Обработка видеофайлов: объединение, обрезка и синхронизация

1. Объединение

Для видеозаписи были использованы три индивидуальные промышленные камеры JAI GO-5000M (частота 100 к/с, разрешение 1392x1000), записывающие крупные планом каждого из трёх коммуникантов, а также камера общего плана GoPro (частота 50 к/с, разрешение 2700x1500). При помощи специально написанной программы видеопоток (raw) с промышленных камер преобразовывался в формат mjpeg (контейнер avi) с автоматическим делением на файлы размером в 2 Гб. Камера общего видео записывала видео в формате mp4 с делением на файлы размером в 4 Гб.

На первом этапе для каждой записи фрагменты каждого из четырёх видео были объединены, а видео с камеры общего плана ещё дополнительно конвертировано в формат mjpeg (что было принципиально важно для дальнейшей работы, предполагающей аннотацию видео), дав в результате четыре файла:

N – индивидуальное видео с камеры, снимавшей Рассказчика (Narrator),

C – индивидуальное видео с камеры, снимавшей Комментатора (Commentator),

R – индивидуальное видео с камеры, снимавшей Пересказчика (Reteller),

W – видео с камеры общего плана.

Для объединения всех видео были написаны 2 специальные программы – AviRecover (исправляет заголовки в avi файлах) и FramesDropper (меняет частоту видеозаписи и скорость воспроизведения видео). Кроме того, для облегчения общего процесса обработки были написаны 2 пакетных .bat файла, позволившие осуществить весь процесс «одной кнопкой», что существенно сэкономило временные затраты. Внутренняя процедура, описанная в каждом из .bat-файлов, включала в себя следующие шаги:

1. Для индивидуального видео:
   1.1. Исходные фрагменты по 2 Гб пропускались через программу AviRecover.
   1.2. Получившиеся новые фрагменты пропускались через программу FramesDropper.
   1.3. Выходные фрагменты склеивались программой ffmpeg.

2. Для общего видео: исходные фрагменты по 4 Гб пропускались через программу ffmpeg, где за один запуск происходило их склеивание и одновременное изменение разрешения до 1352x860 и формата с mp4 на mjpeg.

Таким образом, после этого этапа все четыре видеофайла имели формат mjpeg; три индивидуальных видеофайла имели частоту 100 к/с и разрешение 1392x1000, а видео с камеры общего плана – частоту 50 к/с и разрешение 1352x860.

2. Обрезка

После объединения всех фрагментов видео в единые файлы для каждой из записей осуществлялась первичная синхронизация всех видеофайлов между собой. Эта процедура включала в себя обрезку начал и концов всех видео. Полученные в результате этой процедуры файлы уже начинались с одного и того же момента, с точностью до сотых долей секунды (отправной точкой служил запуск секундомера iPad'a, который по очереди показывался во все видеокамеры), и заканчивались также на идентичном для всех камер моменте.

Данная процедура применялась как для каждого индивидуального видео, так и для двух видеофайлов с айтрекеров (частота записи движений глаз 50 Hz, частота видеозаписи 25 к/с, исходное видеоразрешение 1920x1080) и видео с камеры общего плана. Перед этим глазные файлы были сконвертированы в формат .mjpeg с помощью программы ffmpeg, что позволяло осуществлять дальнейшие операции с точностью до кадра.

Последующая синхронизованная обрезка начал для каждого из 6 видеофайлов каждой записи осуществлялась следующим образом:

1. Видео файл открывался в программе QuickTime Player или программе ELAN
2. При прокрутке видео находился момент показа секундомера iPad'a.
3. Из текущего времени записи вычиталось время показаний секундромера iPad'a. (Полученный тайм-код соответствовал моменту запуска секундомера iPad'a и был точкой синхронизации начал видеофайлов)
4. Файл видео обрезался по полученной точке синхронизации в программе ffmpeg.

Затем для каждого синхронизованного таким образом видеофайла в программе ffmpeg осуществлялась обрезка концов. Для этого осуществлялись следующие шаги:

1. В программе ELAN одновременно открывались 2 файла – видео с камеры общего плана и последовательно каждое из индивидуальных видео N, C и R.
2. В режиме Synchronization Mode максимально близко к концу записи в обоих файлах находилась т. н. «опорная точка», представляющая собой заметный жест, мигание или какое-либо другое ярко видимое на глаз явление.
3. Тайм-код «опорной точки» для каждого из видео заносился в программу ffmpeg, которая затем осуществляла по ней обрезку концов всех видеофайлов.

После этого этапа работы каждая из записей содержала 6 синхронизованных по началу и концу видеофайлов – 3 индивидуальных видео N, C и R, 2 видео с айтрекеров N и R, а также видео с камеры общего плана.

