
ИНФОРМАТИКА И УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ И СОЦИАЛЬНЫХ СИСТЕМАХ

УДК 004.932

DOI: 10.46960/1816-210X_2023_4_7

МЕТОД АВТОМАТИЗИРОВАННОЙ КАЛИБРОВКИ КАМЕР И СШИВКИ ВИДЕОИЗОБРАЖЕНИЙ В МНОГОКАМЕРНЫХ СИСТЕМАХ ВИДЕОНАБЛЮДЕНИЯ

М.Б. БагировORCID: 0000-0003-1656-0849 e-mail: bagirov_mirabbas@mail.ruНижегородский государственный технический университет им. Р.Е. Алексеева
*Нижний Новгород, Россия***Э.С. Соколова**ORCID: 0000-0003-0860-2463 e-mail: essokolowa@gmail.comНижегородский государственный технический университет им. Р.Е. Алексеева
Нижний Новгород, Россия

Разработан алгоритм сшивки видеоизображений, представляющий собой комбинацию алгоритмов калибровки, цветовой коррекции и градиентного смешивания с целью минимизации видимости линии шва между стыкуемыми изображениями, получаемыми с разных камер. Алгоритм калибровки камер в многокамерной системе автоматизирован, что минимизирует участие оператора и повышает точность за счет обнаружения и поэтапного отсеивания неудачных пар опорных точек в поле пересечения углов зрения камер. Обосновано, что для одного кадра общее время, затрачиваемое на цветокоррекцию, градиентное смешение, сшивку и обрезку изображения составляет около 35 миллисекунд, что соответствует требованиям работы алгоритма построения панорамы в режиме реального времени для многокамерной системы видеонаблюдения.

Ключевые слова: многокамерные системы, калибровка, устранение дисторсии, сшивка панорамы, OpenCV, видеоаналитика.

ДЛЯ ЦИТИРОВАНИЯ: Багиров, М.Б. Метод автоматизированной калибровки камер и сшивки видеоизображений в многокамерных системах видеонаблюдения / М.Б. Багиров, Э.С. Соколова // Труды НГТУ им. Р.Е. Алексеева. 2023. № 4. С. 7-19. DOI: 10.46960/1816-210X_2023_4_7

METHOD OF AUTOMATED CAMERA CALIBRATION AND STITCHING VIDEO IMAGES IN MULTI-CAMERA VIDEO SURVEILLANCE SYSTEMS

M.B. BagirovORCID: 0000-0003-1656-0849 e-mail: bagirov_mirabbas@mail.ruNizhny Novgorod State Technical University n.a. R.E. Alekseev
*Nizhny Novgorod, Russia***E.S. Sokolova**ORCID: 0000-0003-0860-2463 e-mail: essokolowa@gmail.comNizhny Novgorod State Technical University n.a. R.E. Alekseev
Nizhny Novgorod, Russia

Abstract. The paper presents the developed stitching video images algorithm, in which a combination of calibration algorithm, color correction and gradient mixing algorithm is carried out to minimize the visibility of the seam line between joined images obtained from different cameras. The algorithm for calibrating cameras in a multi-camera system is automated, which minimizes operator participation and increases the accuracy of calibration by detecting and gradually screening out unsuccessful pairs of reference points in the field of intersection of camera angles. It is confirmed that the total time spent on color correction, gradient mixing, stitching and cropping the image is about 35 milliseconds for one frame, which meets the requirements of the real-time panorama algorithm for a multi-camera video surveillance system.

Key words: multi-camera systems, calibration, distortion elimination, panorama stitching, OpenCV, video analytics.

FOR CITATION: M.B. Bagirov, E.S. Sokolova. Method of automated camera calibration and stitching video images in multi-camera video surveillance systems. Transactions of NNSTU n.a. R.E. Alekseev. 2023. № 4. Pp. 7-19. DOI: 10.46960/1816-210X_2023_4_7

1. Введение

Системы видеоаналитики стали неотъемлемой частью современного мира. В последние годы доступность и эффективность видеонализа делает его все более популярным в различных сферах социальных технологий, включая безопасность, контроль доступа, наблюдение за транспортом, медицинское обслуживание и т.д. Актуальность данных систем обусловлена их способностью автоматизировать процесс анализа видео и сокращать время ручной обработки информации. Кроме того, системы видеоаналитики повышают эффективность работы производств и компаний, снижая затраты на персонал. Точность и производительность их работы во многом зависит от предварительной настройки параметров.

2. Анализ проблемы и постановка задач исследования

Важным направлением видеоаналитики является панорамная сшивка – создание бесшовных панорамных изображений из данных, полученных от множества камер. Известные в настоящее время алгоритмы работают с неподвижными статичными изображениями, тогда как технологии создания панорам на основе видеоданных существенно различаются. Импортозамещение требует создания отечественных систем панорамного обзора высокого качества, устраняющих «слепые» зоны и обеспечивающих бесшовное сшивание панорамных изображений, при этом особенно важна работа систем в режиме реального времени.

