Курсовая работа по теме: Видеосинхронизация в компьютерном зрении

Иван Вячеславович Килимчук

6 мая 2025 г.

Введение

Видеосинхронизация – это ключевая технология в области компьютерного зрения, которая позволяет синхронизировать видеопотоки для анализа и обработки информации. Эта технология находит широкое применение в различных областях. Например, в трансляции спортивных событий.

1 Определение видеосинхронизации

Видеосинхронизация – это процесс согласования временных рядов двух или более видеопотоков для обеспечения их одновременного воспроизведения. Это позволяет точно сопоставлять кадры из разных источников, что важно для дальнейшего анализа данных.

2 Методы видеосинхронизации

Существует несколько методов видеосинхронизации, среди которых:

2.1 Оптическая синхронизация

В этом методе используются оптические маркеры или особенности на изображениях для их синхронизации. Например, можно использовать узнаваемые объекты или точки интереса в кадре.

Рассмотрим статью "Синхронизация камеры с роллинг-шаттером с субмиллисекундной точностью". Обозреваемый в ней метод использует синхронизацию на основе контента и применим к любому количеству камер с роллинг-шаттером, а также при наличии в видео нескольких фотовспышек или других резких изменений освещения. Когда освещение резко меняется во время экспозиции кадра с роллинг-шаттером, край перехода

может быть обнаружен в нескольких камерах и использован в качестве точки синхронизации. Пример отснятых кадров с резким изменением освещения, вызванным одной фотовспышкой, показан на рис. 1.

Рис. 1: Четыре камеры с датчиками роллинг-шаттера снимают сцену, когда срабатывает фотовспышка. Часть рядов изображения интегрирует свет от вспышки. Передний и задний края легко различимы, а на катке еще и хорошо видны. Края служат очень точными точками синхронизации.

Входными данными для алгоритма синхронизации являются временные метки кадров, извлеченные из видеофайлов, и обнаруженные края перехода при резкой смене освещения. Для всех камер c (кроме опорной камеры c_{ref}) мы находим преобразования синхронизации $s^c(f, r) \rightarrow t^{ref}$, которые отображают временную позицию каждой камеры (f,r) на время опорной камеры t^{ref}. Временная позиция определяется парой кадр-строка (f,r). См. рис. 2.

