

## Многокадровый оптический поток на основе траекторий

Михаил Синдеев<sup>1,2</sup>, Антон Конушин<sup>2</sup>, Карстен Ротер<sup>3</sup>

<sup>1</sup> Институт прикладной математики им. М.В. Келдыша РАН, Москва, Россия

<sup>2</sup> Факультет вычислительной математики и кибернетики, МГУ имени М.В. Ломоносова, Москва, Россия

<sup>3</sup>Microsoft Research Cambridge

{msindeev, ktosh}@graphics.cs.msu.ru, carrot@microsoft.com

### Аннотация

В этой статье предлагается новый алгоритм вычисления оптического потока, использующего соседние кадры в качестве вспомогательных для точного анализа перекрытий. Это позволяет оценить движение в областях, которые видны лишь в одном из двух последовательных кадров. Кроме того, может быть смоделировано нелинейное движение пикселей между двумя кадрами для более точной генерации промежуточных кадров в задаче интерполяции.

**Ключевые слова:** оптический поток, траектории, перекрытия, карта видимости.

### 1. ВВЕДЕНИЕ

Пусть требуется построить карту соответствия между двумя изображениями, являющимися соседними кадрами видеопоследовательности  $I_0$  и  $I_1$ . С учетом погрешностей регистрации изображений (шум CCD-матрицы, перекрытия, артефакты дискретизации), соответствие между изображениями можно записать как

$$I_0(x, y) \approx I_1(x + V_x(x, y), y + V_y(x, y)), \quad (1)$$

где  $V_x$ ,  $V_y$  – вертикальная и горизонтальная компоненты видимого движения пикселей между кадрами, называемого оптическим потоком  $V$ . Для нецелых значений сдвига используется билинейная интерполяция. Поскольку для каждого пикселя  $(x, y)$  изображения  $I_0$  можно найти много пикселей в изображении  $I_1$  с совпадающим или близким цветом, решение для  $V$  не единственно. Поэтому задачу нахождения оптического потока обычно формулируют как минимизацию функционала, состоящего из члена данных  $E_D$  и члена гладкости  $E_S$ , умноженного на коэффициент регуляризации  $\lambda > 0$  (см. [4]).

$$E(V) = E_D(V) + \lambda E_S(V), \quad (2)$$

$$E_D(V) = \iint (I_0(x, y) - I_1(x + V_x, y + V_y))^2 dx dy \quad (3)$$

$$E_S(V) = \iint (\|\nabla V_x\|^2 + \|\nabla V_y\|^2) dx dy, \quad (4)$$

$$V = \arg \min_V E(V) \quad (5)$$

Возможны и другие способы регуляризации, использующие неквадратичные нормы для данных слагаемых [10], например:

$$E_D(V) = \iint \psi(I_0(x, y) - I_1(x + V_x, y + V_y)) dx dy \quad (6)$$

$$E_S(V) = \iint \psi(\nabla V) dx dy, \quad (7)$$

$$\psi(z) = \sqrt{z^2 + \sigma^2} \text{ для малого } \sigma. \quad (8)$$

Основная цель такой регуляризации – учесть перекрытия и разрывы в потоке.

Однако робастные нормы лишь примерно интерполируют поток в области перекрытий, используя значения из соседних пикселей. Выбор подходящей нормы в слагаемом гладкости позволяет добиться правдоподобного профиля интерполяции для определенных классов объектов (например, для объектов с гладким контуром или объектов, выделяющихся по цвету – в этом случае норма также зависит от изображения  $I_0$ ).

Некоторые методы поиска перекрытий помимо прямого потока вычисляют обратный поток  $V'$  от  $I_1$  к  $I_0$  и сравнивают их, помечая пиксели  $|V + V'| > \delta$  как перекрытия для некоторого фиксированного порога  $\delta$  (обратный поток берется со знаком «+», т.к. он в сумме с прямым потоком в идеале должен давать 0), либо используют фиксированный штраф для перекрытых пикселей [13].

Иногда принимается во внимание пространственная разреженность карты перекрытий, например, путем фильтрации полученной бинарной карты [13].

