УДК 621.391

СТАБИЛИЗАЦИЯ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЕЙ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

А.С.ПЧЕЛКИН, О.Г. ШЕВЧУК, В.В. ЧЕПИКОВА

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

Поступила в редакцию 31 марта 2020

Аннотация. Предложен алгоритм стабилизации видеопоследовательностей на основе нейронной сети, для обучения которой используются только предшествующие кадры. В качестве кодера выступает адаптированная сеть ResNet-50. Показано, что предложенный алгоритм немного уступает алгоритму подпространственной стабилизации, что компенсируются возможностью его использования в режиме реального времени.

Ключевые слова: стабилизация видео, нейронная сеть, подпространственная стабилизация

Введение

Часто трудно просматривать видеоизображение, снятое с помощью плохо- или не закрепленных бытовых видеокамер, из-за наличия дрожания в них. На сегодняшний день существуют различные методы стабилизации цифрового видео для улучшения качества его воспроизведения. Чаще всего такие методы основаны на удалении резких движений камеры и решают эту проблему путем анализа глобального оптического потока, сначала оценивая, а затем выравнивая траекторию камеры, используя автономные вычисления, что затрудняет их применение в режиме реального времени, онлайн. Некоторые методы онлайн-стабилизации следуют процедуре «захват → вычисление → отображение» для каждого входящего видеокадра в режиме реального времени с низкой задержкой. Движение камеры в таких методах оценивается с помощью аффинной трансформации, гомографии или с использованием расчетных сеток, что гарантирует высокую точность для сцен с маленькими смещениями, но допускает серьезные сбои для кадров с большими.

В отличие от существующих подходов, которые должны явно моделировать путь камеры, чтобы сгладить его, предлагается алгоритм стабилизации видео на основе нейронной сети, используемой для вычисления устойчивого преобразования исходя из обработанных кадров (рис. 1).

Рис. 1. Алгоритм стабилизации видео на основе нейронной сети

Архитектура сети

Предложенный алгоритм стабилизирует видео без использования будущих кадров. Таким образом, задача онлайн-стабилизации превращается в задачу обучения с учителем регрессии условной трансформации без явного вычисления пути камеры.

На вход сети подается нестабильный кадр I_t и шесть условных предыдущих стабильных кадров выбранных приблизительно из одной секунды $S_t = I_{t-32}, I_{t-16}, I_{t-8}, I_{t-4}, I_{t-2}, I_{t-1}$ для временной отметки t. Выборка таких кадров более плотная вблизи и разрежена дальше от входящего кадра. Предложенный алгоритм регрессирует трансформацию $f_t^{i,j}$ для (i, j)-го регулярно разделенной решетки потока $g_t^{i,j}$, где решетка $G_t = \left\{ g_t^{i,j} | 1 \le i, j \le 4 \right\}$ размером 4×4 элемента распределена по кадру I_t. Выход модели – это последовательный набор трансформаций $F_t = \left\{ f_t^{i,j} | 1 \le i, j \le 4 \right\}$ для кадра I_t. После чего стабилизированный кадр получается путем:

$$\hat{I}_t = F_t * I_t,$$

где * – это оператор преобразования.

Предложенная модель является сиамской сверточной нейронной сетью, которая состоит из двух ветвей с общими параметрами сети. Сиамская архитектура использована для сохранения временной согласованности успешно стабилизированных кадров $\hat{I}_{t-1} = F_{t-1} * I_{t-1}$ и $\hat{I}_t = F_t * I_t$. Каждая ветвь является двухшаговой сетью, состоящей из кодера, который выделяет высокоуровневые признаки со входов, и регрессора, который предсказывает стабилизирующую трансформацию из извлеченных карт признаков. На рис. 2 показана архитектура используемой нейронной сети.