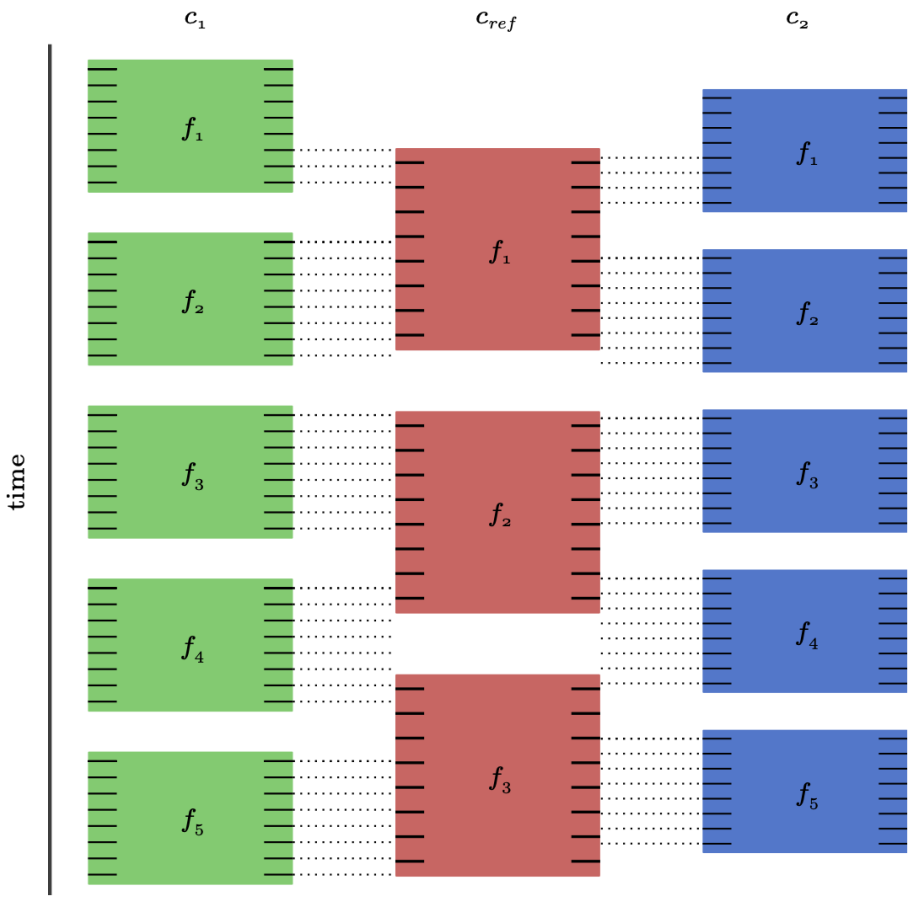

Рис. 2: Синхронизация камер c_1 и c_2 относительно опорной камеры c_{ref}. Временные сдвиги между камерами различны. Короткие черные линии по бокам прямоугольников кадра представляют собой строки изображения. Для каждой камеры c мы находим аффинное преобразование $s^c(f, r) \rightarrow t^{ref}$, которое отображает временную точку, заданную номером кадра f и номером строки r, на время опорной камеры t^{ref}. Пунктирными линиями показано отображение временных моментов захвата строк в c_1 и c_2 на время эталонной камеры.

Резкие изменения освещения легко обнаруживаются. Единственное требование - чтобы большая часть наблюдаемой сцены получала свет от источника. На рис. 3 приведен пример разности интенсивности света, регистрируемого датчиком роллинг-шаттера.

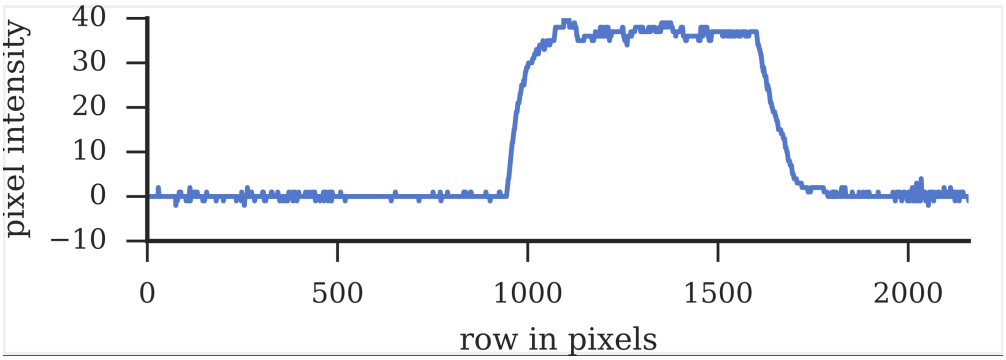

Рис. 3: Разность медианных линий интенсивности между последовательными кадрами в момент вспышки. Ряды в диапазоне 950-1700 были сняты, когда фотографическая вспышка осветила сцену.

Максимум разности медианных линий интенсивности для кадров показывает отчетливые пики. См. рис. 4.

