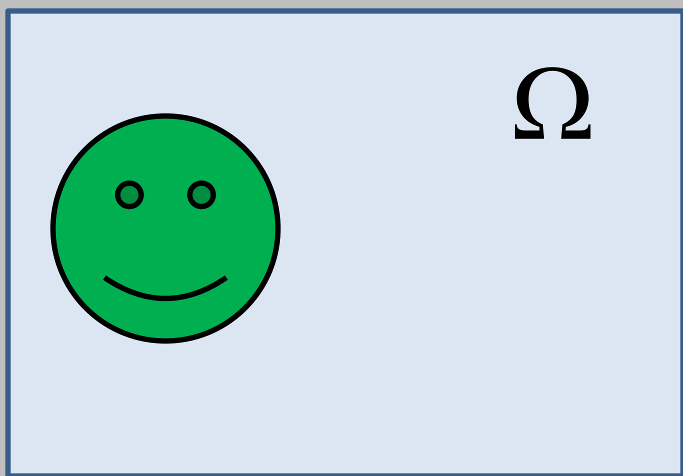
$$\underbrace{I(x,y,t)}_{\text{яркость}},\ \underbrace{(x,y)}_{\text{в точке}}\in\Omega,\ \underbrace{\mathrm{t}}_{\text{в момент времени}}\in[0,T]$$

•**Надо найти**:

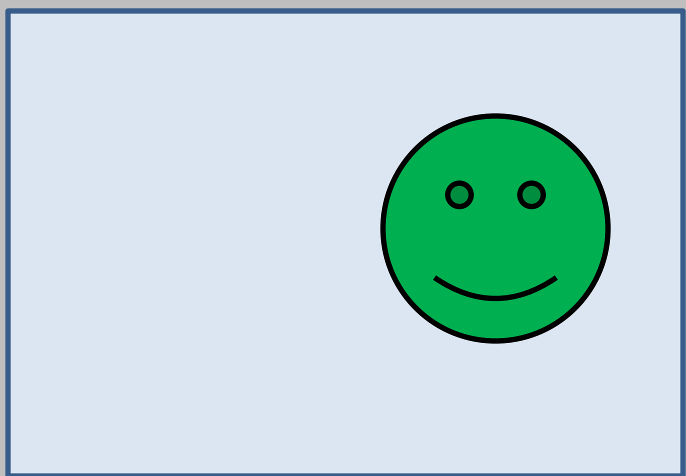
•Поле смещений

$$\mathbf{w}(x,y,t)=\begin{pmatrix} u(x,y,t) \\ v(x,y,t) \\ 1 \end{pmatrix}$$

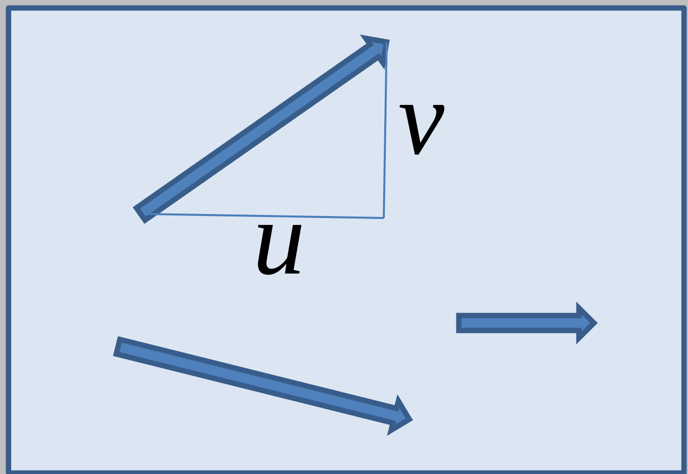
Оптический поток



$I(\cdot,\cdot,t)$



$I(\cdot,\cdot,t+1)$



$\mathbf{w}(\cdot,\cdot,t)$

Построение модели

Предположение 1
Постоянство яркости пикселей

$$I(x+u, y+v, t+1) - I(x, y, t) = 0$$

Предположение 2
$u,\ v$ – малы, I – гладкая функция
В этом случае предыдущее равенство можно линеаризовать в точке (x, y, t)

$$I(x+u, y+v, t+1) - I(x, y, t) \approx I_x(x, y, t)u + I_y(x, y, t)v + I_t(x, y, t) \cdot 1$$

$$\Rightarrow I_x u + I_y v + I_t = 0$$

Построение модели

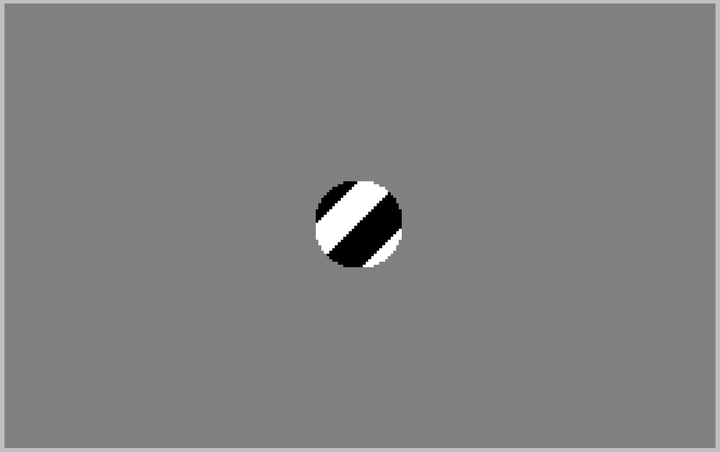
$$I_x u + I_y v + I_t = 0$$

- Одно уравнение для двух неизвестных
- Некорректная задача с бесконечным числом решений
- Известно как **проблема апертуры**(aperture problem)
  - наблюдая лишь за небольшой частью кадра невозможно однозначно определить направление движения

Построение модели

$$I_x u + I_y v + I_t = 0$$

- Одно уравнение для двух неизвестных
- Некорректная задача с бесконечным числом решений
- Известно как **проблема апертуры**(aperture problem)
  - наблюдая лишь за небольшой частью кадра невозможно однозначно определить направление движения

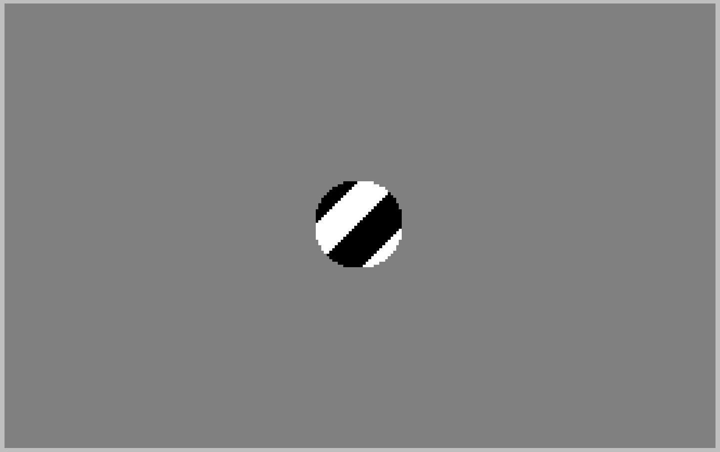


Как движется шаблон?

Построение модели

$$I_x u + I_y v + I_t = 0$$

- Одно уравнение для двух неизвестных
- Некорректная задача с бесконечным числом решений
- Известно как **проблема апертуры**(aperture problem)
  - наблюдая лишь за небольшой частью кадра невозможно однозначно определить направление движения

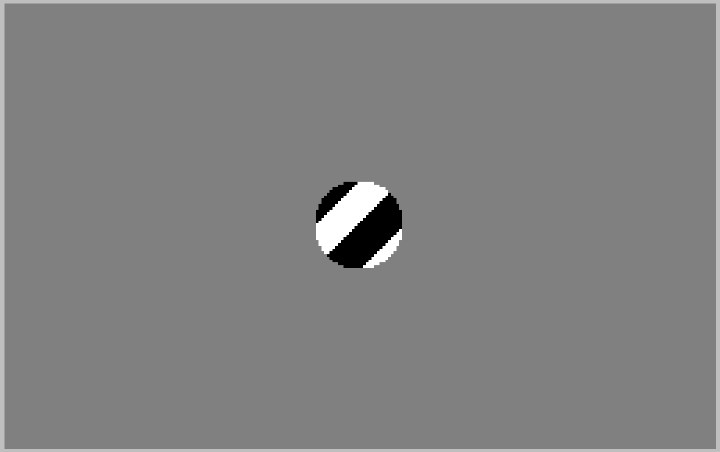


Как движется шаблон?
По диагонали вправо?