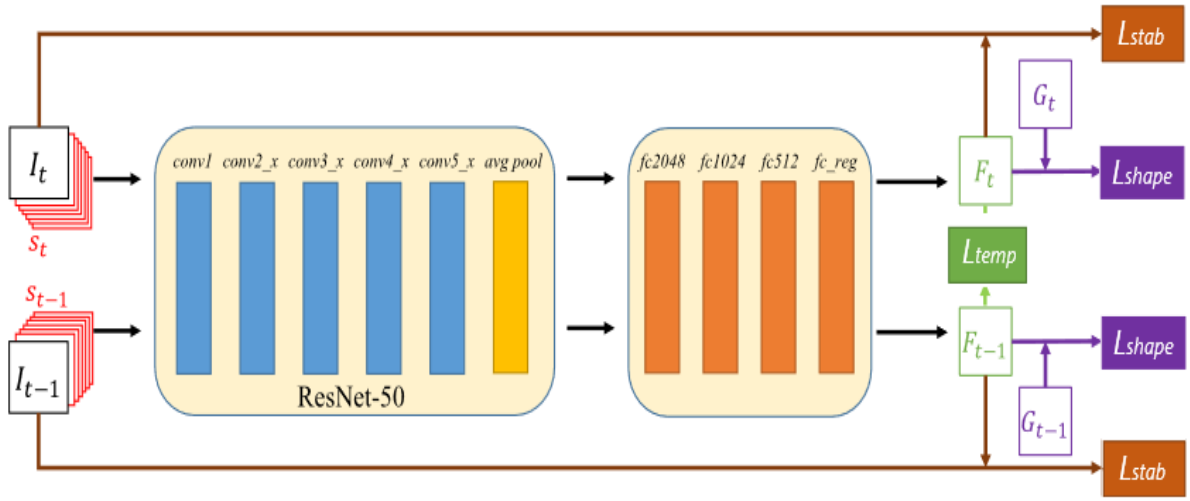

Рис. 2. Архитектура нейронной сети

На рис. 2 на вход поступает семь последовательных кадров в градации серого, каждый размерностью $W \times H \times 1$ пикселей, содержащих шесть последовательных стабильных кадров S_t и одного нестабильного I_t. Кадры передаются кодеру для извлечения признаков. В качестве кодера выступает адаптированная сеть ResNet-50 [1]. Полученные карты признаков имеют размерность $1 \times 1 \times 2048$ пикселей. Далее, используется последовательность полносвязных слоев с выходным пространством признаков $\left\{ 2048, 1024, 512, (h+1) \times (w+1) \times 2 \right\}$, где $w = 4$ и $h = 4$ это размеры решетки по оси x и y соответственно. Выходная размерность сети соответствует общему числу вершин решетки.

Стабилизационные функции потерь

При обучении сети используется три вида функции потерь:

1. Функция потерь стабильности. Переводит деформированные неустойчивые кадры в устойчивые кадры из обучающей выборки, используя выравнивание пикселей и выравнивание характерных точек. Она определяется как:

$$L_{stab}\left(F_t, I_t\right) = \alpha_1 L_{pixel}\left(F_t, I_t\right) + \alpha_2 L_{feature}\left(F_t, I_t\right),$$

где L_{pixel} – это функция выравнивания пикселей, $L_{feature}$ – это функция выравнивания особых точек, F_t – предсказанная трансформация, $\alpha_1 = 50,0$ и $\alpha_2 = 1,0$ константные веса.

Функция выравнивания пикселей L_{pixel} оценивает, как трансформированный кадр $\hat{I}_t = F_t * I_t$ соотносится со стабильным кадром из обучающей выборки I'_t с помощью среднеквадратичной ошибки:

$$L_{pixel}\left(F_t, I_t\right) = \frac{1}{D} I'_t - F_t * I_{t2}^2,$$

где *D*– пространственная размерность кадра.

Функция выравнивания особых точек $L_{feature}$ вычисляется как средняя ошибка выравнивания совпавших особых точек после трансформации нестабильного кадра, используя предсказанную трансформацию F_t:

$$L_{feature}\left(F_t, I_t\right) = \frac{1}{m}\sum_{i=1}^{m} p_t^{ri} - F_t * p_{t2}^{i2},$$

где p_t^{ri}, p_t^i – это *i*-ая совпавшая пара признаков, m – количество пар признаков.