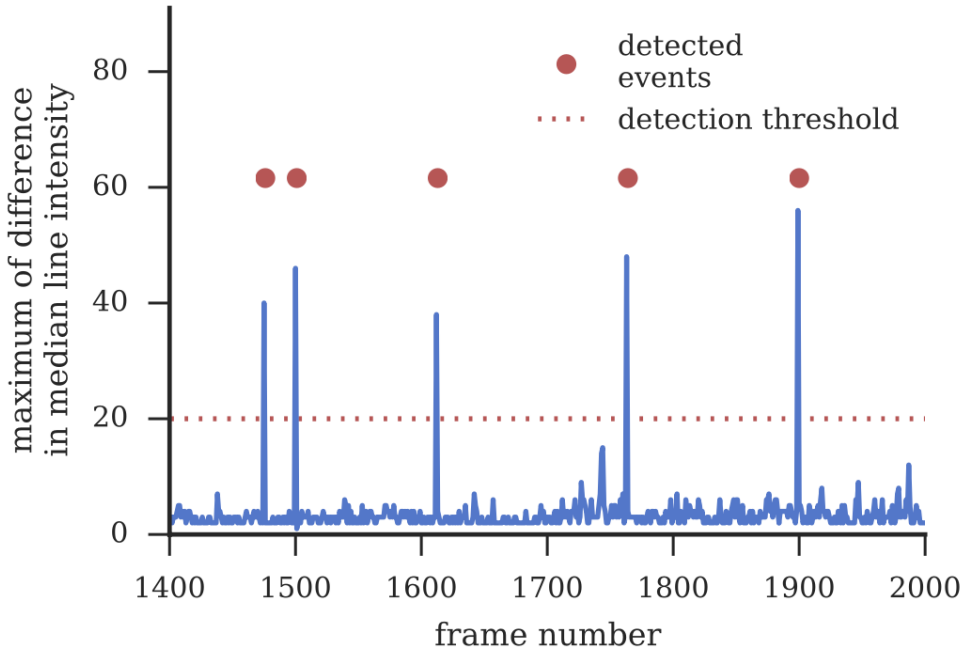

Рис. 4: Обнаружение резких изменений освещения. Для каждого кадра вычисляется медианная линия интенсивности. Затем они вычитаются для последовательных кадров. Максимумы разности для последовательности кадров показаны на графике. Четко видимые пики соответствуют изменениям освещения. Мы выделили пороговые значения и обнаружили события, отмеченные на графике красными точками.

Чтобы получить кадры с событиями синхронизации, мы просто выставляем пороговое значение. Метод кратко изложен на рис. 5.

Algorithm 1 Detection of synchronization events

Input: image sequences
Output: synchronization events
foreach *camera* **do**
  foreach *frame* **do**
    m_f := line median intensity (frame)
    $m_f \in \mathbb{N}^n$, where n is frame height
  end
  foreach *frame* **do** compute difference profiles
    $d_f := m_f - m_{f-1}$
    $d_f \in \mathbb{Z}^n$, where n is frame height
  end
  for f **in** $\{f \mid \max(d_f) > \text{threshold}\}$ **do**
    r := find raising edge row in d_f
    event := (f, r)
  end
end
return *events*

Рис. 5:

Доминирующей частью преобразования $s^c(f, r) \to t^{ref}$ является временной сдвиг между камерами c и c_{ref}. Мы компенсируем дрейф часов с помощью линейной составляющей преобразования. Предлагаемое преобразование имеет вид:

$$s(f, r; \alpha, \beta) = \alpha t_f + \beta + r \cdot \frac{T_{\text{frame}}}{R}$$

где α - компенсация дрейфа часов камеры, β - временной сдвиг, f - номер кадра, r - номер строки, t_f - временная метка кадра, R - общее количество строк датчика, а T_{frame} - длительность кадра.

Цель синхронизации - найти $s^c(f, r; \alpha^c, \beta^c)$ для всех камер $C = \{c_1, c_2, ...\}$.

Для эталонной камеры единое глобальное время для кадра f и строки r вычисляется по формуле:

$$t(f,r) = t_f + \frac{R_0 + r}{R} \cdot T_{\text{frame}}$$

где $R = R_1 + R_h + R_0$, а R_1, R_h, R_0 - количество строк, указанное на рис. 6. Константы R_0 и R_1 можно найти в техническом паспорте датчика изображения.