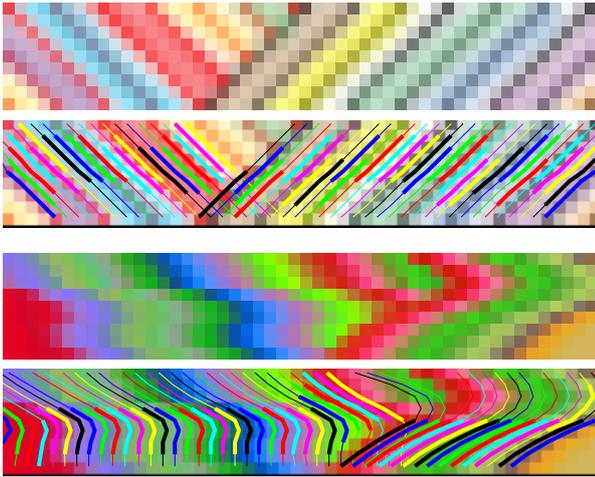
Однако в силу неоднозначности потока возможны случаи, когда при похожих цветах объектов в изображениях слагаемое данных получается слабым и поток  $V$ ,  $V'$  различаются лишь из-за регуляризации, которая оптимизируется независимо для прямого и обратного потока. Возможна формулировка совместной минимизации с общей регуляризацией, но она не улучшает ситуацию, т.к. результирующий функционал имеет много локальных минимумов.

Другая проблема – выбор порога отсекающего перекрытий, т.к. количество перекрытых пикселей может существенно различаться в разных кадрах. Одним из решений является добавление в задачу неизвестной бинарной маски перекрытий и попиксельное умножение слагаемого данных  $E_D$  на эту маску. Такой подход приводит к тому, что глобальный минимум  $E$  достигается при отметке всех пикселей как перекрытых. Попытка выбора оптимального штрафа за избыток перекрытий равносильна проблеме поиска порога  $\delta$ , т.е. не может быть решена сразу для всех кадров.

В данной статье рассматривается возможность уточнения слагаемого данных за счет использования соседних кадров с явным вычислением карты видимости. Наш подход основан на представлении многокадрового потока в виде траекторий [12] и идее моделирования перекрытий с помощью сортирующего суммирования, применяемой в алгоритмах многокадрового стереосопоставления [9], [5].

### 2. ПРЕДЛАГАЕМЫЙ ПОДХОД

Вместо пространственной разреженности перекрытий, применяемой в двухкадровых алгоритмах оптического потока [13], сосредоточимся на их временной разреженности. Временная разреженность означает, что каждая точка сцены попадает в перекрытие не слишком часто. Это условие является более естественным, чем пространственная разреженность в отдельно взятом кадре, т.к. нельзя заранее сказать, сколько точек в кадре перекрыто, в то время как частоту перекрытий во времени можно примерно оценить снизу.



**Рисунок 1.** Искусственные примеры. Исходные одномерные видеопоследовательности ( $T = 9$  кадров, по вертикали – ось времени) и найденные траектории. Толстые части траекторий – диапазоны видимости.

Предположим, что каждая точка в изображении  $I_0$  видима хотя бы на протяжении 5 кадров. Тогда, рассматривая 9 последовательных изображений  $I_{-4}, \dots, I_4$  можно ввести переменную, отвечающую за перекрытие  $\rho \in \{0, 1, 2, 3, 4\}$ .

При этом  $\rho = 0$  означает, что данный пиксел является видимым в кадрах  $I_{-4}, \dots, I_0$ ,  $\rho = 1$  – в кадрах  $I_{-3}, \dots, I_1$  и т.д.,  $\rho = 4$  – в кадрах  $I_0, \dots, I_4$ . Обратим внимание, что в такой формулировке пиксел всегда виден в центральном кадре  $I_0$ . Карта видимости  $\rho$  по координатам  $(x, y)$  соответствует именно этому кадру (в то время как координаты пиксела в других кадрах зависят от потока между  $I_0$  и данным кадром).

В общем случае мы рассматриваем произвольное нечетное число кадров  $T \geq 3$ . Для обозначения диапазона индексов введем значение  $K = \frac{T-1}{2}$ . Пиксел является видимым по

меньшей мере в  $K+1$  изображении из  $T$  изображений  $I_{-K}, \dots, I_K$ . Карта видимости:  $\rho \in \{0, \dots, K\}$ , пиксел является видимым в кадрах  $\rho - K, \dots, K$ . Пример траекторного потока приведен на рис. 1.