Построение модели

$$I_x u + I_y v + I_t = 0$$

- Одно уравнение для двух неизвестных
- Некорректная задача с бесконечным числом решений
- Известно как **проблема апертуры**(aperture problem)
  - наблюдая лишь за небольшой частью кадра невозможно однозначно определить направление движения

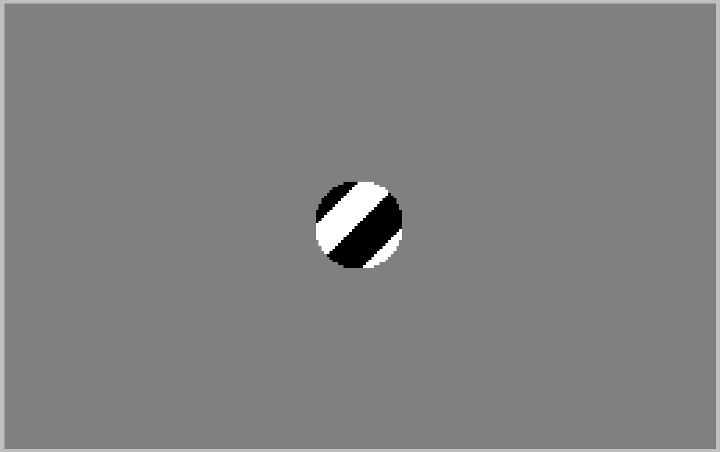


Как движется шаблон?
По диагонали вправо?
Только вправо?

Построение модели

$$I_x u + I_y v + I_t = 0$$

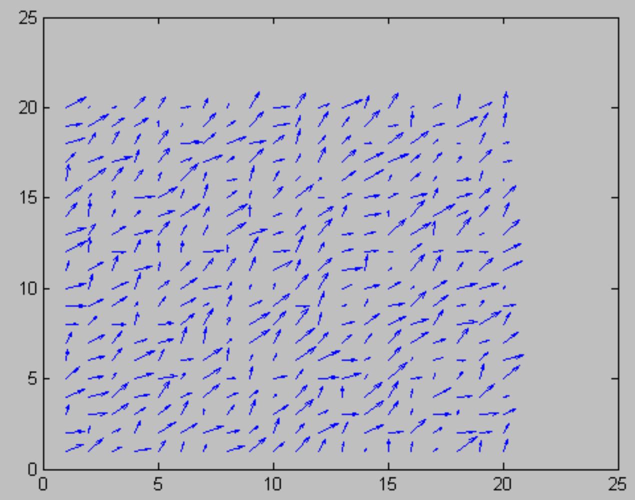
- Одно уравнение для двух неизвестных
- Некорректная задача с бесконечным числом решений
- Известно как **проблема апертуры**(aperture problem)
  - наблюдая лишь за небольшой частью кадра невозможно однозначно определить направление движения



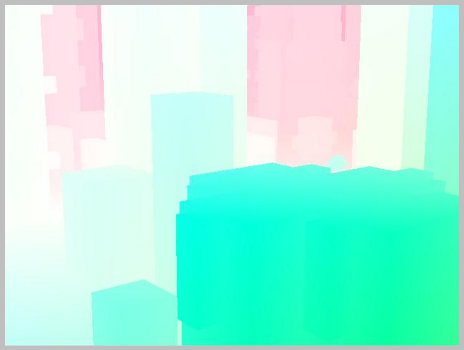
Как движется шаблон?
По диагонали вправо?
Только вправо?
Или только вниз?

Как визуализировать векторное поле

При помощи стрелок, предварительно понизив разрешение



При помощи цвета:
•Направление – цвет
•Абсолютное значение – яркость



Как оценить качество рассчитанного потока

Настоящий поток (ground truth) w^t

Вычисленный поток (estimated) w^e

Размер изображения $M \times N$

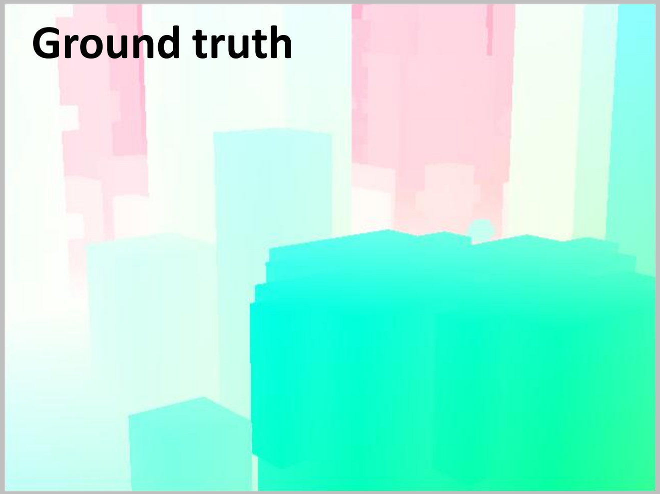
Средняя угловая ошибка (AAE – Average Angular Error)

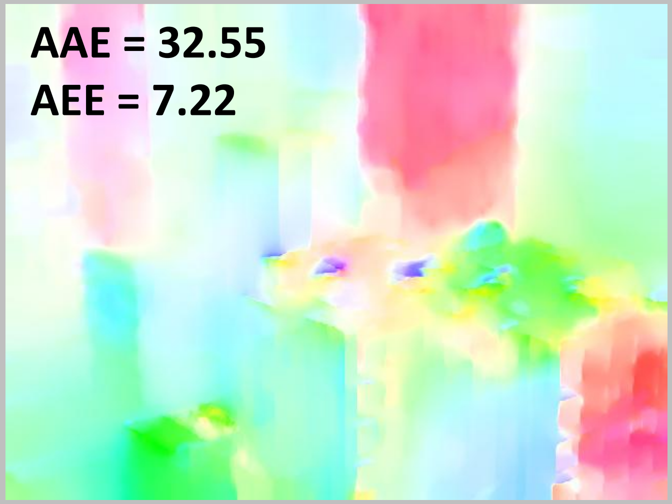
$$AAE = \frac{1}{NM} \sum_{i=1}^{M} \sum_{j=1}^{N} \arccos\left(\frac{w_{i,j}^{t}}{\left| w_{i,j}^{t} \right|}^{T} \frac{w_{i,j}^{e}}{\left| w_{i,j}^{e} \right|} \right)$$

Средняя ошибка по конечной точке (AEE – Average Endpoint Error)

$$AEE = \frac{1}{NM} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| w_{i,j}^{t} - w_{i,j}^{e} \right|$$

Как оценить качество рассчитанного потока





МЕТОД LUCAS-KANADE

Lucas B. D. and Kanade T. 1981, An iterative image registration technique with an application to stereo vision.

Proceedings of Imaging understanding workshop, pp 121--130

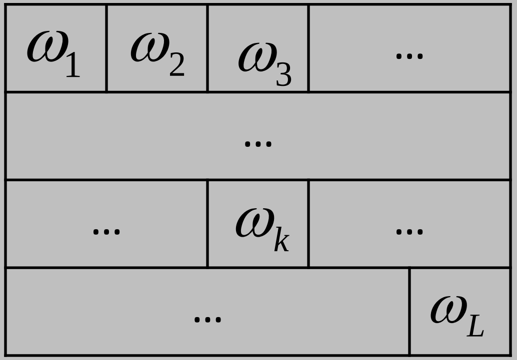
Метод Lucas-Kanade

Предположение

•Поток кусочно-постоянный

Разобьем изображение на небольшие части (например, блоки 8x8), в которых все точки двигаются одинаково

$$\Omega = \coprod_k \omega_k$$



Для каждой из частей запишем линеаризованное уравнение постоянства яркости

$$I_x(i, j)u(\omega_k) + I_y(i, j)v(\omega_k) = -I_t(i, j),\ (i, j) \in \omega_k$$

Метод Lucas-Kanade

Полученную систему уравнений можно записать в матричном виде

$$A \cdot \begin{pmatrix} u \\ v \end{pmatrix} = -b$$

$$A = \begin{pmatrix} I_x(i_1, j_1) & I_y(i_1, j_1) \\ \vdots & \vdots \\ I_x(i_L, j_L) & I_y(i_L, j_L) \end{pmatrix}, \quad b = \begin{pmatrix} I_t(i_1, j_1) \\ \vdots \\ I_t(i_L, j_L) \end{pmatrix}$$

Метод Lucas-Kanade

Полученную систему уравнений можно записать в матричном виде

$$A \cdot \begin{pmatrix} u \\ v \end{pmatrix} = -b$$

$$A = \begin{pmatrix} I_x(i_1, j_1) & I_y(i_1, j_1) \\ \vdots & \vdots \\ I_x(i_L, j_L) & I_y(i_L, j_L) \end{pmatrix}, \quad b = \begin{pmatrix} I_t(i_1, j_1) \\ \vdots \\ I_t(i_L, j_L) \end{pmatrix}$$