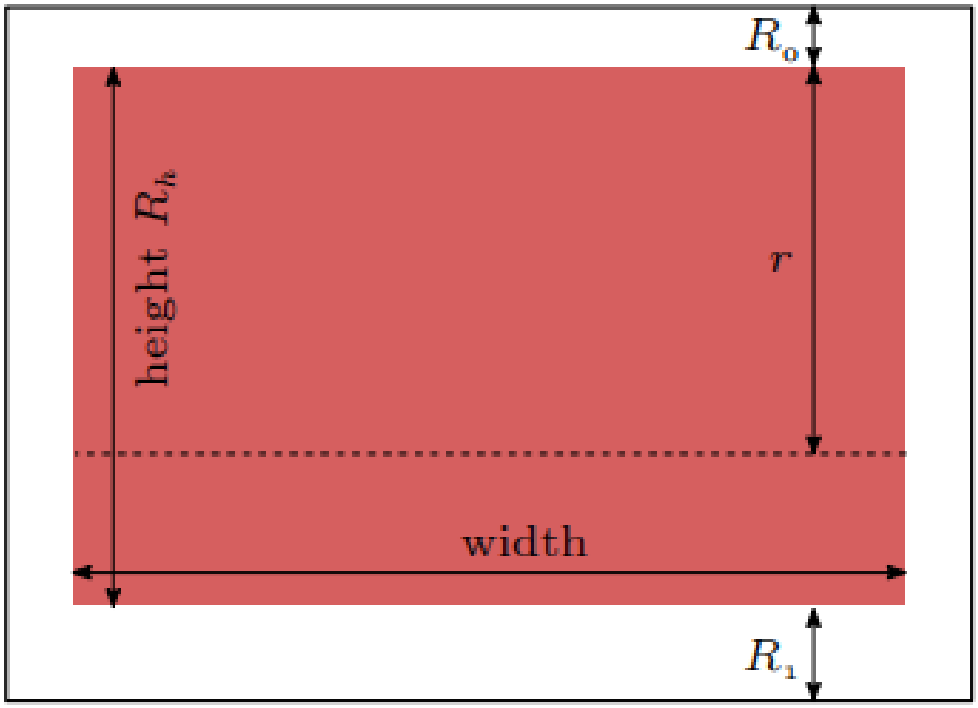

Рис. 6:

Для события, наблюдаемого в камере c и c_{ref} в $(f^c,\, r^c)$ и $(f^{c_{ref}},\, r^{c_{ref}})$, время синхронизированной камеры и время эталонной камеры должны быть равны:

$$s^c(f^c, r^c; \alpha^c, \beta^c) = t^{ref}(f^{c_{ref}}, r^{c_{ref}})(*)$$

На следующем этапе мы вручную выравниваем время в камерах c и c_{ref} вплоть до целых кадров, например, для первого события синхронизации, и автоматически сопоставляем остальные события, чтобы получить:

$$E^{c,c_{\text{ref}}} = \left\{\left((f_1^c, r_1^c), (f_1^{c_{\text{ref}}}, r_1^{c_{\text{ref}}})\right), \ldots, \left((f_k^c, r_k^c), (f_k^{c_{\text{ref}}}, r_k^{c_{\text{ref}}})\right)\right\}.$$

Теперь мы можем построить систему из уравнений (*) для k пар событий синхронизаций $E^{c,c_{ref}}$. Решение методом наименьших квадратов дает неизвестные α и β.

2.2 Анализ движения

Этот метод предполагает использование алгоритмов анализа движения для сопоставления видеопотоков. Такие алгоритмы могут определять общие паттерны движения и синхронизировать видео на их основе.

Рассмотрим статью "PoseSync: Надежная синхронизация видео на основе позы". PoseSync, который синхронизирует два видео, состоит из трех этапов: обрезка видеокадра, определение позы и синхронизация видео (использование DTW). На первом этапе обрезается область, где на изображении присутствует человек, после чего происходит определение позы на обрезанном изображении, затем применяется динамическая временная деформация (DTW) для измерения угла/расстояния между ключевыми точками позы.

Сначала PoseSync обрезает видеокадры с помощью YOLO v5[3]. Операция обрезки исходных кадров повышает точность определения позы, поскольку позволяет избавиться от других людей на заднем плане и любой другой ложной информации. Эти обрезанные кадры передаются в модель определения позы MoveNet, которая возвращает ключевые точки позы для каждого кадра. Наконец, динамическое искажение времени (DTW)[4] используется для сопоставления ключевых точек для обоих видео (с помощью метрик, основанных на расстоянии или угле, описанных ниже). Чтобы решить проблему разницы в размерах между двумя позами, авторы предлагают метрику Angle-Mean Absolute Error, которая вычисляет MAE (средняя абсолютная ошибка) между углами ключевых суставов скелета. Эта метрика инвариантна к масштабу и положению позы.