Заметим, что в такой формулировке не ставится вопрос о наличии или отсутствии перекрытия в каком-либо пикселе какого-либо кадра, поэтому не придется выбирать функцию штрафа за чрезмерные перекрытия. Мы лишь выбираем набор из  $K+1$  последовательных кадров, в которых пиксел точно является видимым (видимость в остальных  $K$  кадрах может быть произвольной).

Для работы с пикселями в такой формулировке нам требуется  $T - 1$  (или, что то же самое,  $2K$ ) двумерных оптических потоков – из кадра  $I_0$  в каждый из остальных кадров. Обозначим эти потоки как  $V_{-K}, \dots, V_{-1}, V_1, \dots, V_K$ . Если бы они были последовательными ( $I_1 \rightarrow I_2, I_2 \rightarrow I_3$  и т.д.), для вычисления позиции пиксела требовалось бы интегрирование нескольких потоков, что сильно усложняет формулировку и снижает точность из-за погрешностей интерполяции. Поэтому поток  $V_k$  отвечает за трансформацию между кадрами  $I_0 \rightarrow I_k, k = -K, \dots, -1, 1, \dots, K$ .

Выходным результатом алгоритма считается поток  $V_1$ , и, если нужно, обратный поток в предыдущий кадр  $V_{-1}$ .

Остальные потоки являются лишь вспомогательными для уточнения перекрытий. Также для выходных потоков можно вычислить бинарную маску перекрытий как  $[\rho = 0]$  и  $[\rho = K]$  соответственно (т.е. случаи, когда не виден кадр  $I_1$  и  $I_{-1}$  соответственно). Квадратными скобками обозначена индикаторная функция.

Т.к. потоки вычисляются относительно кадра  $I_0$ , их можно считать одним «траекторным» потоком, где каждому пикселу сопоставлен  $4K$ -мерный (или, что то же самое,  $2(T - 1)$ -мерный) вектор потока

$$\vec{V} = (V_{-K,x}, V_{-K,y}, \dots, V_{-1,x}, V_{-1,y}, V_{1,x}, V_{1,y}, \dots, V_{K,x}, V_{K,y}) \quad (9)$$

Такое векторное представление позволяет использовать любые известные функционалы гладкости, определенные для двумерного потока, которые могут быть сформулированы в терминах векторной алгебры, а также многие алгоритмы минимизации данных функционалов. Исключением будут алгоритмы, явно использующие факт двумерности, а также алгоритмы, несовместимые с членом данных  $E_D$ , который будет сформулирован в следующем разделе (однако их часто можно адаптировать для данной задачи, либо использовать методы раздельной оптимизации, используя другой алгоритм для члена данных). Кроме того, стохастические и переборные алгоритмы могут потерять свою эффективность из-за увеличения пространства поиска.

## 2.1 Функционал энергии

Задача поиска траекторного потока может быть сформулирована как задача минимизации функционала энергии

$$E(V, \rho) = E_D(V, \rho) + \lambda E_S(V) + \mu E_T(V) + \nu E_\rho(\rho) \quad (10)$$

где  $E_D$  – член данных,  $E_S$  – член пространственной гладкости потока,  $E_T$  – член временной гладкости потока,  $E_\rho$  – член гладкости карты видимости,  $\lambda, \mu, \nu$  – веса слагаемых.

$$E_D(V, \rho) = \frac{1}{K} \sum_x \sum_y \sum_{\substack{\rho(x,y) \\ t \neq 0}} \left( I_0(x, y) - I_t(x + V_{t,x}, y + V_{t,y}) \right)^2 \quad (11)$$

В этом члене не используется робастная норма (8) или аналогичная, как в формуле (6), т.к. за перекрытия явно определяются значением  $\rho$ , и цвета не перекрытых пикселей должны совпадать с высокой точностью.

Член пространственной гладкости соответствует случаю двухкадрового потока (7) с некоторыми модификациями:

$$E_S(V) = \frac{1}{T-1} \sum_x \sum_y \left[ \sum_{\substack{t=-K \\ t \neq 0}}^K \left( \frac{|\nabla V|}{|t|} \right)^2 \right]^\gamma \quad (12)$$

В качестве робастной нормы взята функция, инвариантная к повороту. Степень  $\gamma = 0,5$  соответствует робастной норме  $L_1$ , инвариантной к повороту, которая определяется как

$$\psi(V) = \sqrt{\left( \frac{\partial V_x}{\partial x} \right)^2 + \left( \frac{\partial V_x}{\partial y} \right)^2 + \left( \frac{\partial V_y}{\partial x} \right)^2 + \left( \frac{\partial V_y}{\partial y} \right)^2}, \quad (13)$$

причем в нашем случае суммирование по  $t$  также производится под знаком радикала. Это означает, что в случае разрывности решения разрыв потока в некоторой точке происходит одновременно в горизонтальной и вертикальной

компонентах потока и сразу во всех кадрах. Таким образом, траектории в двух соседних пикселах не могут разойтись, а через несколько кадров слиться в единый объект.