Тем не менее даже синхронизованные таким образом, индивидуальные файлы всегда оказывались короче остальных (разница составляла от 0.4 до 3 с). Эта особенность объяснялась потерей кадров, которая была связана с перегревом индивидуальных видеокамер, в результате чего они периодически записывали видео с меньшей частотой. Так, при одновременной прокрутке глазного (или общего) видео и синхронизованного с ним по началу и концу индивидуального видео внутри всегда обнаруживалась видимая на глаз рассинхронизация, которая нарастала к концу записи. Начиная с некоторого момента, видео с айтрекера и камеры общего плана всё сильнее отставали от индивидуального видеофайла, что отражалось в более поздних началах жестов и соответственно в сильной внутренней рассинхронизации в движениях одного и того же участника. В целом в индивидуальных видеофайлах пропало от 40 до 300 кадров, что потребовало большой дополнительной работы по восстановлению потерь и синхронизации. Эта работа была осуществлена на 3 этапе.

3. Детальная синхронизация (восстановление потерянных кадров)

Необходимо отметить, что несмотря на кажущуюся незначительность потерь (1-2 секунды на 20-30-минутное видео), они оказались критичными для дальнейшей работы проекта, т.к. в разметке корпуса отмечаются мельчайшие и очень непродолжительные по времени явления естественной коммуникации. Пренебрежение потерей кадров естественным образом приводит, таким образом, не только к видимой на глаз рассинхронизации видеозаписей, но к полной рассогласованности аннотаций.

Покадровая синхронизация и восстановление применялись практически для каждого из индивидуальных видео N, C, R в каждой из 24 записей. Однако прежде всего в программе ELAN оценивалась длительность индивидуального файла R (N/C) по сравнению с соответствующим ему видео с айтрекеров N (R). Эта процедура фактически соответствовала приблизительной оценке количества потерянных кадров: если глазной файл в среднем превышал индивидуальное видео более, чем на 5 с (соответственно число потерянных кадров превышало 20 кадров в минуту), то такая запись временно откладывалась.

Для точной оценки потерь видео с камеры общего плана оказалось неподходящим, так как 1) ракурс съемки с камеры общего плана и с индивидуальных камер не совпадает, 2) общий план с камеры общего плана не позволяет различать микродвижения (например, перебирания пальцами, моргания), которые помогли бы выявить рассинхронизацию ровно в тех моментах, когда она возникала. Поэтому для оценки потерь и процедуры восстановления видеозаписей было решено использовать видео с айтрекеров. Следует отметить, что данный метод синхронизации с использованием какого-либо «опорного» видео не является новым и ранее применялся, в частности, в бельгийском мультимодальном ресурсе *InSight Interaction corpus*, также содержащем некоторое количество видео- и айтрекерных данных от двух участников диалога (всего 15 диалогов по 20 минут каждый). В этом корпусе в силу особенностей записи потеря кадров была характерна не для индивидуальных, а для айтрекерных видео, и фиксированные данные индивидуальных видеокамер служили опорными для дальнейшего восстановления потерянных кадров с помощью т.н. «растягивания» (“stretch” [Brône, Oben 2015: 204]).

В случаях, когда запись подлежала восстановлению (потеряно не больше 20 кадров в минуту) синхронизация осуществлялась с помощью незаметного на глаз добавления кадров в статичные места (когда участник эксперимента не двигался), максимально приближенные к месту видимой рассинхронизации. Добавленные кадры представляли собой копии соседних относительно статичных кадров. Данная процедура осуществлялась с помощью предварительно написанного на языке Python скрипта, которому на вход подавались таймкоды для дублирования кадров и сам индивидуальный файл. На текущем этапе работы восстановление кадров стало одной из самых трудоёмких процедур, во многом из-за ручного поиска статичных мест и соответствующих кадров для дублирования. Однако параллельно с этим для будущих исследований была создана и протестирована программа, которая в режиме реального времени (еще при конвертации первичного видеопотока в mjpeg) находит потерянные кадры, автоматически дублирует потери, а также дает статистику, где и сколько кадров было вставлено. Предполагается, что в будущем данная программа сможет свести на нет возможные проблемы рассинхронизации видеофайлов.

В редких случаях в текущих записях индивидуальное видео не опережало, а, наоборот, отставало от глазного файла. По-видимому, индивидуальные камеры при наличии каких-то факторов могли записывать видео с частотой, как меньшей, так и большей 100 к/с. Точные причины этого пока не установлены и остаются предметом дальнейших исследований. Для самих же файлов в этих случаях проходила процедура изъятия лишних кадров, по своей сути аналогичная процедуре вставки. Для этой

процедуры был написан аналогичный скрипт на Python, который не дублировал кадры, а наоборот удалял «лишние» по заданному тайм-коду.