Уравнений больше чем неизвестных
Ищем **псевдорешение** методом наименьших квадратов

Метод Lucas-Kanade

Псевдорешение

$$A^T A \begin{pmatrix} u \\ v \end{pmatrix} = -A^T b$$

$$A^T A = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix}, \quad A^T b = \begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}$$

Для каждой части изображения необходимо
•вычислить квадратную матрицу 2-го порядка
•правую часть – вектор размерности 2
•**решить СЛАУ** 2-го порядка

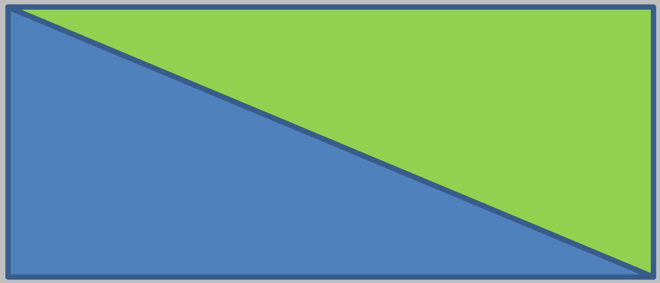
Метод Lucas-Kanade

Недостаток метода:
Матрица может оказаться вырожденной
•В однородных областях (без текстуры)

$$\left.\begin{aligned} I_x &\approx 0 \\ I_y &\approx 0 \end{aligned}\right\} \Rightarrow A^T A \approx 0$$

•На гранях

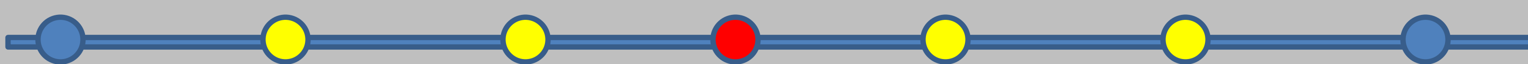
$$I_x = c \cdot I_y \Rightarrow A^T A = \begin{pmatrix} c^2 I_y^2 & c I_y I_y \\ c I_y I_y & I_y^2 \end{pmatrix}$$



Метод Lucas-Kanade

Пространственные производные удобно вычислить заранее
•свертка столбцов(строк) с ядром 5x1 (1x5)

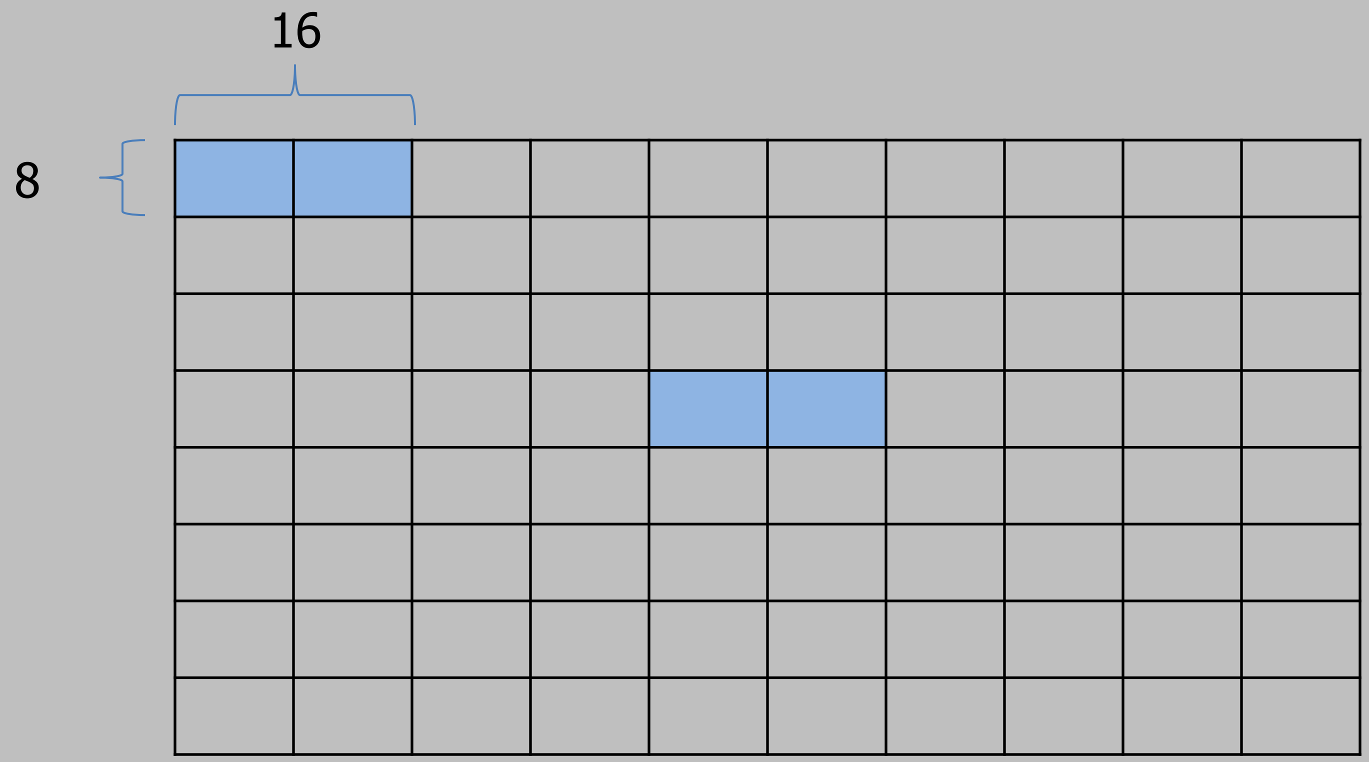
$$\frac{\partial f}{\partial x}(x_i, y_j) \approx \frac{f(x_{i-2}, y_j) - 8f(x_{i-1}, y_j) + 8f(x_{i+1}, y_j) - f(x_{x+2}, y_j)}{12h}$$



	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
Ядро	$\frac{1}{12}$	$-\frac{8}{12}$	0	$\frac{8}{12}$	$-\frac{1}{12}$

Метод Lucas-Kanade

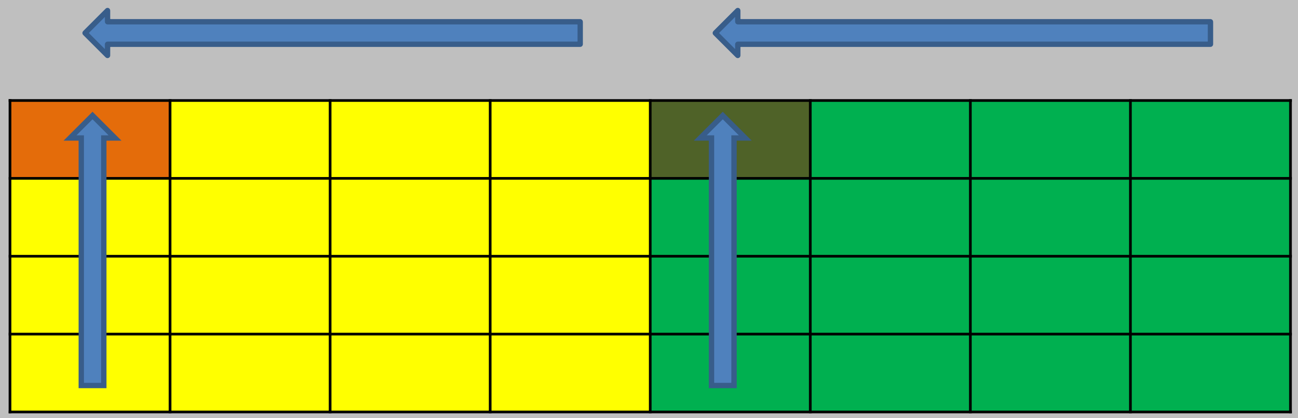
Решение систем
Распределение работы между потоками



Метод Lucas-Kanade

Каждый блок потоков обрабатывает две части изображения

Вычисление матрицы и правой части
•использовать **reduce**



Решение системы уравнений 2x2
•выписывается явно
•удобно вычислять потоками (0,0) и (8,0)

Вариационные методы

Основная идея
поле смещений минимизирует некоторый энергетический функционал

$$E(u,v)=\int_{\Omega} D(u,v)+\alpha\cdot S(u,v)dxdy$$

$D(u,v)$ (**data term**) – штраф за отклонения от предположений о постоянстве какой-либо величины (например, яркости)

$S(u,v)$ (**smooth term**) – штраф за отклонения от предположений гладкости векторного поля

α параметр регуляризации – определяет гладкость получаемого векторного поля

МЕТОД HORN-SCHUNK

B.K.P. Horn and B.G. Schunck, "Determining optical flow." *Artificial Intelligence*, vol 17, pp 185-203, 1981