На практике мы используем значение  $\gamma = 0,45$ , как рекомендовано в статье [10], порождающее невыпуклую норму, которая «поощряет» более четкие края в случае разрывного потока.

Деление на  $|t|$  в формуле (12) уравнивает разброс значений по разным кадрам, т.к. он возрастает при удалении от центрального кадра.

Энергия временной гладкости использует вторую производную потока по времени, т.е. поощряет траектории, близкие к линейным:

$$E_T(V) = \frac{1}{T-2} \sum_x \sum_y \sum_{t=-K+1}^{K-1} (V_{t-1} - 2V_t + V_{t+1})^2, \quad (14)$$

при этом полагается  $V_0 = 0$ .

Коэффициенты  $1/K$ ,  $1/(T-1)$ ,  $1/(T-2)$  в формулах (11), (12), (14) соответствуют количеству слагаемых на пиксел, зависящему от числа рассматриваемых кадров  $T$ , что упрощает подгонку коэффициентов  $\lambda$ ,  $\mu$ ,  $\nu$  при изменении числа кадров.

Функционал гладкости карты видимости также является квадратичным:

$$E_\rho(\rho) = \sum_x \sum_y (\nabla \rho(x, y))^2 \quad (15)$$

При этом градиент означает конечную разность ввиду дискретности значений  $\rho$ . Выбор квадратичной функции штрафа обусловлен структурой перекрытий в видео: движущийся объект оставляет за собой «след» из перекрытий, каждое из которых сдвинуто на 1 кадр относительно предыдущего. Скачки в карте видимости, превышающие 1 кадр, чаще всего соответствуют неправильно найденным перекрытиям. Надежными перекрытиями являются такие, которые последовательно возникают по пути движения объекта, поэтому имеет смысл искать перекрытия именно с такой структурой.

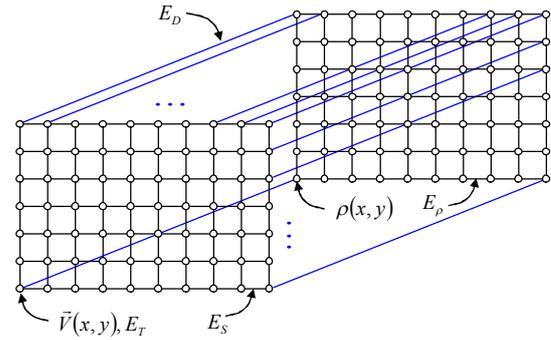
Также можно ввести два дополнительных члена, связанных с картой видимости.

Первый член запрещает выход видимой части траектории за края изображения, т.к. обращает энергию в бесконечность. Таким образом, данная ситуация рассматривается как обычное перекрытие, в отличие от алгоритмов двухкадрового потока, где такие случаи приходится обрабатывать отдельно, чаще всего обнулением производных для векторов потока, выходящих за края изображения.

Второй дополнительный член поощряет, с очень маленьким весом, близость  $\rho$  к значению  $K/2$ , т.е. симметрию диапазона видимости. Такая регуляризация улучшает маску видимости для выходного потока  $V_t$ , т.к. иначе в простых случаях при отсутствии перекрытий может быть выбран диапазон видимости  $-K, \dots, 0$  и выходной поток будет менее информативным.

### 3. МИНИМИЗАЦИЯ

В данном разделе приводится предлагаемый алгоритм минимизации полного функционала энергии (10). Важным аспектом является совместная минимизация по потоку и карте видимости. При поочередной минимизации возможны локальные минимумы [11].



**Рисунок 2.** Структура графа, позволяющая описать функционал энергии (10) в виде суммы унарных и парных потенциалов на двух двумерных сетках (для  $\vec{V}$  и  $\rho$ ).