2. Функция потерь сохранения формы состоит из функции потерь трансформации внутри решетки L_{intra} и функции согласованности между решетками L_{inter}:

$$L_{shape}\left(F_t, G_t\right) = \gamma_1 L_{intra}\left(F_t, G_t\right) + \gamma_2 L_{inter}\left(F_t, G_t\right),$$

где $\gamma_1 = 1,0$ и $\gamma_2 = 20,0$ – веса членов функции потерь.

Функции потерь трансформации внутри решетки L_{intra} поощряет треугольник соседних, деформированных вершин $\left\{\hat{v}_t, \hat{v}_t^0, \hat{v}_t^1\right\} \subset f_t * g_t$ следовать трансформации соответствия:

$$L_{intra}\left(F_t, G_t\right) = \frac{1}{N}\sum_{\hat{v}_t} \hat{v}_t - \hat{v}_t^1 - sR\vec{v}_t^{012}{}_2,$$

где $\vec{v}_t^{01} = \hat{v}_t^0 - \hat{v}_t^1$ и $S = \hat{v}_t - \hat{v}_t^1 / \hat{v}_t^0 - \hat{v}_t^1$, $R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ – матрица вращения, N – общее количество треугольных вершин, G_t – расчетная решетка.

Чтобы стимулировать последовательное преобразование соседних решеток, вводится функции согласованности между решетками L_{inter}. Для каждой вершины и ее соседних вершин $\hat{v}_t^0, \hat{v}_t^1$, расположенных вдоль ребра двух исходных соседних решеток, два вектора $\vec{v}_t = \hat{v}_t^1 - \hat{v}_t$ и $\vec{v}_t^0 = \hat{v}_t - \hat{v}_t^0$, образованные деформированными вершинами, должны быть идентичными:

$$L_{inter}\left(F_t, G_t\right) = \frac{1}{M}\sum_{\left\{\hat{v}_t, \hat{v}_t^0, \hat{v}_t^1\right\}} \hat{v}_t^1 - \hat{v}_t - \left(\hat{v}_t - \hat{v}_t^0\right)_2^2,$$

где *M* – общее количество последовательных вершин сетки.

3. Функция временных потерь обеспечивает временную когерентность между соседними кадрами, используя сиамскую сетевую архитектуру. Каждый раз, когда два последовательных примера $\{I_t, S_t\}$ и $\{I_{t-1}, S_{t-1}\}$ подаются на вход сети, два последовательных преобразования F_t и F_{t-1}. Функция временных потерь задана как:

$$L_{temp}\left(F_t, F_{t-1}, I_t, I_{t-1}\right) = \lambda \frac{1}{D} F_t * I_t - w\left(F_{t-1} * I_{t-1}\right)_2^2.$$

Комплексная функция потерь основана на соседних входящих кадрах I_t и I_{t-1} и задана как:

$$L = \sum_{i \in \{t, t-1\}} L_{stab}\left(F_i, I_i\right) + L_{shape}\left(F_i, G_i\right) + L_{temp}\left(F_t, F_{t-1}, I_t, I_{t-1}\right),$$

где L_{stab} – это функция потерь стабильности, L_{shape} – это функция потерь сохранения формы, L_{temp} – это функция временных потерь.

Оценка эффективности предложенного алгоритма стабилизации

Для оценки работы алгоритма были использованы следующие количественные метрики: коэффициент кадрирования, искажение и стабильность.

Коэффициент кадрирования измеряет площадь оставшегося содержимого после стабилизации. Более высокий коэффициент кадрирования с меньшим обрезанием является предпочтительным. Коэффициент кадрирования по кадру есть составляющая глобальной гомографии H_t, рассчитанная по входному и выходному кадру. Затем коэффициенты кадров усредняются для генерации значения коэффициента кадрирования всего видео

$$Cr\left(I_t, \hat{I}_t\right) = \sqrt{H_t\left[0,0\right]^2 * H_t\left[0,1\right]^2}.$$

Значение искажения оценивает степень искажения, вносимого стабилизацией. Для каждого кадра оно вычисляется как отношение двух самых больших собственных значений аффинной части гомографии H_t. Минимальное значение, которое обозначает наихудшее искажение, выбирается как значение искажения для всего видео.

$$Dv_t = \frac{E_{t2}}{E_{t1}},$$

где E_{t1}, E_{t2} - два самых больших собственных значения аффинной части гомографии H_t.