Метод Horn-Schunck

Предположение
Поле смещений – гладкая функция в области Ω

Поле смещений минимизирует функционал

$$E(u,v) = \int_{\Omega} \underbrace{\left(I_x u + I_y v + I_t\right)^2}_{\text{data}} + \alpha \underbrace{\left(\left|\nabla u\right|^2 + \left|\nabla v\right|^2\right)}_{\text{smooth}} dxdy$$

data term – штраф за отклонения от предположения постоянства яркости
smooth term – штраф за отклонение от предположения гладкости поля

Минимизация функционала

$$E(u,v)=\int_{\Omega}F(x,y,u,v,u_x,u_y,v_x,v_y)dxdy$$

Уравнения Эйлера-Лагранжа

$$0=F_u-\frac{\partial}{\partial x}F_{u_x}-\frac{\partial}{\partial y}F_{u_y}$$

$$0=F_v-\frac{\partial}{\partial x}F_{v_x}-\frac{\partial}{\partial y}F_{v_y}$$

С граничными условиями

$$\mathbf{n}^T\begin{pmatrix}F_{u_x}\\F_{u_y}\end{pmatrix}=0,\qquad \mathbf{n}^T\begin{pmatrix}F_{v_x}\\F_{v_y}\end{pmatrix}=0$$

Метод Horn-Schunck

Уравнения Эйлера-Лагранжа

$$\begin{cases} I_x(I_x u + I_y v + I_t) - \alpha \cdot \Delta u = 0 \\ I_y(I_x u + I_y v + I_t) - \alpha \cdot \Delta v = 0 \end{cases}$$

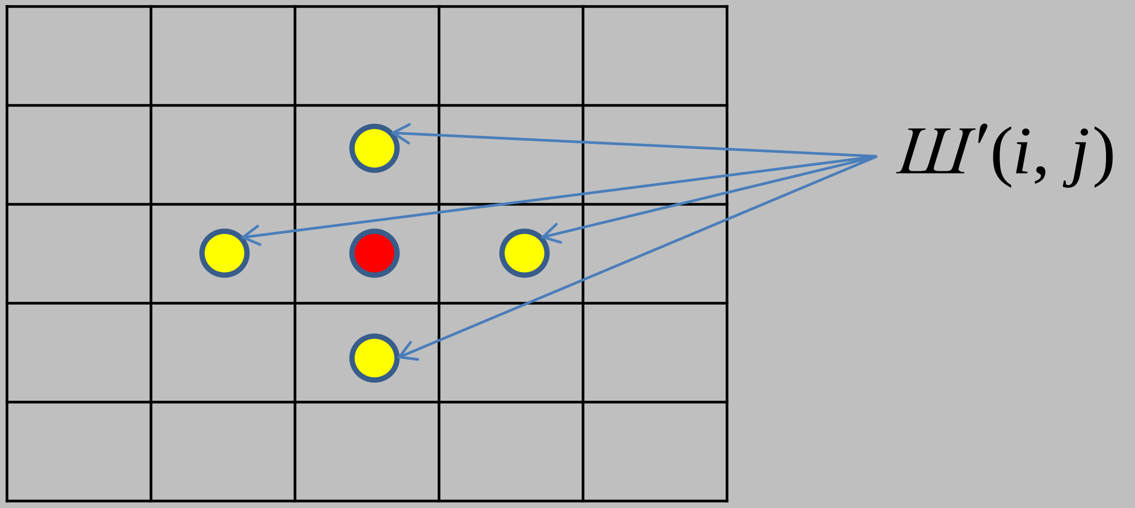
$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Граничные условия

$$\mathbf{n}^T \nabla u = 0, \quad \mathbf{n}^T \nabla v = 0$$

Метод Horn-Schunck

Аппроксимация оператора Лапласа



$$\Delta u_{i,j} \approx \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2}$$

Метод Horn-Schunck

Дискретизация уравнений

$$\begin{cases} 0 = I_x^2 u_{i,j} + I_x I_y v_{i,j} + I_x I_t - \alpha \cdot \sum\limits_{(\bar{i},\bar{j}) \in Ш'(i,j)} \dfrac{u_{\bar{i},\bar{j}} - u_{i,j}}{h^2} \\ 0 = I_x I_y u_{i,j} + I_y^2 v_{i,j} + I_y I_t - \alpha \cdot \sum\limits_{(\bar{i},\bar{j}) \in Ш'(i,j)} \dfrac{v_{\bar{i},\bar{j}} - v_{i,j}}{h^2} \end{cases}$$

Граничные условия уже учтены в дискретизованном уравнении
•Шаг сетки *h* обычно принимают равным 1

•**Система с *2xNxM* неизвестными** – метод Гаусса неприменим из-за высокой сложности и плохой устойчивости

Метод Якоби для решения СЛАУ

Рассмотрим СЛАУ вида $Ax = b$

$$A = L + D + U$$

Итерационный метод

$$x^0 = 0$$

$$x^{k+1} = D^{-1}(b - (L+U)x^k) \quad \Leftrightarrow \quad x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j \neq i} a_{ij} x_j^k\right)$$

Метод Гаусса-Зейделя для решения СЛАУ

Рассмотрим СЛАУ вида $Ax = b$

$$A = L + D + U$$

Итерационный метод

$$x^0 = 0$$

$$x^{k+1} = (D+L)^{-1}(b - Ux^k) \Leftrightarrow x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j>i} a_{ij} x_i^k - \sum_{j<i} a_{ij} x_i^{k+1}\right)$$

Метод релаксации (SOR) для решения СЛАУ

Рассмотрим СЛАУ вида $Ax = b$

$$A = L + D + U$$

Итерационный метод

$$x^0 = 0$$

$$x^{k+1} = (D + \omega L)^{-1}(\omega b - (\omega U + (\omega - 1)D)x^k) \Leftrightarrow$$

$$x_i^{k+1} = (1-\omega)x_i^k + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j>i} a_{ij} x_j^k - \sum_{j<i} a_{ij} x_j^{k+1}\right)$$

Метод Horn-Schunck

Метод Якоби

$$u_{i,j}^{k+1} = \frac{\left(-I_x I_t - \left(I_x I_y v_{i,j}^k - \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'(i,j)} \frac{1}{h^2} u_{\bar{i},\bar{j}}^k\right)\right)}{I_x^2 + \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'(i,j)} \frac{1}{h^2}}$$

$$v_{i,j}^{k+1} = \frac{\left(-I_y I_t - \left(I_x I_y u_{i,j}^k - \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'(i,j)} \frac{1}{h^2} v_{\bar{i},\bar{j}}^k\right)\right)}{I_y^2 + \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'(i,j)} \frac{1}{h^2}}$$

Просто распараллеливается

Приближение на следующей итерации полностью определяется по приближению на текущей итерации – нет зависимости между разными пикселями

Метод Horn-Schunck

Метод Гаусса-Зейделя

$$u_{i,j}^{k+1} = \frac{\left(-I_x I_t - \left(I_x I_y v_{i,j}^{k} - \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'_-(i,j)} \frac{1}{h^2} u_{\bar{i},\bar{j}}^{\boxed{k+1}} - \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'_+(i,j)} \frac{1}{h^2} u_{\bar{i},\bar{j}}^{k}\right)\right)}{I_x^2 + \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'(i,j)} \frac{1}{h^2}}$$

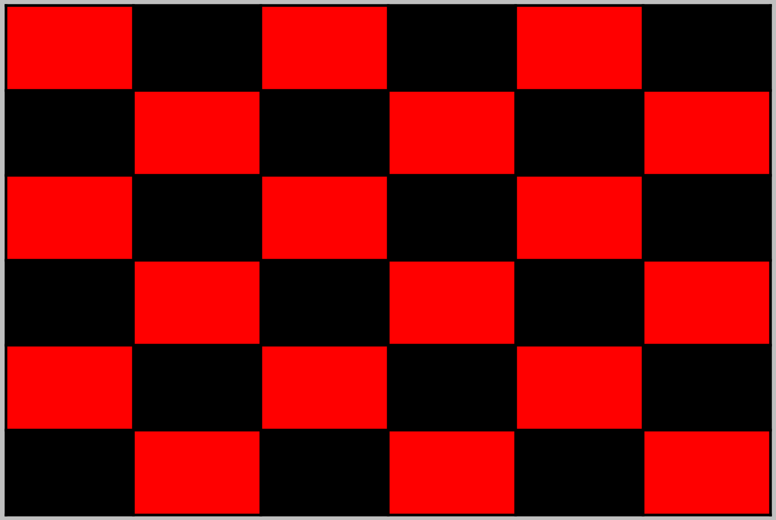
$$v_{i,j}^{k+1} = \frac{\left(-I_y I_t - \left(I_x I_y u_{i,j}^{\boxed{k+1}} - \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'_-(i,j)} \frac{1}{h^2} v_{\bar{i},\bar{j}}^{\boxed{k+1}} - \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'_+(i,j)} \frac{1}{h^2} v_{\bar{i},\bar{j}}^{k}\right)\right)}{I_y^2 + \alpha \sum\limits_{(\bar{i},\bar{j}) \in Ш'(i,j)} \frac{1}{h^2}}$$

	i,j+1	
i-1,j	i,j	i+1,j
	i,j-1	

	$Ш'_+(i,j)$
	$Ш'_-(i,j)$

Нельзя напрямую использовать для параллельных вычислений

Красно-черная схема Гаусса-Зейделя



- Новое значение в красном узле зависит только от текущих значений в самом узле и в его «черных соседях»
- Новое значение в черном узле зависит только от текущих значений в самом узле и в его «красных соседях»