Идея алгоритма заключается в использовании метода QRBO [8], минимизирующего энергию для бинарных задач, и алгоритма слияния [6], использующего QRBO для последовательной склейки решений-кандидатов для небинарных задач. При этом сама задача склейки является бинарной.

Будем осуществлять минимизацию на двумерной сетке, соответствующей изображению  $I_0$ . Каждому узлу сетки соответствует вектор потока  $\vec{v}$  (см. обозначение (9)) и значение  $\rho$ . Можно строить решения-кандидаты, состоящие из пар  $(\vec{v}, \rho)$ , но можно сделать их независимыми, разместив их на двух сетках (рис. 2).

Первая сетка содержит пространственные ребра, отвечающие за слагаемое  $E_S$ . Узлы содержат унарный потенциал  $E_T$ . Ребра второй сетки отвечают за слагаемое  $E_\rho$ . Ребра, соединяющие соответственные узлы первой и второй сеток, отвечают за слагаемое  $E_D$ .

Инвариантные к повороту нормы не могут быть заданы с помощью ребер двумерной сетки, поэтому используется аппроксимация – каждая вершина соединяется не с 4, а с 16 ближайшими [3].

#### 3.1 Начальное приближение

В качестве начального приближения вычислим двухкадровые потоки между последовательными кадрами в прямом и обратном направлении от центрального кадра:  $I_0 \rightarrow I_1, I_1 \rightarrow I_2, \dots; I_0 \rightarrow I_{-1}, I_{-1} \rightarrow I_{-2}, \dots$ . Затем выполним рекурсивную трансформацию этих потоков, чтобы получить потоки между центральным кадром и остальными кадрами. Для вычисления двухкадровых потоков использовалась библиотека mexOpticalFlow [7].

#### 3.2 Решения-кандидаты

Будем генерировать решения-кандидаты, используя несколько различных эвристик, и сливать их с текущим решением методом QRBO. Алгоритма слияния [6] гарантирует невозрастание энергии, что снижает требования к решениям-кандидатам.

Решения-кандидаты для траекторного потока  $\vec{V}$ :

1. Всевозможные сдвиги всей сетки на  $\pm 1$  пиксел по каждой оси (т.е. значение потока  $\vec{V}$  копируется из соседнего пиксела, компоненты вектора при этом не меняются)
2. Выбор траектории  $\vec{V}$  в случайном пикселе  $(x, y)$  и использование его в качестве константного кандидата. Т.к. сцены часто состоят из нескольких объектов,

вместо равномерного распределения для  $(x, y)$  сначала осуществляется сегментация потока методом «к средних», затем случайно выбирается сегмент, а в нем – случайный пиксел. Это нужно, чтобы объекты большей площади не имели преимущества.

3. Двухмерное гауссово размытие потока  $\vec{V}$
4. Один шаг градиентного спуска для слагаемых, отвечающих за данные на двумерной сетке (т.е. без учета пространственной гладкости):  $E_D(V, \rho) + \mu E_T(V)$
5. Попиксельная минимизация  $E_D(V, \rho) + \mu E_T(V)$  по  $\vec{V}$  методом динамического программирования. Используется ускоренный приближенный вариант, основанный на идее алгоритма PatchMatch [2].

Как видно, часть решений-кандидатов соответствует копированию «удачных» траекторий  $\vec{V}$  из одних пикселей в другие, другая часть генерирует новые решения в окрестности текущего решения.

Кандидаты для карты видимости являются константами  $\rho = 0, \dots, K$  (также были опробованы сдвиги и увеличение/уменьшение  $\rho$  на 1, но эти варианты не дают существенного улучшения по сравнению с использованием констант).

Кандидаты для всех переменных  $(\vec{V}, \rho)$  строятся в виде декартова произведения описанных кандидатов (для каждого кандидата  $\vec{V}$  пробуем каждый кандидат  $\rho$ ). Т.к. кандидаты зависят от текущего решения, цикл оптимизации надо повторять до тех пор, пока полная энергия не перестанет уменьшаться.

### 3.3 Иерархический подход

Аналогично двумерным методам оптического потока, возможно применение иерархического подхода с уменьшением размера всех кадров до небольшого размера (например, 40x30 пикселей), а затем поэтапным увеличением до исходного размера ([10], [7]). При этом, после каждого уровня иерархии можно выполнить дополнительное слияние: решение с предыдущего уровня плюс двухмерный поток для данного разрешения, преобразованный в траектории (как описано в разделе 3.1).

