

Колесса А.Е., Лагуткин В.Н., Лукьянов А.П., Слынько Ю.В.

Комплекс алгоритмов оценки и компенсации геометрических линейных искажений входной информации, полученной при неизвестных динамических изменениях системы координат наблюдателя.

Аннотация

Предлагается для рассмотрения несколько алгоритмов компенсации линейных искажений видео изображений, главной частью которых является определение параметров этих искажений. Эта задача решается простыми корреляционными методами или их модификациями. При этом алгоритмы должны удовлетворять следующим условиям: во-первых, они должны устойчиво работать с любыми видеопоследовательностями, в том числе и последовательностями очень плохого качества, во-вторых, алгоритмы должны работать в реальном времени на современных вычислительных средствах.

1. Введение

При съёмке видео изображений зачастую имеет место вибрация камеры, приводящая к нежелательному дрожанию получаемых изображений. В простейшем случае это небольшие колебания оптической оси камеры. В принципе, возможны существенные колебания, достигающие полукадра, особенно при съёмках с сильным увеличением. Кроме того, возможны и более сложные случаи, например, камера может не только колебаться, но и изменять ориентацию и положение в пространстве.

Тогда встаёт задача компенсации таких колебаний камеры и получения стабилизированных изображений. Основной частью задачи является определение параметров деформации изображения от кадра к кадру. Зная параметры деформации, можно провести обратное преобразование входного кадра к прежней системе координат (компенсацию деформации), и отображение полученного кадра на монитор стандартными средствами с использованием специализированной аппаратуры. В частности, на IBM совместимых компьютерах решить данную задачу позволяет графический процессор и библиотеки функций Direct 3D или OpenGL.

В данной статье особое внимание уделено алгоритмам определения параметров деформации изображений от кадра к кадру. Здесь мы ограничиваемся классом линейных (аффинных) деформаций. Несмотря на относительную простоту модели деформации, она позволяет описать такие типичные искажения, как смещение, поворот, сжатие, перекося и их комбинации.

Основной целью при разработке алгоритмов было обеспечение следующих двух качеств.

Во-первых, алгоритмы должны работать устойчиво с самыми разными видеопоследовательностями. Причем эти последовательности могут быть сняты в сложных условиях и иметь очень плохое качество.

Во-вторых, данные алгоритмы должны обеспечивать обработку видеопотоков в реальном времени на современных вычислительных средствах.

2. Постановка задачи

В данной работе рассматривается следующая задача.

Пусть даны два кадра, один из которых является опорным, а другой является геометрической трансформацией опорного. Под геометрической трансформацией подразумевается аффинное преобразование, т.е. преобразование вида:

$$\vec{r}' = A\vec{r} + \Delta\vec{r}$$

Необходимо найти параметры этой трансформации.

При этом в процессе создания и отработки алгоритмов на реальных изображениях были учтены возможные отклонения от данной математической модели:

1. Изображение мало контрастно с малым отношением сигнал-шум.
2. Шумы негауссовы, коррелированные по времени и пространству и нестационарные.
3. Дополнительные искажения изображения. Например – смазы, нелинейные геометрические искажения, нелинейные изменения яркости и цветности по кадру, искажения в оптике камеры и т.д.
4. Искажение изображения за счет чересстрочной развёртки.
5. Нестационарность самой снимаемой картины, т.е. движение одних её частей относительно других.

3. Описание алгоритмов

3.1 Общие положения

Для определения преобразования одного кадра относительно другого используется следующая принципиальная формула:

$$Q = \arg \min_Q X(\|F(t-1)_{ij}\|, \|T_Q(F(t))_{ij}\|), \text{ где} \quad (1)$$

Q – вектор параметров преобразования, который может включать как сдвиг кадров, так и угол поворота и коэффициент масштабирования, или 6 аффинных коэффициентов,

X – функция «непохожести», которая должна быть тем меньше, чем сильнее совпадают её первый и второй аргументы,

$F(t)_{ij}$ – кадр момента времени t ,

T_Q – преобразование кадра с заданным вектором параметров преобразования.

Существуют различные методы оценки параметров деформации кадра, в частности, на основе оптического потока внутри кадра [5], на основе выделения характерных точек изображения [3],[4] или напрямую используя яркости точек входного изображения [2].

В данной работе под функцией $X(F_1, F_2)$ подразумевается функция среднеквадратичной невязки или какие-либо её обобщения. Данная функция достаточно быстро вычисляется. При этом вычисления легко распараллеливаемы на современных векторных процессорах.