В работе [1] представлена классификация методов и средств формирования панорамных изображений, рассмотрены методы получения данных о глубине пространства и устройства получения панорамных изображений, выявлены их преимущества и недостатки. Проведен аналитический обзор, позволяющий учесть специфику предлагаемых решений с целью дальнейшей разработки программно-аппаратных решений для построения цифровой модели пространства в режиме реального времени.

В настоящее время существуют два основных метода создания панорам: сглаживание переходов между фрагментами и минимизация видимых швов между изображениями. Однако оба метода имеют свои ограничения, связанные с высокими вычислительными затратами и требованиями к памяти. В работе [2] рассматривается метод поиска швов в области перекрытия полей зрения камер. Он позволяет минимизировать различия между исходными изображениями, однако требует больших вычислительных ресурсов и занимает много времени. Кроме того, для работы метода необходимо наличие большого количества памяти. Для повышения скорости работы и уменьшения требований к ресурсам в работе [3] предлагается алгоритм оптимизации построения панорам путем вычисления кратчайших путей на графах. Однако точность этого метода оказалась ниже, чем ожидалось, что говорит о необходимости доработки алгоритма для повышения точности и применения его в режиме реального времени. Самым популярным алгоритмом для сшивки изображений в панораму является RANSAC

[4-5], однако в последние годы появляется все больше исследований альтернативных методов для решения данной задачи. Это во многом связано с недостатками RANSAC – большими временными затратами и наличием ошибок в поиске матрицы гомографии. При использовании алгоритмов сшивки панорам в многокамерных системах возникает проблема цветокоррекции [6]. Различия в настройках экспозиции, балансе белого и положении камер могут привести к несоответствию цветов в итоговой панораме. Это становится очень заметным, если входные изображения получены с разных камер.

Существующие методы и алгоритмы панорамной сшивки, представленные на рынке, могут показывать хорошие результаты, но применение большого числа камер усложняет процесс обработки информации и не дает возможности обрабатывать данные и принимать решения в режиме реального времени. Необходима разработка и реализация алгоритма, комбинирующего калибровку системы камер, алгоритм цветокоррекции входных изображений с алгоритмом создания панорамы, что позволит повысить скорость работы системы построения панорамы без потери точности.

3. Разработка и исследование алгоритмов

Создание панорамного видео в общем виде включает регистрацию входных изображений, калибровку камер (поиск опорных точек, вычисление и уточнение матриц гомографии), проецирование изображений, цветовую коррекцию и сшивку изображений, включающую определение оптимальной границы сшивания, градиентное смешивание вдоль этой границы и определение границ обрезки.

3.1. Алгоритм калибровки многокамерных систем

Процесс калибровки камер является первым этапом при построении панорамного обзора в многокамерной системе для решения задач видеоаналитики. Предлагаемый алгоритм представляет собой модификацию метода калибровки камер на основе опорных точек, выбранных в пересечении полей зрения двух камер. На начальном этапе калибровки любых двух камер системы, имеющих пересекающиеся поля зрения, выбираются четыре точки, координаты которых задает оператор, и выполняется ручная калибровка этой пары камер. В результате получаем матрицы гомографии откалиброванных камер, которые используются для автоматической калибровки остальных камер системы.

Алгоритм ручной калибровки двух камер системы с пересекающимися полями зрения заключается в следующем. Имеется система из n камер, которые представлены в виде множества $S = \{C_1, C_2, \dots, C_n\}$. Выберем две камеры с пересекающимися полями зрения C_1 и C_2 (рис. 1), в области пересечения полей зрения которых выберем четыре опорных точки, расположенные в одной плоскости, координаты которых известны, и которые расположены по сторонам прямоугольника на плоскости земли [7]. При этом прямоугольник между четырьмя точками должен занимать более 25 % площади пересечения полей зрения двух камер [8]. Обозначим P^i – набор из четырех точек $p_j^i, j \in \{1, 2, 3, 4\}$ в поле зрения i -й камеры, P – набор точек в мировой системе координат. В результате проведения ручной калибровки двух камер C_1 и C_2 на выходе получим три набора, состоящих из четырех точек с координатами:

$$P = \{p_1, p_2, p_3, p_4\}, P^1 = \{p_1^1, p_2^1, p_3^1, p_4^1\}, P^2 = \{p_1^2, p_2^2, p_3^2, p_4^2\}. \quad (1)$$