## 4. РЕЗУЛЬТАТЫ

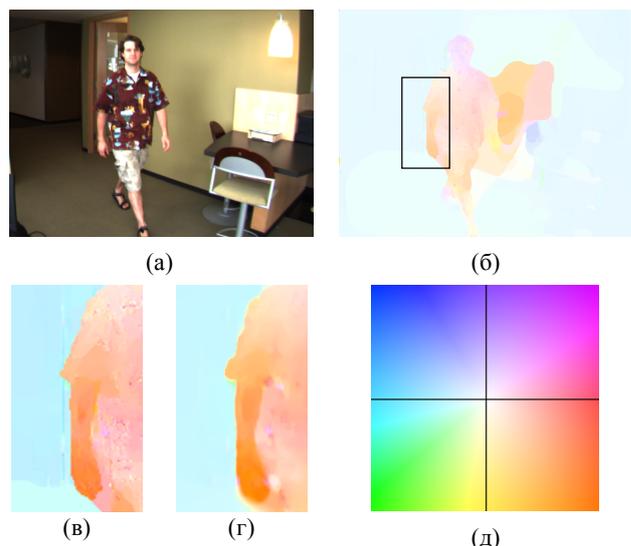
Пример результата работы алгоритма приведен на рис. 3. По сравнению с результатом алгоритма [7] контур объекта более точный и не содержит гладких переходов между объектом и фоном, которые видны в виде белой окантовки на рис. 3 (г). Это достигается за счет построения карты видимости и совместной оптимизации видимости и потока.

## 5. ЗАКЛЮЧЕНИЕ

В статье предложен алгоритм вычисления оптического потока на основе траекторий, дающий стабильный результат при наличии перекрытий за счет использования нескольких последовательных кадров. В дальнейшем планируется провести подробное экспериментальное сравнение метода с аналогами и получить численные оценки

## 6. БЛАГОДАРНОСТИ

Работа выполнена при поддержке проекта МРЛ-2010-050 с компанией Microsoft Research.



**Рисунок 3.** Пример результата. (а) изображение  $I_0$  из тестовой базы [1],  $T = 7$  кадров, (б) поток  $V_1$ , (в) увеличенный фрагмент, выделенный рамкой в (б), (г) результат алгоритма [7], (д) цветовая карта обозначения потока, предложенная в статье [1].

## 7. ССЫЛКИ

- [1] Baker S., Scharstein D., Lewis J., Roth S., Black M., Szeliski R., *A database and evaluation methodology for optical flow*, IJCV, 92(1):1–31, 2011.
- [2] Barnes C., Shechtman E., Finkelstein A., Goldman D., *PatchMatch: a randomized correspondence algorithm for structural image editing*, SIGGRAPH 2009
- [3] Boykov Y., Kolmogorov V., *Computing geodesics and minimal surfaces via graph cuts*, ICCV 2003
- [4] Horn B., Schunck B., *Determining optical flow*, Artificial Intelligence, vol 17, pp 185–203, 1981
- [5] Kang S., Szeliski R., *Extracting view-dependent depth maps from a collection of images*, International Journal of Computer Vision (IJCV), vol. 58, number 2, pp. 139–163, 2004
- [6] Lempitsky V., Roth S., Rother C., *FusionFlow: discrete-continuous optimization for optical flow estimation*, CVPR 2008
- [7] Liu C., *Beyond pixels: exploring new representations and applications for motion analysis*, Doctoral Thesis. Massachusetts Institute of Technology, 2009
- [8] Rother C., Kolmogorov V., Lempitsky V., Szummer M., *Optimizing binary MRFs via extended roof duality*, CVPR 2007
- [9] Satoh K., Ohta Y., *Occlusion detectable stereo – systematic comparison of detection algorithms*, ICPR 1996
- [10] Sun D., Roth S., Black M., *Secrets of optical flow estimation and their principles*, CVPR 2010
- [11] Sun D., Sudderth E., Black M., *Layered segmentation and optical flow estimation over time*, to appear: CVPR 2012
- [12] Volz S., Bruhn A., Valgaerts L., Zimmer H., *Modeling temporal coherence for optical flow*, ICCV 2011
- [13] Xiao J., Cheng H., Sawhney H., Rao C., Isnardi M., *Bilateral filtering-based optical flow estimation with occlusion detection*, ECCV 2006