Обычно функция $T_Q(F)$ может быть представлена как комбинация функций целочисленного сдвига. Тогда она может быть выполнена всего в несколько операций на один кадр (число операций не зависит от размера кадра) путём изменения формального адреса начала кадра в памяти, т.е., фактически, её вычисление занимает пренебрежимо малое время по сравнению с остальными вычислениями.

3.2 Простейший метод корреляции

В простейшем случае функция $X(F_1, F_2)$ может быть представлена в виде:

$$X(F(t-1), T_Q(F(t))) = \frac{1}{S(M)} \sum_{(i,j) \in M(dx, dy)} (F(t-1)_{ij} - F(t)_{i+dx, j+dy})^2, \quad (2)$$

а само преобразование

$$Q = (dx, dy) = \arg \min_{(dx, dy) \in D} X(F(t-1), T_Q(F(t))), \quad (3)$$

где $Q=(dx, dy)$ – сдвиги вдоль обеих осей,

D - область возможных сдвигов, по которой производится поиск,

$M(dx, dy)$ – область, по которой производится вычисление невязки между кадрами

$S(M)$ – площадь области M (число точек).

В простейшем случае можно под областью M понимать область, для которой точки (i,j) и $(i+dx,j+dy)$ принадлежат $F(t-1)$, т.е. производить вычисление среднеквадратичной невязки по области пересечения кадров.

Очень существенным моментом в данном случае является многомодальность функции (на Рис1 хорошо видны многочисленные локальные минимумы) $X(F_1, F_2)$, и для нахождения сдвига нельзя напрямую использовать методы спуска. Тут необходим перебор всех возможных вариантов. Это существенно сказывается на трудоемкости данного алгоритма и, если априорная область возможных сдвигов велика, скорость его работы становится недопустимо низкой.

Другим недостатком метода перебора является сложность обобщения данного подхода на случай определения поворота кадра. Ведь оператор поворота кадра, в отличие от оператора целочисленного сдвига, весьма трудоемок, и его использование замедлит и без того медленный алгоритм еще на порядки.