Новые значения в узлах одного цвета можно вычислять параллельно

Схема одной итерации

- Обновить значения в красных узлах
- Обновить значения в черных узлах

Аналогично можно сделать параллельную версию SOR

Метод Horn-Schunck

- **Ядро выполняет одну итерацию итерационного метода**

Схема вычислений

- Вычислить производные изображения (пространственные и временные)
- Установить начальное приближение - нулевой поток
- Выполнить некоторое количество итераций одного из итерационных методов

Новые значения сохраняются в массив, отличный от массива со старыми значениями. В противном случае возникает конфликт чтения-записи.

$$\boxed{\max_i \left| x_i^{n+1} - x_i^n \right| \le \varepsilon}$$ критерий останова

$$\max_i \left| \left(A x^{n+1} \right)_i - b_i \right| \le \varepsilon$$

МЕТОД BROX ET AL

T. Brox, A. Bruhn, N. Papenberg, J. Weickert
High accuracy optical flow estimation based on a theory for warping, T. Pajdla and J. Matas (Eds.), *European Conference on Computer Vision (ECCV)* Prague, Czech Republic, Springer, LNCS, Vol. 3024, 25-36, May 2004

Метод Brox et al

Рассмотренные методы чувствительны к изменению освещения

Метод Brox et al

Рассмотренные методы чувствительны к изменению освещения

Идея

- **Рассмотреть другой data term**

Предположение

Сохраняется градиент изображения

$$\nabla I(x+u, y+v, t+1) - \nabla I(x, y, t) = 0$$

Плюс:

- градиент не чувствителен к аддитивному изменению яркости

Минус:

- чувствительность к шуму

Метод Brox et al

Рассмотренные методы чувствительны к изменению освещения

Идея

•Рассмотреть другой data term

Предположение

Сохраняется градиент изображения

$$\nabla I(x+u, y+v, t+1) - \nabla I(x, y, t) = 0$$

С самого начала ищутся только небольшие векторы смещения

Метод Brox et al

Рассмотренные методы чувствительны к изменению освещения

Идея

•Рассмотреть другой data term

Предположение

Сохраняется градиент изображения

$$\nabla I(x+u, y+v, t+1) - \nabla I(x, y, t) = 0$$

С самого начала ищутся только небольшие векторы смещения

Идея

Использовать нелинеаризованное уравнение сохранения яркости

Метод Brox et al

Тогда штраф за отклонение от предположений о сохранении яркости и градиента яркости примет вид

$$E_{Data}(u,v) = \int_{\Omega} \left(\left| I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x}) \right|^2 + \gamma \left| \nabla I(\mathbf{x}+\mathbf{w}) - \nabla I(\mathbf{x}) \right|^2 \right) dxdy$$

Метод Brox et al

Проблема:
•квадратичный штраф чувствителен к выбросам

Метод Brox et al

Проблема:
•квадратичный штраф чувствителен к выбросам

Идея
Заменить квадратичный штраф выпуклой возрастающей функцией

$$\Psi(s^2) = \sqrt{s^2 + \varepsilon^2}$$

ε малый положительный параметр

Метод Brox et al

Получим новый функционал

$$E_{Data}(u,v)=\int_{\Omega}\Psi\left(\left|I(\mathbf{x}+\mathbf{w})-I(\mathbf{x})\right|^2+\gamma\left|\nabla I(\mathbf{x}+\mathbf{w})-\nabla I(\mathbf{x})\right|^2\right)dxdy$$

Метод Brox et al

Предположение
Поток кусочно-гладкий

Штраф за отклонения от предположения о гладкости потока

$$E_{Smooth}(u,v) = \int_{\Omega} \Psi\left(\left|\nabla_3 u\right|^2 + \left|\nabla_3 v\right|^2\right) dxdy$$

Метод Brox et al

Задача
Найти u и v, минимизирующие функционал

$$E(u,v) = E_{Data}(u,v) + \alpha \cdot E_{Smooth}(u,v)$$

$$E_{Data}(u,v) = \int_{\Omega} \Psi\left(\left| I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x}) \right|^2 + \gamma \left| \nabla I(\mathbf{x}+\mathbf{w}) - \nabla I(\mathbf{x}) \right|^2 \right) dxdy$$

$$E_{Smooth}(u,v) = \int_{\Omega} \Psi\left(\left| \nabla_3 u \right|^2 + \left| \nabla_3 v \right|^2 \right) dxdy$$

Метод Brox et al

Уравнения Эйлера-Лагранжа
Введем обозначения:

$$
\begin{aligned}
I_x &:= \partial_x I(\mathbf{x}+\mathbf{w}) \\
I_y &:= \partial_y I(\mathbf{x}+\mathbf{w}) \\
I_z &:= I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x}) \\
I_{xx} &:= \partial_{xx} I(\mathbf{x}+\mathbf{w}) \\
I_{yy} &:= \partial_{yy} I(\mathbf{x}+\mathbf{w}) \\
I_{xz} &:= \partial_x I(\mathbf{x}+\mathbf{w}) - \partial_x I(\mathbf{x}) \\
I_{yz} &:= \partial_y I(\mathbf{x}+\mathbf{w}) - \partial_y I(\mathbf{x})
\end{aligned}
$$

Использование *z* вместо *t* говорит о том, что соответствующее выражение **НЕ** производная, а разность, которую будем минимизировать

Метод Brox et al

Уравнения Эйлера-Лагранжа

$$\Psi'(I_z^2+\gamma(I_{xz}^2+I_{yz}^2))\cdot(I_x I_z+\gamma(I_{xx}I_{xz}+I_{xy}I_{yz}))-$$
$$-\alpha\cdot\operatorname{div}(\Psi'(|\nabla_3 u|^2+|\nabla_3 v|^2)\nabla_3 u)=0,$$

$$\Psi'(I_z^2+\gamma(I_{xz}^2+I_{yz}^2))\cdot(I_y I_z+\gamma(I_{yy}I_{yz}+I_{xy}I_{xz}))-$$
$$-\alpha\cdot\operatorname{div}(\Psi'(|\nabla_3 u|^2+|\nabla_3 v|^2)\nabla_3 v)=0$$

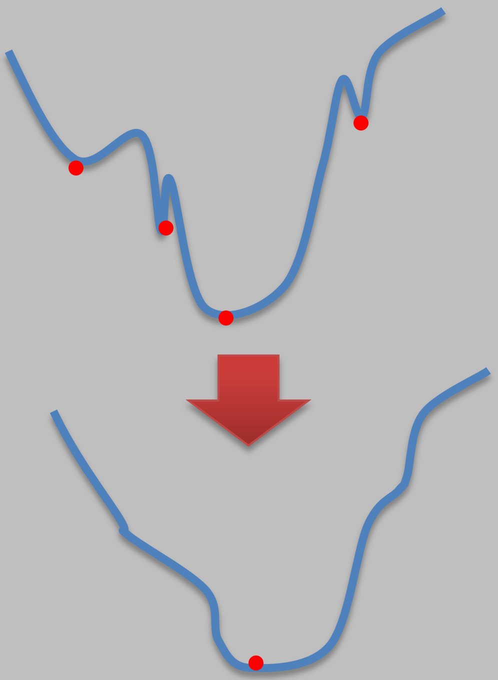
Граничные условия Неймана

Метод Brox et al

Численный метод

•Уравнения Эйлера-Лагранжа нелинейные
•Используем метод неподвижной точки для **w**
•Если есть смещения на расстояние, большее размера пикселя, то функционал может иметь множество локальных минимумов