Показатель стабильности оценивает, насколько стабилизировано видео. Для его вычисления используется анализ частотной области траектории камеры. Пространственно распределенные траектории камеры вычисляются как пересечения вершин решетки 4×4 последовательных кадров. Затем пересечения вершин представляются в виде одномерного компонента для всех пересечений, что дает финальный результат.

$$Stab_t = \min\left(G_t \wedge G_{t-1}\right),$$

где G_t, G_{t-1} – расчетные решетки последовательных кадров.

Предложенный алгоритм был сравнен с оффлайн алгоритмом подпространственной стабилизации [2] на тестовом наборе данных. В этот набор данных входят видео, разделенные на следующие категории: стандартное, быстрое вращение, быстрое приближение, большой параллакс, бег и толпа. Результаты сравнения представлены на рис. 3.

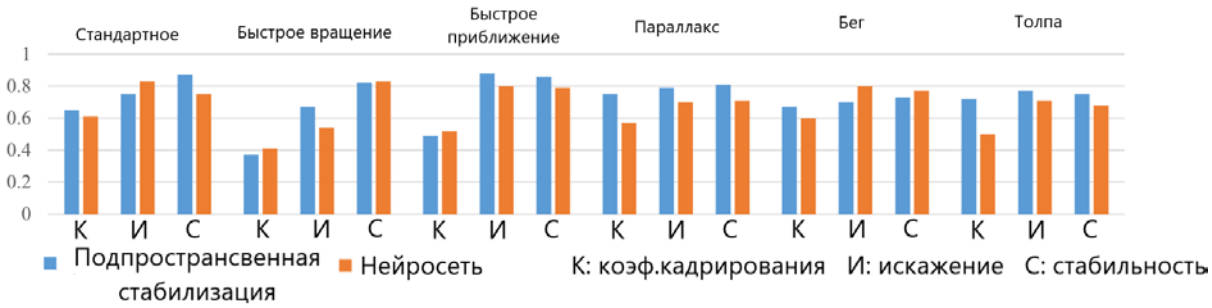

Рис.3. Количественное сравнение алгоритма подпространственной стабилизации с алгоритмом стабилизации на основе нейросетей на тестовом наборе данных

Из рис. 3 видно, что предложенный алгоритм слегка уступает алгоритму подпространственной стабилизации (средние показатели для алгоритма подпространственной стабилизации: коэффициента кадрирования – 0,62, искажение – 0,75, стабильность – 0,8; средние показатели для нейросети: коэффициента кадрирования – 0,54, искажение – 0,745, стабильность – 0,79), но является более устойчивым к резким движениям камеры.

Заключение

Предложена модель нейронной сети для стабилизации видеопоследовательностей. Количественное сравнение с алгоритмом подпространсвенной стабилизации демонстрирует сопоставимые результаты. Нужно заметить, что задача стабилизации в режиме реального времени является значительно более сложной, т. к. для стабилизации используются только кадры без определения глобального пути камеры. Соответственно количественные результаты эффективности стабилизации в реальном времени слегка уступают оффлайн стабилизации, однако это компенсируется работой алгоритма в режимереального времени и большей устойчивостью к резким движениям камеры.

VIDEO STABILIZATION USING NEURAL NETWORKS

A.S. PCHELKIN, A.G. SHAUCHUK, V.V. CHEPIKOVA

Abstract. A video stabilization algorithm based on a neural network is proposed, for the training of which only previous frames are used. The ResNet-50 network has been adapted as a conference code. It is shown that the proposed algorithm is slightly inferior to the subspace video stabilization algorithm, which allows it to be used in real time.

Keywords: video stabilization, neural network, subspace video stabilization

Списокиспользованныхисточников

1. HeK., [et. al.] // Scientific Conference – CVPR. 2016. P. 770 – 778.
2. LiuF., [et. al.] // ACMTransactionsonGraphics, Vol. 30.2011. P. 1 –10.
3. Фурман,Я.А., [и др.] Введениевконтурныйанализ; приложениякобработкеизображенийисигналовМ.: ФИЗМАТЛИТ, 2002.
4. LiuS.,[et. al.] // ECCV. 2016.P. 800–815.