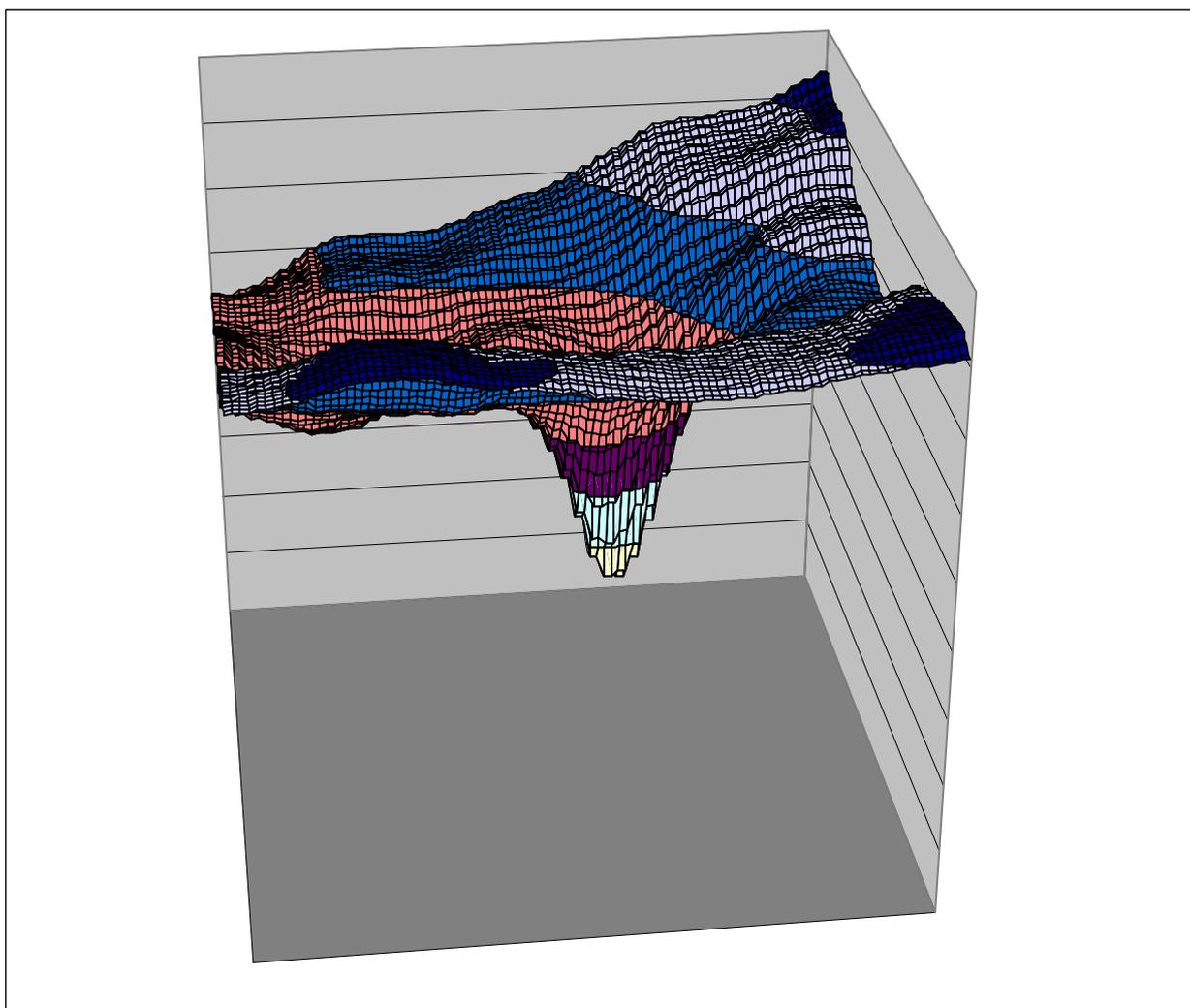


Рис 1.

3.3 Метод выделения корреляционных окон

Для уменьшения количества вычислений можно использовать не весь кадр, а только некоторую его часть. Однако произвольно ее выбирать нельзя, т.к. зачастую в кадрах попадаются неинформативные участки, ошибка определения сдвига по которым даже полным перебором может достигать размеров самого сдвига.

Удобно выбирать некоторое количество прямоугольных областей с наибольшими информативностями, и только по ним определять невязку. Т.е. под областью M понимать объединение нескольких прямоугольников (корреляционных окон). Это позволяет существенно снизить количество выполняемых операций и ускорить алгоритм.

Также это очень удобный способ определения угла поворота и коэффициента масштабирования кадра. Для этого надо подсчитать не сразу сдвиг всего кадра, а сдвиг отдельных окон. Если приписать точке, находящейся в центре окна, сдвиг этого окна, то методом наименьших квадратов нетрудно, зная сдвиг нескольких точек, определить сдвиг, угол поворота и коэффициент масштабирования всего кадра. Т.е. производить вычисление параметров преобразования по следующей формуле:

$$Q = \arg \min_Q \sum_{k=1}^N (\bar{\rho}_k - \tau_Q(\bar{r}_k))^2, \text{ где} \quad (4)$$

\bar{r}_k – координаты центров областей в кадре $F(t-1)$,

$\bar{\rho}_k$ - координаты тех же областей в кадре $F(t)$ (вычисленные по формулам (2), (3))

N – количество прямоугольников,

$\tau_Q(\bar{r}_k)$ - функция преобразования координат. Если необходимо определить только сдвиг, то эта функция принимает вид:

$$\tau_Q(\bar{r}_k) = \bar{r}_k + (dx, dy). \quad (5a)$$

Если же надо определить также коэффициент масштабирования и поворота, то можно записать её в таком виде:

$$\tau_Q(\bar{r}_k) = \begin{pmatrix} a & b \\ -b & a \end{pmatrix} \cdot \bar{r}_k + \begin{pmatrix} dx \\ dy \end{pmatrix}, \text{ тогда} \quad (5b)$$

угол поворота $\varphi = \arctg(b/a)$, а коэффициент масштабирования $\kappa = a^2 + b^2$.

Такая запись в обоих случаях позволяет свести задачу нахождения минимума к поиску решения системы линейных уравнений с 2-го и 4-го порядка соответственно. Возможно обобщить данный метод на случай 6 аффинных коэффициентов (это приведёт к системе 6 линейных уравнений), однако практическая ценность такого обобщения невелика, т.к. редко кадры деформируются по всем 6 аффинным параметрам.