Идея
•Использовать пирамиду изображений разного разрешения



Метод неподвижной точки

$$f(x) = x$$

$$\Rightarrow x^{n+1} = f(x^n)$$

$$\Rightarrow \lim_{n\to\infty} x^{n+1} = x^*$$

$$\Rightarrow f(x^*) = x^*$$

Метод Brox et al

Численный метод

- Совместим метод неподвижной точки с масштабированием
- Масштабировать будем с коэффициентом $\eta \in (0,1)$
- Для более плавного перехода от разрешения к разрешению коэффициент выберем близким к 1
- Используем полную пирамиду изображений, начиная с наименьшего возможного разрешения (например, с 10x10)

Метод Brox et al

Численный метод
Итерации по разрешению назовем внешними

Пусть k – номер изображения в пирамиде (0 – самое низкое разрешение)

$\mathbf{w}^k = (u^k, v^k, 1)^T$ – решение на k-ой внешней итерации

$$\Psi'((I_z^{k+1})^2 + \gamma((I_{xz}^{k+1})^2 + (I_{yz}^{k+1})^2)) \cdot (I_x^k I_z^{k+1} + \gamma(I_{xx}^k I_{xz}^{k+1} + I_{xy}^k I_{yz}^{k+1})) -$$
$$- \alpha \cdot \mathrm{div}(\Psi'(\left|\nabla_3 u^{k+1}\right|^2 + \left|\nabla_3 v^{k+1}\right|^2)\nabla_3 u^{k+1}) = 0$$

Осталась нелинейность в

- I_*^{k+1}
- Ψ'

Метод Brox et al

Численный метод
Устранение нелинейности в производных изображения

$$I_z^{k+1} \approx I_z^k + I_x^k du^k + I_y^k dv^k$$

$$I_{xz}^{k+1} \approx I_{xz}^k + I_{xx}^k du^k + I_{xy}^k dv^k$$

$$I_{yz}^{k+1} \approx I_{yz}^k + I_{xy}^k du^k + I_{yy}^k dv^k$$

$$u^{k+1} = u^k + du^k, \quad v^{k+1} = v^k + dv^k$$

Метод Brox et al

Численный метод

Введем обозначения

$$(\Psi')^k_{Data} := \Psi'((I^k_z + I^k_x du^k + I^k_y dv^k)^2 +$$
$$+\gamma((I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k)^2 + (I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k)^2))$$

$$(\Psi')^k_{Smooth} := \Psi'(\left|\nabla_3 (u^k + du^k)\right|^2 + \left|\nabla_3 (v^k + dv^k)\right|^2)$$

Метод Brox et al

Численный метод
Аппроксимация слагаемого с дивергенцией

$$\operatorname{div}(k \cdot \nabla u)_{i,j} =$$

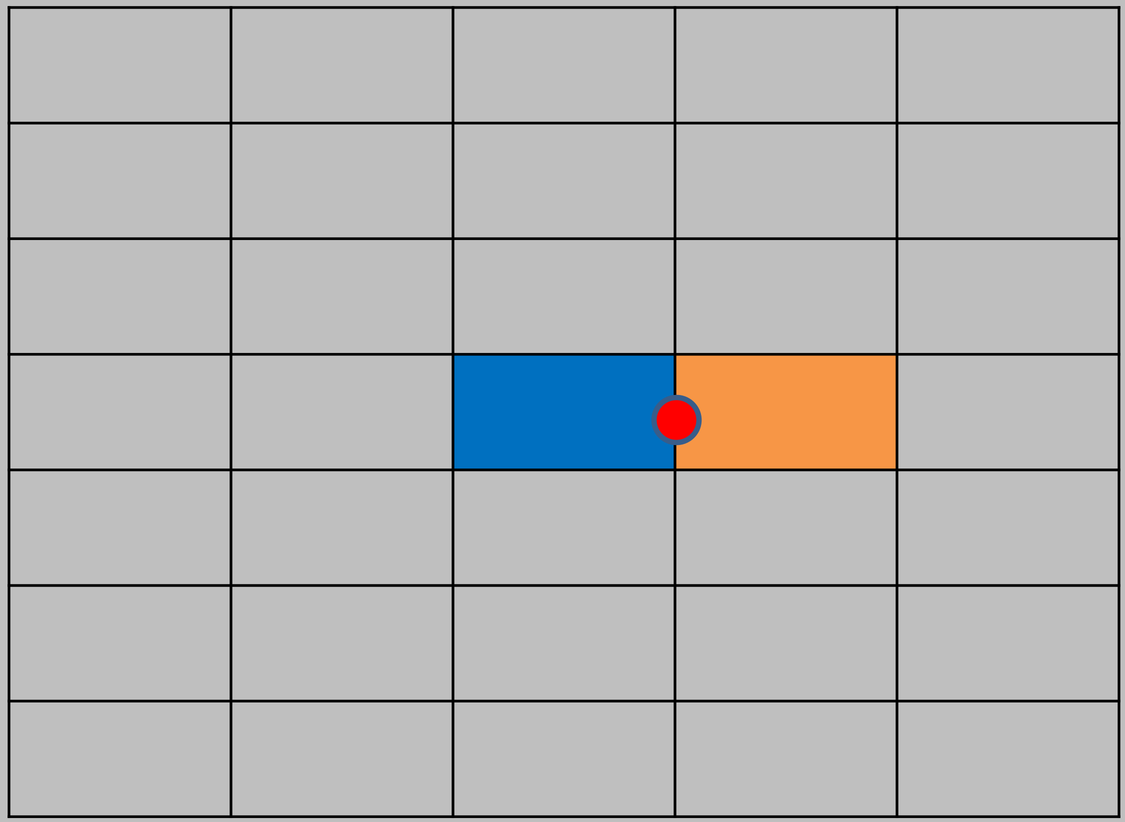
$$= \frac{1}{h}\left(k_{i+\frac{1}{2},j} \frac{u_{i+1,j} - u_{i,j}}{h} - k_{i-\frac{1}{2},j} \frac{u_{i,j} - u_{i-1,j}}{h} \right) +$$

$$+ \frac{1}{h}\left(k_{i,j+\frac{1}{2}} \frac{u_{i,j+1} - u_{i,j}}{h} - k_{i,j-\frac{1}{2}} \frac{u_{i,j} - u_{i,j-1}}{h} \right)$$