Movenet - это архитектура глубокого обучения, специально созданная для точного обнаружения и отслеживания человеческих поз в реальном времени из видеопотоков. Это модель использует тепловые карты для точного определения местоположения ключевых точек на теле человека. Подробнее см. в статье [5]. Стоит отметить, что MoveNet ограничен в определении позы одного человека на изображении, поэтому авторы используют YOLO v5 для обрезки видеокадров.

Динамическое искажение времени (DTW) - это метод, используемый для расчета сходства между двумя временными рядами. Основная цель DTW - выявить соответствующие совпадающие элементы во временных рядах и измерить расстояние между ними. Авторы используем DTW для синхронизации последовательностей поз человека, найденных из двух видео. Элементы последовательностей представляют собой множества ключевых точек.

Совмещение поз может быть выполнено, используя ключевые точки позы (координаты суставов человеческого тела: x,y) и среднюю абсолютную ошибку (MAE) в качестве метрики расстояния. Ограничением этой метрики является то, что она не являются инвариантной к масштабу, повороту и сдвигу. То есть даже если позы похожи, то MAE

между ключевыми точками может быть высокой из-за разницы в масштабе или положении. Чтобы преодолеть эту проблему, мы используем метод средней абсолютной ошибки на основе угла. Для этого сначала вычисляются углы, образованные тремя точками суставов, один из которых является шарниром. Затем вычисляется MAE между углами как метрика расстояния. Для расчета углов мы используем 8 суставов: левый плечевой сустав, правый плечевой сустав, правый локтевой сустав, левый локтевой сустав, правый тазобедренный сустав, левый тазобедренный сустав, правый коленный сустав, левый коленный сустав. Авторы используют угловую MAE в качестве cost function для динамического искажения времени (DTW). В зависимости от задачи можно придать вес различным ключевым точкам (суставам), и это может помочь в лучшей синхронизации видео.

На рисунках 7 и 8 показано выравнивание двух видео с различными типами человеческих движений. Первый столбец состоит из ключевых кадров эталонного видео, а второй столбец содержит тестовые видеокадры с тем же индексом, что и эталонные кадры. Третий столбец состоит из тестовых кадров, сопоставленных с эталонными с помощью PoseSync. Тестовые видео создавались путем увеличения/уменьшения скорости всего эталонного видео или его начала/середины/конца.

Рис. 7: Синхронизация танцевальных видео: ключевые кадры эталонного видео (столбец 1), соответствующие кадры тестового видео (колонка 2) и кадры тестового видео, сопоставленные с соответствующими эталонными кадрами с помощью DTW (столбец 3)

Рис. 8: Синхронизация теннисных кадров: ключевые кадры эталонного видео (колонка 1), соответствующие тестовые видеокадры (столбец 2) и тестовые видеокадры, сопоставленные с соответствующими эталонными кадрами с помощью DTW (столбец 3)

Заключение

В данной курсовой работе была рассмотрена проблема видеосинхронизации в области компьютерного зрения, а также два основных метода, применяемых для её решения. Мы изучили оптическую синхронизацию и анализ движения. Каждая из этих методик позволяет достичь высокой точности синхронизации видеопотоков, что особенно важно при анализе данных с нескольких камер, а также имеет свои преимущества и ограничения в зависимости от конкретного применения и условий съемки.

Список литературы

1. Matej Smid and Jiri Matas. (2019). *Rolling Shutter Camera Synchronization with Sub-millisecond Accuracy*. Czech Technical University in Prague. `https://github.com/smidm/flashvideosynchronization?tab=readme-ov-file`
2. Rishit Javia, Falak Shah, and Shivam Dave. (2023). *PoseSync: Robust pose based video synchronization*. Infocusp Innovations LLP. `https://github.com/infocusp/posesync`
3. Jocher, G.: Yolov5 by ultralytics (2020). `https://github.com/ultralytics/yolov5`
4. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD workshop. vol. 10, pp. 359–370. Seattle, WA, USA: (1994)
5. Chen, Y.H., Oerlemans, A., Belletti, F., Bunner, A., Sundaram, V.: MoveNet: Next-generation pose detection model (2021), `https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html`