При этом поиск минимума не обязательно проводить по всей области. Можно начать двигаться от центра области D по «спирали», т.е. по квадратам, постепенно увеличивая их сторону. При нахождении минимума целесообразно обойти еще несколько более крупных квадратов. Если в них не окажется точки с меньшим значением невязки, поиск можно остановить. Тем самым можно добиться уменьшения количества операций в случаях малых сдвигов, оставляя при этом возможность определять и большие сдвиги. Такой подход может однако привести в ошибкам в следствии выбора не глобального, а локального минимума.

Корреляционные окна выбираются из следующих соображений. Во-первых, они должны обладать максимальной информативностью. Во-вторых, они должны по возможности отстоять как можно дальше друг от друга. Т.е. критерием выбора является следующий:

$$\vec{r}_k = \arg \max_{\vec{r}_k} \left(I(\vec{r}_k) \cdot \left(\min_{i=1..k-1} |\vec{r}_k - \vec{r}_i| + C \right) \right), \text{ где} \quad (6)$$

$I(\vec{r}_k)$ - информативность точки \vec{r}_k ,

C – некоторая константа, подбираемая экспериментально (можно взять порядка половины размера кадра).

Информативность окна вычисляется как минимум автокорреляционной функции вдоль нескольких направлений, т.е. по формуле:

$$I(\vec{r}_k) = \min_{l=1..L} \frac{1}{\sqrt{dx^2(l) + dy^2(l)}} \sum_{(i,j) \in W(\vec{r}_k)} (F_{ij} - F_{i+dx(l), j+dy(l)})^2, \text{ где} \quad (7)$$

$(dx(l), dy(l))$ – набор направлений,

$W(\vec{r}_k)$ - корреляционное окно,

L – количество направлений.

В данной работе использовались следующие параметры:

$L=4$,

$(dx(1), dy(1))=(0, 1)$,

$(dx(2), dy(2))=(1, 1)$,

$(dx(3), dy(3))=(1, 0)$,

$(dx(4), dy(4))=(1, -1)$.

Выбор автокорреляционной функции в качестве критерия информативности обоснован тем, что и в качестве целевой функции используется функция невязки, а следовательно, ее крутизна пропорциональна точности оценки сдвига.

Здесь необходимо заметить, что после того, как был вычислен сдвиг первого окна, за область D удобно принять окрестность вычисленного сдвига. Тогда можно существенно уменьшить её площадь по сравнению с априорной, т.е. сократить объём вычислений.

3.4 Метод поиска минимума с крупным шагом

Для уменьшения количества операций можно применять следующую схему. Сначала перебрать всю область M с крупным шагом, т.е. перебирать не все точки, а пропускать их с каким-то шагом. В данной работе был выбран шаг 4. После этого надо обойти область вблизи вычисленного минимума.

Однако здесь есть определённые проблемы. Т.к. вид функции $X(F_1, F_2)$ от координат может быть очень сложный, то зачастую нельзя ограничиться окрестностью одной точки. Приходится обходить окрестность нескольких точек. Для этого вычисляется значение шума в отдельном пикселе. При этом считается, что шум в пикселях гауссов, независимый и с одинаковой дисперсией. Значение же функции $X(F_1, F_2)$ в минимуме принимается обусловленной только шумом. Тогда по формулам для χ^2 -распределения получаем:

$$\sigma = \sqrt{\frac{X_{\min}}{N_p}}, \text{ где} \quad (8)$$

X_{\min} – значение функции $X(F_1, F_2)$ в минимуме,

N_p – количество точек в области $M(dx, dy)$, соответствующей минимуму.

Далее вычисляется шумовой уровень по формулам:

$$L_i = X_{\min} + \frac{B \cdot \sigma}{\sqrt{N_{pi}}}, \text{ где} \quad (9)$$

B – выбираемый экспериментально порог (~ 3),

N_{pi} – количество точек в области $M(dx_i, dy_i)$.

Далее значение функции $X(F_1, F_2)$ в сравниваются с порогом, точка выбирается как претендент на минимум.

Далее под областью M подразумевается новая область M_2 . Она строится как объединение окружностей, построенных вокруг выбранных точек с радиусом, равным величине выбранного шага. И дальнейший поиск минимума производится уже в этой области M_2 .

Несмотря на введённые здесь предположения о характеристиках шума, которые на практике никогда не выполняется, этот метод работает устойчиво при достаточно большом значении параметра B .