Метод Brox et al

Численный метод

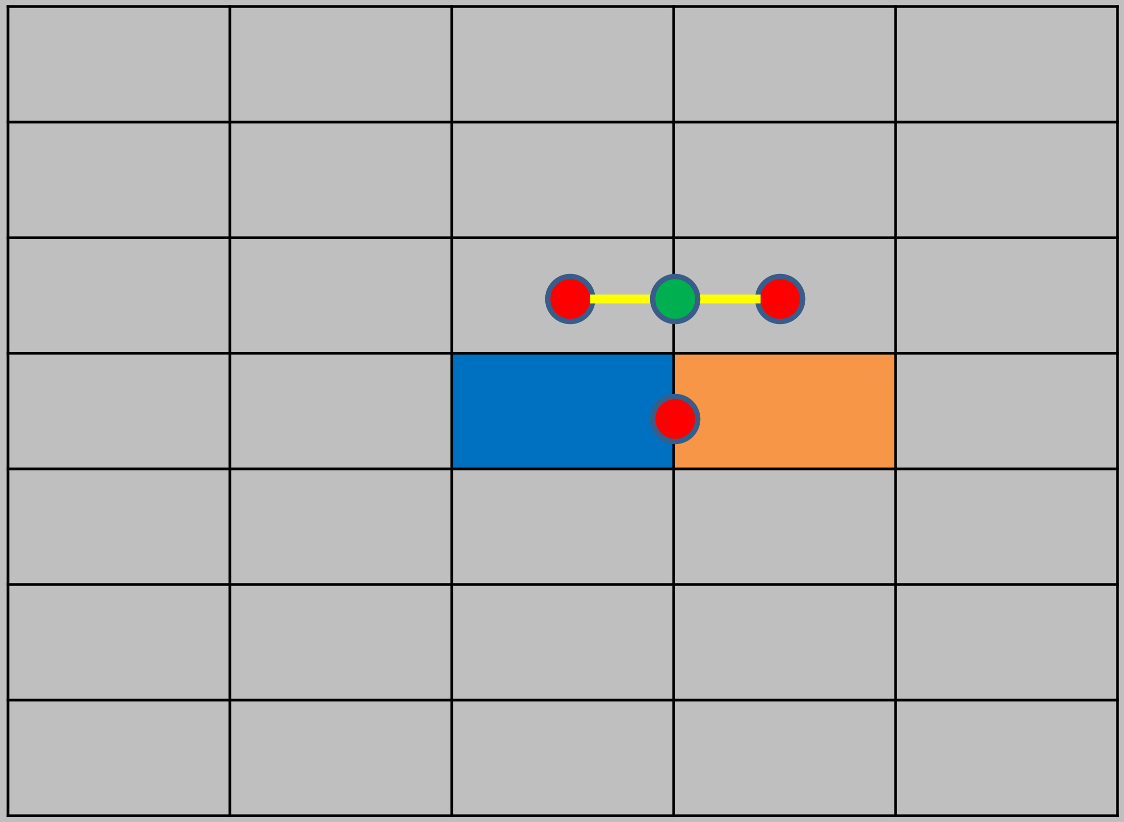
Аппроксимация smooth term



Метод Brox et al

Численный метод

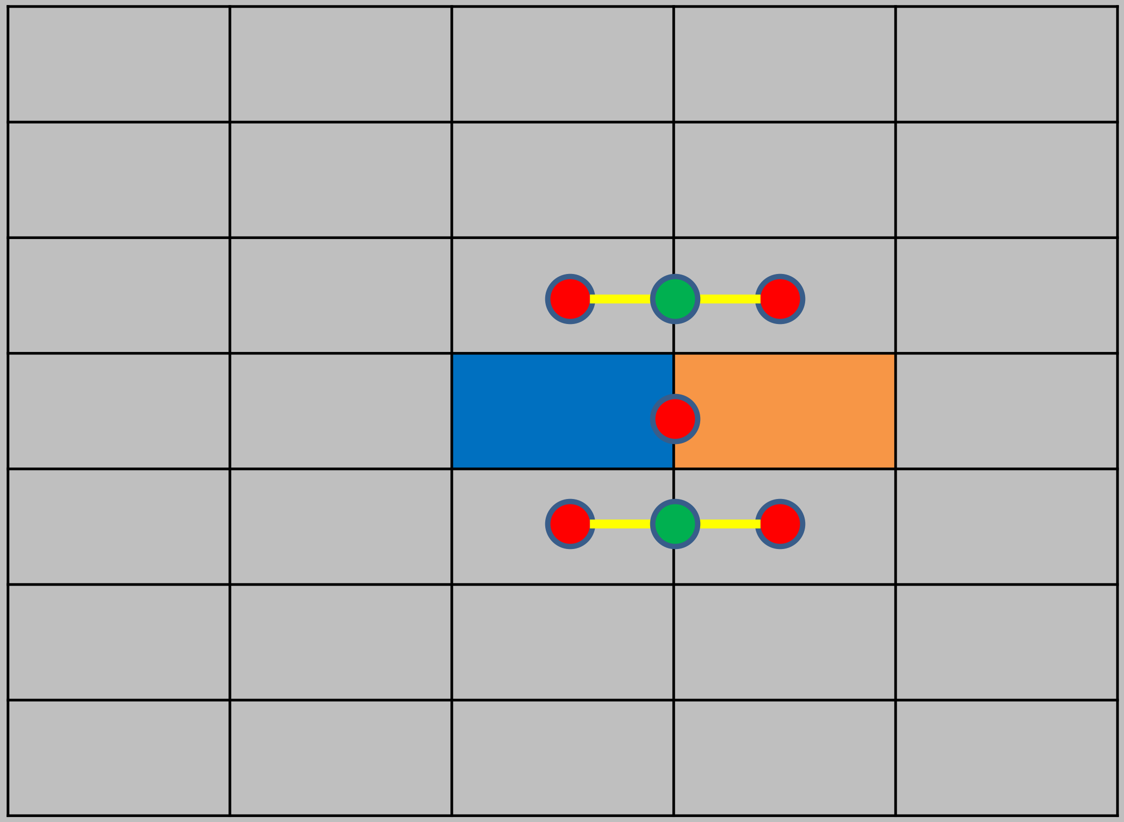
Аппроксимация smooth term



Метод Brox et al

Численный метод

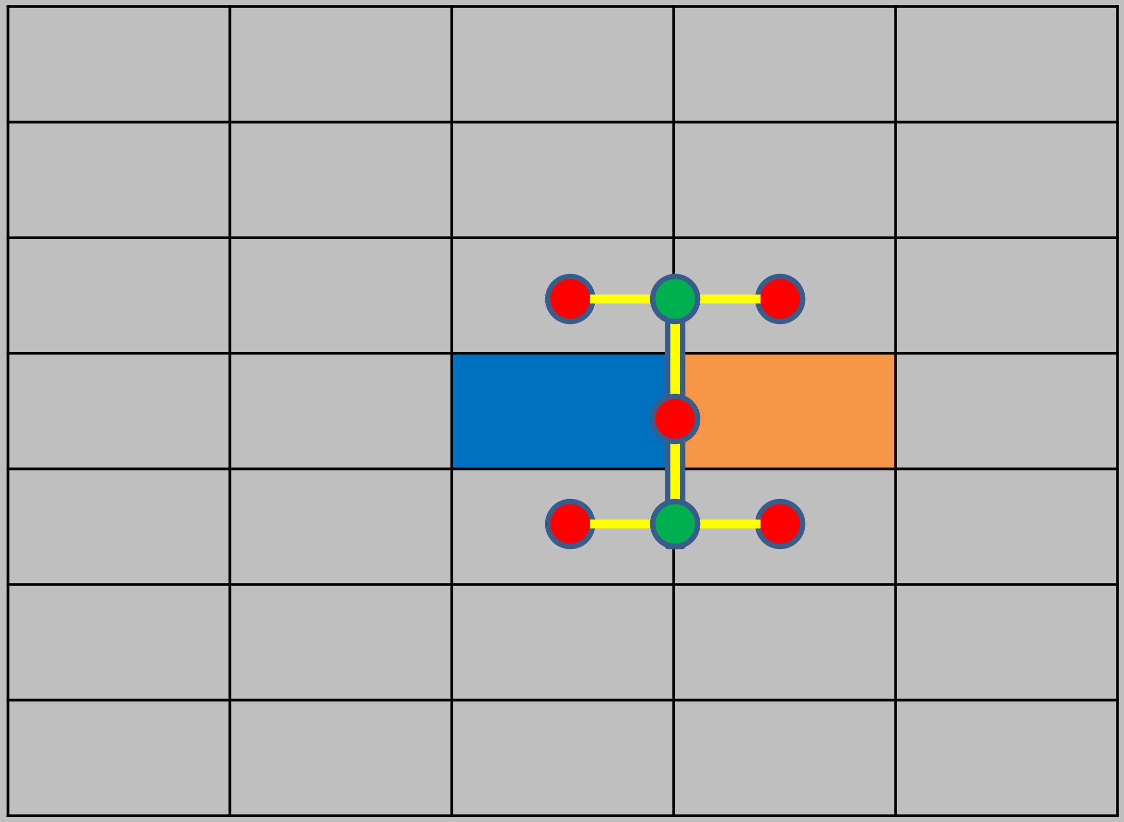
Аппроксимация smooth term



Метод Brox et al

Численный метод

Аппроксимация smooth term



Метод Brox et al

Численный метод

Аппроксимация smooth term

$$\left|\nabla u\right|^2_{i+\frac{1}{2},j} = \left(u_{i+1,j} - u_{i,j}\right)^2 + \left(\left(\frac{u_{i+1,j+1} + u_{i,j+1}}{2}\right) - \left(\frac{u_{i+1,j-1} + u_{i,j-1}}{2}\right)\right)^2$$

$$\left|\nabla u\right|^2_{i,j+\frac{1}{2}} = \left(u_{i,j+1} - u_{i,j}\right)^2 + \left(\left(\frac{u_{i+1,j+1} + u_{i+1,j}}{2}\right) - \left(\frac{u_{i-1,j+1} + u_{i-1,j}}{2}\right)\right)^2$$

Метод Brox et al

Численный метод
Оставшаяся нелинейность в Ψ устраняется повторным применением метода неподвижной точки

Начальные значения $\mathbf{w}^0 = (0,0,1)^T, \; du^{k,0} = 0, \; dv^{k,0} = 0$

$$
\begin{aligned}
0 = (\Psi')^{k,l}_{Data} \cdot (I^k_x (I^k_z + I^k_x du^{k,l+1} + I^k_y dv^{k,l+1}) + \\
+ \gamma \cdot I^k_{xx} (I^k_{xz} + I^k_{xx} du^{k,l+1} + I^k_{xy} dv^{k,l+1}) + \gamma \cdot I^k_{xy} (I^k_{yz} + I^k_{xy} du^{k,l+1} + I^k_{yy} dv^{k,l+1})) - \\
- \alpha \cdot \mathrm{div}\left((\Psi')^{k,l}_{Smooth} \nabla_3 (u^k + du^{k,l+1})\right)
\end{aligned}
$$

Полученная **система уравнений**
- **линейная**
- может быть решена одним из рассмотренных методов