Сам процесс вставки кадров проходил следующим образом:

1. В программе ELAN одновременно открывались индивидуальный видеофайл R (N/C) и соответствующий ему глазной видеофайл N (R). Глазной файл устанавливался как главное видео (master media).
2. Оба файла покадрово прокручивались от начала к концу.
3. В случае разной длительности и постоянном опережении глазного видео со стороны индивидуального последнее дополнительно обрезалось по глазному файлу, а сам временной сдвиг (какой файл опережает другой на сколько кадров) заносился в параллельно созданный текстовый файл с комментариями.
4. В остальных случаях оба видео прокручивались в режиме Synchronization mode до первого заметного моргания/ движения рук/ корпуса, которые были видны на обоих видео[1]. Затем в этих местах проверялась синхронизация. В случае, если всё совпадало, осуществлялась дальнейшая прокрутка до следующей подобной «опорной» точки.
5. Если между файлами обнаруживалась рассинхронизация, первым делом на глаз определялось число кадров, на которое глазное видео отставало от индивидуального. При этом учитывалось, что частота глазного видео 25 к/с, тогда как индивидуального файла – 100 к/с, и следовательно, 1 кадр глазного видео соответствует 4 кадрам индивидуального. Таким образом, если была видна рассинхронизация на 1 глазной кадр, в индивидуальное видео необходимо было добавить 4 кадра.
6. Если рассинхронизация составляла более 2 глазных кадров (соответственно, в индивидуальном видео на текущем отрезке было потеряно более 8 кадров), осуществлялся более детальный предварительный просмотр предыдущей области, с целью детекции более слабой предварительной рассинхронизации на 4 кадра. Это было возможно практически всегда, поскольку во всех записях, кроме одной, рассинхронизация нарастала постепенно, и можно было детально отследить этот процесс с максимальной точностью – до 3-4 кадров на каждом временном промежутке.
7. При детекции первичной рассинхронизации на 1 глазной кадр (минимально допустимое значение в программе ELAN) осуществлялась прокрутка обоих видео назад, до максимально близкой «опорной» точки, где рассинхронизации ещё не было. Параллельно в текстовом файле фиксировались таймкоды точки впервые замеченной рассинхронизации и последней «опорной» точки, где всё ещё происходит синхронно.
8. В получившемся временном интервале от последней «синхронной» точки до первой «рассинхронной» на 4 кадра в индивидуальном видео фиксировались максимально статичные места и их таймкоды, в которые затем с помощью скрипта вставлялось 4 пропущенных кадра. При этом кадры для вставления выбирались не подряд, а по возможности максимально распределённо, чтобы дополнительно не «поехали» синхронные отрезки.
9. Параллельно в программе ELAN в точке рассинхронизации глазной файл виртуально сдвигался на 40 мс, файлы становились таким образом виртуально синхронизованными, и прокручивались аналогичным образом дальше, с повторением той же самой процедуры.

[1] Метод использования подобных жестов в качестве «опорных» точек синхронизации так же применялся в ходе создания бельгийского корпуса *InSight Interaction corpus*: “the onset or offset of hand gestures were particularly frequently used as anchor points because those actions were clear signals in each of the video files” [Brône, Oben 2015: 204]

10. В конце процедуры в программе ELAN открывались 3 файла – глазное видео, восстановленный индивидуальный файл и видео с камеры общего плана, с которым так же осуществлялась проверка синхронизации.

На выходе после восстановления кадров и дальнейшей проверки каждая из обрабатываемых записей содержала 3 полностью синхронизированных на глаз индивидуальных видеофайла N, C и R, в которых было ровно в 2 раза больше кадров, чем в видеозаписи с камеры общего плана и в 4 раза больше, чем в каждом из глазных файлов N и R. Все 6 файлов имели одинаковую длительность с точностью до сотых секунды.

Так, для одной видеозаписи файлы выглядели следующим образом:

Восстановлено кадров	**Файл**
100	Pears04N-vi-mute-recover-2ndcut.avi
64	Pears04C-vi-mute-recover-2ndcut.avi
300	Pears04R-vi-mute-recover-2ndcut.avi
-	Pears04R-ey-2ndcut.avi
-	Pears04N-ey-2ndcut.avi
-	Pears04W-vi-orig-2ndcut.avi

Литература:

1. Brône G., Oben B. (2015), InSight Interaction. A multimodal and multifocal dialogue corpus, Language Resources and Evaluation, 49(1), pp. 195–214