3.5 Определение параметров искажений с использованием эллипсов ошибок

Т.к. в ходе поиска максимума были вычислены значения функции $X(F_1, F_2)$ в довольно большой области, то эту информацию можно использовать для улучшения качества оценки параметров преобразования. В частности, на её основе можно вычислить ковариационную матрицу ошибок оценивания сдвигов отдельных окон. Для этого, по той же схеме, что и при выборе точек, вокруг которых необходимо производить поиск минимума (т.е. по формулам (8), (9)) можно выбирать точки внутри итоговой области M_2 . Полученные точки методом наименьших квадратов надо аппроксимировать эллипсом вида:

$$A(x-x_{\min})^2+2B(x-x_{\min})(y-y_{\min})+C(y-y_{\min})^2=0, \text{ где} \quad (10)$$

(x_{\min}, y_{\min}) – точка минимума функции $X(F_1, F_2)$.

Тогда матрица

$$C^{-1} = \begin{vmatrix} A & B \\ B & C \end{vmatrix} \text{ является оценкой ковариационной матрицы ошибок оценивания}$$

сдвигов отдельных окон.

Тогда для оценки параметров аффинных искажений можно преобразовать формулу (4) так, чтобы в ней использовались вычисленные ковариационные матрицы, следующим образом:

$$Q = \arg \min_Q \sum_{k=1}^N (\vec{\rho}_k - \tau_Q(\vec{r}_k)) C_k^{-1} (\vec{\rho}_k - \tau_Q(\vec{r}_k))^T \quad (11)$$

Данная схема позволяет улучшить качество оценки параметров искажений, и, что более важно, позволяет не учитывать (или учитывать с малым весом) неточные измерения. Если процедура поиска минимума попала в локальный минимум, то эллипс ошибок обычно очень велик, и это позволяет автоматически не учитывать плохие измерения.

3.6 Методы определения сдвига менее одного пикселя

Для улучшения работы алгоритма желательно определять сдвиг с точностью более одного пикселя. Это особенно важно в рассмотренном ниже методе поиска сдвига по сжатым кадрам, т.к. там дробный сдвиг сжатого кадра может соответствовать большому целочисленному сдвигу исходного изображения.

Различные способы отыскания дробного сдвига подробно рассмотрены в статье [1]. Каждый из них можно использовать для уточнения сдвига и в данных алгоритмах.

Здесь, однако, есть одна особенность. Она заключается в том, что в ходе определения целочисленного сдвига уже были вычислены значения функции $X(F_1, F_2)$ в окрестности минимума. Т.к. сдвиг соответствует минимуму этой функции, то можно аппроксимировать её значения в дискретных точках гладкой поверхностью и найти минимум на этой поверхности. Удобно брать 9 точек вокруг максимума и методом наименьших квадратов находить параметры параболоида $z=ax^2+bxу+cy^2+dx+ey$. Этот метод дает несколько меньшую точность, чем рассмотренные в статье [1], но, в силу указанных выше причин, существенно быстрее работает.

3.7 Метод поиска сдвига по сжатым кадрам

Другим способом улучшения скорости (а зачастую и качества) работы алгоритма является процедура оценки сдвига по сжатым кадрам. Работает она по следующей итеративной схеме:

1. Для первого шага выбирается область D_1 как прямоугольник заданных размеров (как показывает практика, достаточно надёжно определяются сдвиги в пол кадра).
2. На k -ом шаге кадры сжимается в 2^{P-k} раз. Здесь P выбирается таким, чтобы на первом шаге после сжатия размер сторон кадра составлял величину 15 – 30 точек.
3. Вычисляется по формуле (2) значения функции $X(F_1, F_2)$ в области D_k .
4. Если шаг k – последний (но не обязательно $k=P$), то считается точный дробный сдвиг (см. главу 3.6) около точки минимума, и значения сдвига приводятся к начальному масштабу кадра. На этом итеративная схема заканчивается.
5. Выбирается область D_{k+1} из следующих соображений. Выбираются точки, значение функции $X(F_1, F_2)$ в которых, не более чем в z_k раз больше значений в минимуме. Значения z_k подбираются экспериментально ($\sim 1.01 - 1.5$).
6. Переходится на следующий шаг ($k+1$) на пункт 2.

Однако эту процедуру нецелесообразно проводить до шага P , т.к. сравнение целого кадра требует очень большого количества операций. Удобнее сделать 2 – 3 шага, а затем перейти к описанной выше процедуре определения параметров преобразования с использованием нескольких окон.