Метод Brox et al

Решение линейной системы

Используем красно-черный метод релаксации (**Red-Black SOR**)

$$du_i^{k,l,m+1} = (1-\omega)\underline{du_i^{k.l,m}} +$$

$$+\omega \frac{\sum_{j\in Ш'(i)} (\Psi'_S)^{k,l}_{i\leftrightarrow j} (u_j^k + du_j^{k,l,m}) - \sum_{j\in Ш'(i)} (\Psi'_S)^{k,l}_{i\leftrightarrow j} u_i^k}{\sum_{j\in Ш'(i)} (\Psi'_S)^{k,l}_{i\leftrightarrow j} + \frac{(\Psi'_D)^{k,l}_i}{\alpha}((I^k_{x,i})^2 + \gamma\cdot((I^k_{xy,i})^2 + (I^k_{xx,i})^2))} -$$

$$-\omega \frac{\frac{(\Psi'_D)^{k,l}_i}{\alpha}((I^k_{x,i} I^k_{y,i} + \gamma\cdot(I^k_{xx,i} I^k_{xy,i} + I^k_{xy,i} I^k_{yy,i}))dv_i^{k,l,m} + I^k_{x,i} I^k_{z,i} + \gamma\cdot(I^k_{xx,i} I^k_{xz,i} + I^k_{xy,i} I^k_{yz,i}))}{\sum_{j\in Ш'(i)} (\Psi'_S)^{k,l}_{i\leftrightarrow j} + \frac{(\Psi'_D)^{k,l}_i}{\alpha}((I^k_{x,i})^2 + \gamma\cdot((I^k_{xy,i})^2 + (I^k_{xx,i})^2))}$$

Граничные условия

$$du^{k,l,m}\Big|_{\partial\Omega} = 0,\ dv^{k,l,m}\Big|_{\partial\Omega} = 0$$

Метод Brox et al

Решение линейной системы
Используем красно-черный метод релаксации (**Red-Black SOR**)

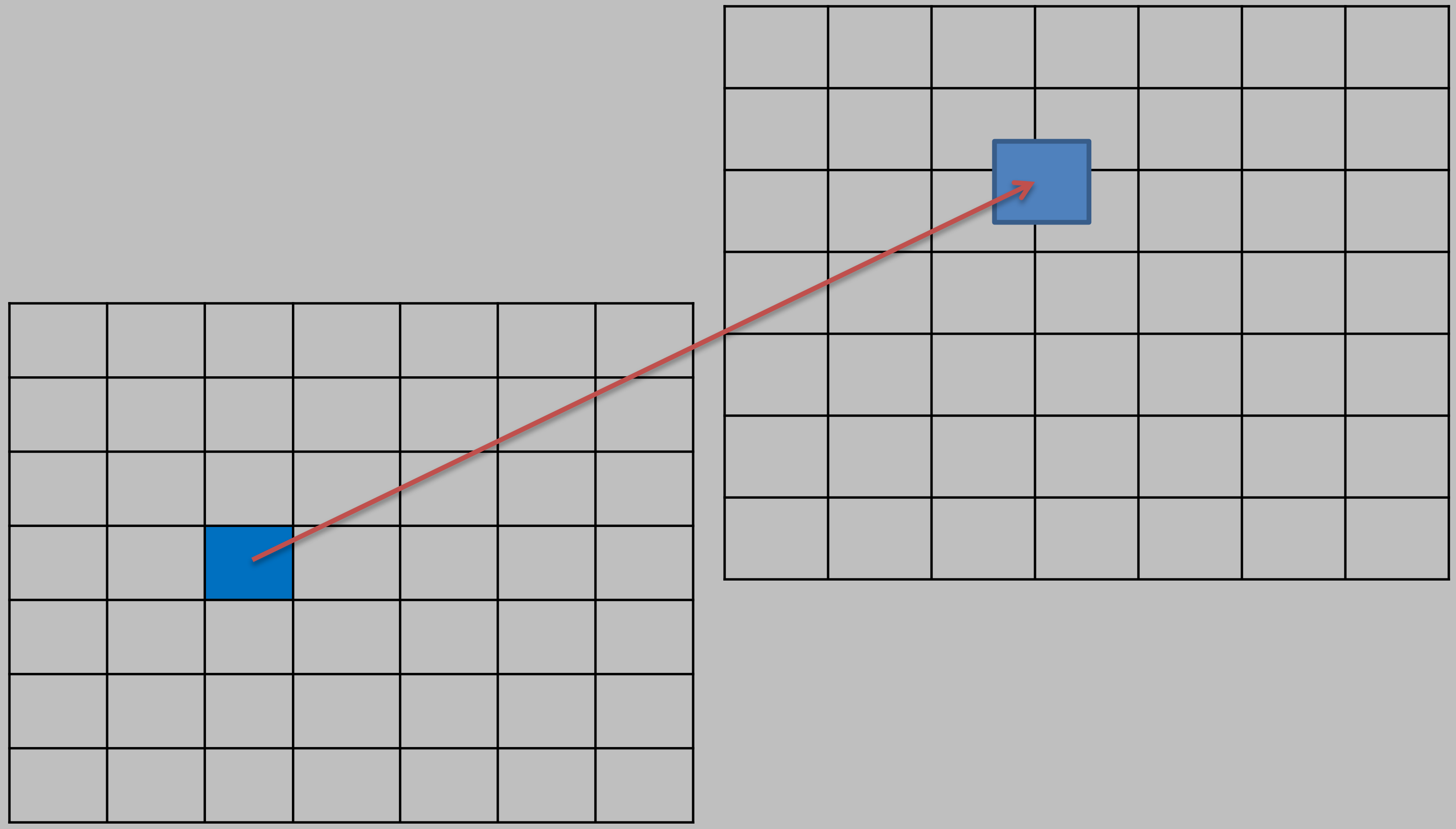
$$du_i^{k,l,m+1} = (1-\omega)\,du_i^{k.l,m} +$$

$$+\,\omega\frac{\sum_{j\in Ш'(i)}(\Psi'_S)^{k,l}_{i\leftrightarrow j}(u_j^k + du_j^{k,l,m}) - \sum_{j\in Ш'(i)}(\Psi'_S)^{k,l}_{i\leftrightarrow j}u_i^k}{\boxed{\sum_{j\in Ш'(i)}(\Psi'_S)^{k,l}_{i\leftrightarrow j} + \frac{(\Psi'_D)_i^{k,l}}{\alpha}((I_{x,i}^k)^2 + \gamma\cdot((I_{xy,i}^k)^2 + (I_{xx,i}^k)^2))}} -$$

$$-\,\omega\frac{\frac{(\Psi'_D)_i^{k,l}}{\alpha}\left(\left(\boxed{I_{x,i}^k I_{y,i}^k + \gamma\cdot(I_{xx,i}^k I_{xy,i}^k + I_{xy,i}^k I_{yy,i}^k)}\right)dv_i^{k,l,m} + \boxed{I_{x,i}^k I_{z,i}^k + \gamma\cdot(I_{xx,i}^k I_{xz,i}^k + I_{xy,i}^k I_{yz,i}^k)}\right)}{\boxed{\sum_{j\in Ш'(i)}(\Psi'_S)^{k,l}_{i\leftrightarrow j} + \frac{(\Psi'_D)_i^{k,l}}{\alpha}((I_{x,i}^k)^2 + \gamma\cdot((I_{xy,i}^k)^2 + (I_{xx,i}^k)^2))}}$$

Метод Brox et al

Билинейная интерполяция для **I(x+w)**



Метод Brox et al

Реализация

•Кадры храним в виде текстур

•Для продолжения решения с разрешения k на разрешение k+1 используем текстуры

Общая схема метода

Подготовить текстуры с изображениями

Задать начальное значения для u и v

- Для каждого разрешения, начиная с наименьшего
    - Установить начальное значение du, dv
        - Подготовить данные для итераций SOR
        - Выполнить некоторое число итераций SOR
        - Обновить u, v
    - Продолжить решение на большее разрешение
- Сохранить результат

Метод Brox et al

Реализация

•Кадры храним в виде текстур

•Для продолжения решения с разрешения k на разрешение k+1 используем текстуры

Общая схема метода

Подготовить текстуры с изображениями

Задать начальное значения для u и v

- Для каждого разрешения, начиная с наименьшего
  - Установить начальное значение du, dv
    - Подготовить данные для итераций SOR
    - Выполнить некоторое число итераций SOR
    - Обновить u, v
  - Продолжить решение на большее разрешение
- Сохранить результат

warping FP iteration

lagged nonlinearity FP iteration

solver FP iteration

Метод Brox et al

- Сложные шаблоны доступа к памяти
- Большая часть данных не меняется во время итераций

Используем текстуры

Текстуры + shared memory работает медленнее чем просто текстуры

Для объединения запросов при записи индексы начала строк округляем до значений, кратных 16

?

Дополнительные материалы:

- http://www.google.com
- http://www.mia.uni-saarland.de
- http://vision.middlebury.edu/flow/