При этом в качестве области возможных сдвигов D надо выбирать окрестность уже измеренного сдвига. Здесь возможно существенное улучшение. Т.к. сдвиг уже измерен, и нам следует лишь уточнить его, то нет вероятности попасть в локальный минимум. Значит

можно использовать для поиска минимума процедуры спуска, например, дискретный градиентный спуск.

Метод с использованием сжатия кадра обладает существенными преимуществами по сравнению с другими вышеописанными. Во-первых, он позволяет вычислять сдвиги до полукадра, что невозможно другими способами ввиду большого количества требуемых операций. Во-вторых, вероятность ошибок существенно меньше, т.к. можно полностью проверить все возможные варианты сдвигов, а значит нельзя попасть в локальный минимум. Также существенно, что здесь сравнивается весь кадр, а не какая-то его часть, что опять приводит к уменьшению количества ошибок. В-третьих, этот метод работает несколько быстрее описанных выше. В-четвертых, он гораздо лучше работает со смазанными кадрами.

Также следует заметить, что, несмотря на то, что при сжатии не определяются угол поворота и коэффициент масштабирования, они вычисляются в процедуре уточнения, а их наличие в кадре не сказывается на работоспособности алгоритма определения сдвига с помощью масштабирования.

3.8 Итеративный алгоритм

Все вышеописанные алгоритмы могут определять поворот до 5° - 10° . Однако большой сдвиг определяется с ошибками, т.к. везде используется сравнение кадров или окон без их поворота. Но, несмотря на это, ошибка определения угла не очень велика. Тогда, если запустить этот алгоритм снова к преобразованному кадру, то итоговая точность оценки будет существенно лучше. Проблемой здесь является необходимость интерполяции кадра в соответствии с полученными на первом шаге значениями угла поворота. Однако, если интерполировать не весь кадр, а только выбранные окна, можно добиться существенного увеличения точности при незначительном уменьшении скорости (менее чем в полтора раза).

3.9 Особенности алгоритмов

Помимо собственно оценки параметров деформации в форме, сформулированной в главе 2, были решены также задачи, выпадающие из математической модели, но без решения которых невозможно качественно решить задачу стабилизации изображения.

3.9.1 Шумоподобные кадры

Зачастую на вход алгоритма подаётся видеопоследовательность монотонных кадров. Т.е. кадров одного цвета (возможно зашумлённых). Тогда, в силу случайности шумов, функция $X(F_1, F_2)$ имеет минимум, который в подавляющем большинстве случаев

находится не в точке $(dx,dy)=(0,0)$. Это приводит к сильным колебаниям выходного изображения при статичном входном.

Простое сравнение входного и выходного кадра не даёт результата, т.к. при наличии шумов они не абсолютно идентичны. Тогда встаёт проблема нахождения порога, по которому отсеиваются подобные последовательности. Задача выбора такого порога, работающего адекватно на любых последовательностях, не может быть решена простыми способами без вычисления уровней шумов.

Однако уровни шумов уже были вычислены при работе алгоритма. Метод их вычисления описан в главе 3.4. Если же используется метод со сжатием, то можно отдельно вычислить область значений, близких к минимальному в смысле шумоподобности. Если размер такой области велик, или велик эллипс ошибок (например, содержит более 20 точек), то последовательность принимается за шумоподобную (а значит входной кадр недеформирован).

3.9.2 Кадры с линейной структурой

Часто на вход алгоритма подаётся видеопоследовательность кадров, имеющих линейную структуру. Например, это может быть стандартная телевизионная заставка. Здесь та же проблема, что и в шумоподобных кадрах, но ошибка измерения сдвига может быть большой только вдоль одного направления.

Здесь, как и в предыдущем случае надо построить область, близкую к минимуму (исходя из уровня шума). Затем, если эта область имеет линейный вид, принимается решение о линейности входных кадров.

Здесь возможно две альтернативы. Во-первых, можно прекратить обработку такой последовательности и на выходе выдавать то же, что и на входе.

Во-вторых, можно подсчитать только сдвиг в направлении, перпендикулярном к линейной структуре в кадре, и угол поворота. Для этого надо увеличить (в пределе до бесконечности) полуось эллипса ошибок вдоль линейной структуры. Затем, в реализации функции нахождения минимума (11) (т.е. в функции решения соответствующей линейной системы) ввести регуляризацию, описанную, например, в [6].

3.9.3 Кадры с чересстрочной развёрткой

Зачастую кадры видеопоследовательности снимаются чересстрочной развёрткой, т.е. четные и нечетные строчки снимаются в разные моменты времени. При сильной тряске камеры это приводит к сильным искажениям изображения, когда принятая математическая модель вовсе перестаёт быть справедливой. Однако, полукадры,

собранные только из четных (или нечетных) строк имеют только обычные, указанные в главе 2 деформации.

Для получения качественного выходного изображения можно использовать только один полукадр, либо последовательно на вход алгоритма подавать сначала кадр из нечетных, затем кадр из четных строк. Тем самым будет решена задача стабилизации изображения, но при этом уменьшится формальная разрешающая способность вдоль вертикальной оси. Однако, в силу того, что на входном кадре изображение было размазано с амплитудой колебания оптической оси, фактическая разрешающая способность выходного кадра повышается.

3.9.4 Смазанные кадры

Зачастую входные кадры могут быть смазаны за счет сильных колебаний камеры (или испорчены каким-либо другим образом). Но, т.к. эти искажения носят случайный характер, и сильно меняются по степени от кадра к кадру, то для улучшения качества выходного изображения и качества работы алгоритма определения деформации можно пропускать кадры, испорченные сильнее других.

Критерием качества здесь может считаться информативность кадра, вычисленная по формуле (7). Сам же критерий пропуска выбирается из других соображений, например, сколько кадров можно пропустить подряд, насколько сильно может ухудшаться изображение и т.д.

4. Оценка качества и скорости работы

4.1 Методы исследований

Качество работы алгоритмов оценивалось двумя способами.

Во-первых, на основе модельных кадров. Т.е. одно и то же изображение преобразовывалось с заданными параметрами и подавалось на вход алгоритма. При этом можно точно оценить ошибки определения параметров преобразования. В тоже время, такое моделирование не отражает многих проблем, с которыми приходится сталкиваться в реальности.

Во-вторых, для тестирования использовались реальные видеопоследовательности. По ним можно определить сдвиг лишь приближенно и субъективно, и здесь основным критерием должно являться вероятность сбоя, т.е. «очевидной» ошибки более нескольких пикселей.

При оценке качества работы алгоритмов особое внимание уделялось их скорости.

4.2 Особенности программной реализации

Все представленные алгоритмы реализованы и тестировались на машине со следующими параметрами:

- Процессор Intel Pentium 4 2.4ГГц
- 256Кб памяти
- Видеокарта NVIDIA GeForce4 MX 440
- Операционная система Windows 2000
- Среда разработки Microsoft Visual Studio .NET 7.0
- Язык программирования C++
- Компилятор Intel® C++ Compiler 7.1

При этом особо учитывалась архитектура компьютера и применялись специфичные для неё средства. В частности, использовалась библиотека функций Intel® Integrated Performance Primitives v3.0, и SSE2 инструкции сопроцессора.

4.3 Анализ алгоритмов определения сдвигов

В таблице 1 даны результаты работы различных алгоритмов определения сдвига. При этом использовался кадр размером 640x480. Сдвиги моделировались случайные по обеим осям, равномерно распределенные в интервале от -20 до $+20$ пикселей.

| | СКО ошибки определения сдвига | Количество обрабатываемых кадров в секунду |
|--|----------------------------------|--|
| Сравнение всего кадра (глава 3.2) | 0.04-0.10 | 7 |
| Метод корреляционных окон (главы 3.3 – 3.6) | 0.04-0.10 | 230 |
| Метод, использующий сжатие кадров (глава 3.7) | 0.04-0.10 | 350 |

Таблица 1.

Т.е. при равных результатах, скорости работы различных алгоритмов существенно отличаются. Особо следует заметить, что трудоемкость первого алгоритма растёт пропорционально количеству точек кадра (т.е. квадратично относительно длины

стороны). Трудоёмкость же второго и третьего растёт существенно медленнее (приблизительно пропорционально размеру стороны).

На реальных последовательностях их работоспособность тоже выглядит приблизительно одинаково. Соотношение скоростей также сохраняется. Однако вероятность сбоя (особенно на последовательностях со смазами) для третьего алгоритма приблизительно в десять раз меньше, чем для второго. Также второй алгоритм даёт сбои, если сдвиг превышает априорный диапазон, что также случается на реальных последовательностях.

Также была протестирована вся процедура стабилизации, включающая чтение кадров из файла или с камеры, определение сдвига и отображение входного и выходного кадров на экран. В таком случае возможно обрабатывать с теми же величинами ошибок до 30 кадров в секунду размером 1152x854.

4.4 Анализ алгоритмов определения сдвигов и поворотов

В таблице 2 даны результаты работы различных алгоритмов определения сдвига и поворота. При этом использовался кадр размером 640x480. Сдвиги моделировались случайные по обеим осям, равномерно распределенные в интервале от -20 до $+20$ пикселей. Поворот также выбирался случайным, равномерно распределённым в интервале от -2.5° до $+2.5^\circ$.

| | СКО ошибки определения сдвига | СКО ошибки определения угла поворота | Количество обрабатываемых кадров в секунду |
|---|-------------------------------|--------------------------------------|--|
| Метод корреляционных окон (главы 3.3 – 3.6) | 0.2-0.4 | 0.05-0.1 | 200 |
| Метод, использующий сжатие кадров (глава 3.7) | 0.1-0.3 | 0.2-0.3 | 350 |
| Итеративный метод (глава 3.8) | 0.1-0.2 | 0.01-0.05 | 200 |

Таблица 2.

Результаты в целом схожи с теми, что были и при не вращающихся входных кадрах.

Здесь стоит отметить следующее. Два первых метода начинают давать значительные сбои при угле между кадрами, превышающем 10° . Итеративный же алгоритм может с той же точностью определять углы до 15° - 20° .

Что касается реальных, а не модельных последовательностей, то здесь имеет место та же ситуация, что и в предыдущей главе. Стоит только заметить, что при больших углах поворота алгоритм со сжатием начинает давать сбои несколько чаще других алгоритмов.

5. Выводы

Были созданы и реализованы различные алгоритмы компенсации геометрических линейных искажений кадров видеопоследовательности. При этом использовались только алгоритмы, основанные на вычислении среднеквадратичной невязки кадров или их частей.

Было показано, что такие алгоритмы на модельных последовательностях дают ошибку измерения сдвигов кадров не более 0.2 пикселей при максимальном значении сдвига в пол кадра, а ошибку определения углов поворота – менее 0.05 при угле поворота в 15° - 20° .

Что существенно, они дают хорошие результаты и при работе с реальными последовательностями, даже снятыми в сложных условиях и при искажениях, не описываемых используемой математической моделью. При этом колебания выходного изображения не превышают 1 пиксель с вероятностью по крайней мере 0,999.

Использование в качестве основного критерия среднеквадратичной невязки позволило создать достаточно быстрые алгоритмы. Это связано с лёгкостью вычисления и распараллеливания данной функции. В результате данные алгоритмы работают со скоростью до 350 кадров в секунду при размере кадра 640×480 . Вообще данные алгоритмы позволяют проводить обработку (включающую в себя, помимо алгоритмов определения параметров деформации, также считывание исходных данных и отображение на экран) до 30 кадров размером в 1.3 мегапикселя в секунду.

Ещё одним преимуществом данных алгоритмов является легкая их реализуемость на специализированных сигнальных процессорах.

В целом, разработка и тестирование комплекса алгоритмов показали принципиальную возможность реализации обработки больших потоков видео информации в реальном времени на современных, неспециализированных вычислительных средствах.

6. Литература

1. А.К. Ким, А.Е. Колесса, В.Н. Лагуткин, А.В. Лотоцкий, В.Г. Репин. Алгоритмы идентификации и подавления нестационарного мешающего фона и повышение разрешающей способности в оптическом сенсоре с хаотически колеблющейся оптической осью и динамическими аберрациями. *Радиотехника*, 1998, №12
2. В. К. Р. Horn and E. J. Weldon, Jr. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76, 1988.
3. R. Chipolla, Y. Okamoto, and Y. Kuno, “Robust Structure from Motion Using Motion Parallax,” *Int’l Conf. Computer Vision*, pp. 374–382, Berlin, May 1993.
4. F. Lustman, O.D. Faugeras, and G. Toscani, “Motion and Structure from Motion from Point and Line Matching,” *Proc. First Int’l Conf. Computer Vision*, pp. 25–34, London, 1987.
5. S. Negahdaripour and S. Lee, “Motion Recovery from Image Sequences Using First-Order Optical Flow Information,” *IEEE Workshop Visual Motion*, pp. 132–139, Princeton, N.J., Oct. 1991.
6. Й. Абаффи, Э. Спендикато. Математические методы для линейных и нелинейных уравнений. Проекционные ABS-алгоритмы. Мир, 1996.