Координаты точки p_j^i в системе координат камеры C_i определим следующим образом:

$$p_j^i = (x_{pj}^i, y_{pj}^i). \quad (2)$$

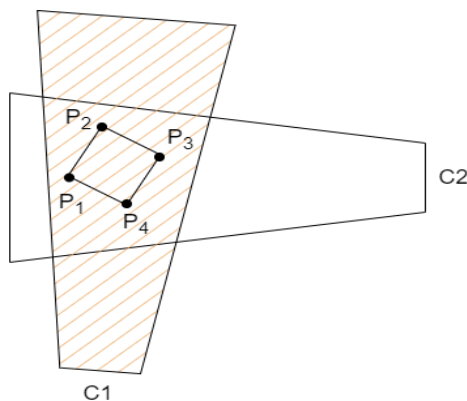


Рис. 1. Выбор опорных точек для калибровки камер

Fig. 1. Selecting reference points for camera calibration

С помощью полученных наборов точек строятся матрицы гомографии для каждой камеры. H_g^i – матрица гомографии, отображающая пиксельные координаты камеры C_i ($i = 1, \dots, n$) в мировую систему координат. Чтобы найти матрицу гомографии для камеры, составляется система уравнений для каждой пары из четырех точек (3):

$$(x_{pj}, y_{pj}, 1)^T = H_g(x'_{pj}, y'_{pj}, 1)^T, \quad (3)$$

где x'_{pj}, y'_{pj} – координаты по оси X и Y точки p_j в мировой системе координат; x_{pj}, y_{pj} – координаты по оси X и Y точки p_j в системе координат камеры, $j \in \{1, 2, 3, 4\}$.

Алгоритм ручной калибровки двух камер системы изображен на рис. 2.

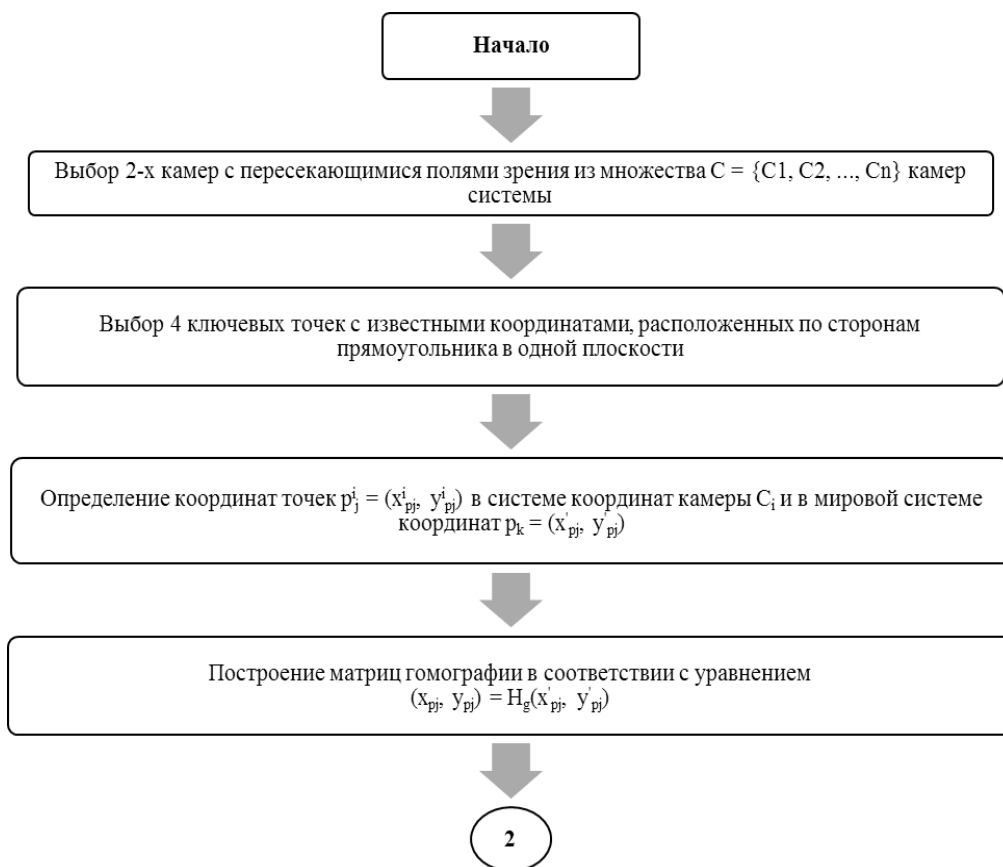


Рис. 2. Алгоритм ручной калибровки двух камер системы с пересекающимися углами обзора

Fig. 2. Algorithm for manual calibration of two cameras of a system with overlapping viewing angles

В результате ручной настройки двух камер системы построены их матрицы гомографии, используемые для автоматической калибровки остальных камер системы по разработанному методу, включающему алгоритмы поиска опорных точек в пересечении полей откалиброванной и неоткалиброванной камер и алгоритм уточнения матрицы гомографии калибруемой камеры. На рис. 3 представлен алгоритм поиска опорных точек в пересечении полей зрения двух камер, позволяющий автоматически откалибровать и вычислить матрицу гомографии очередной j -ой камеры системы.

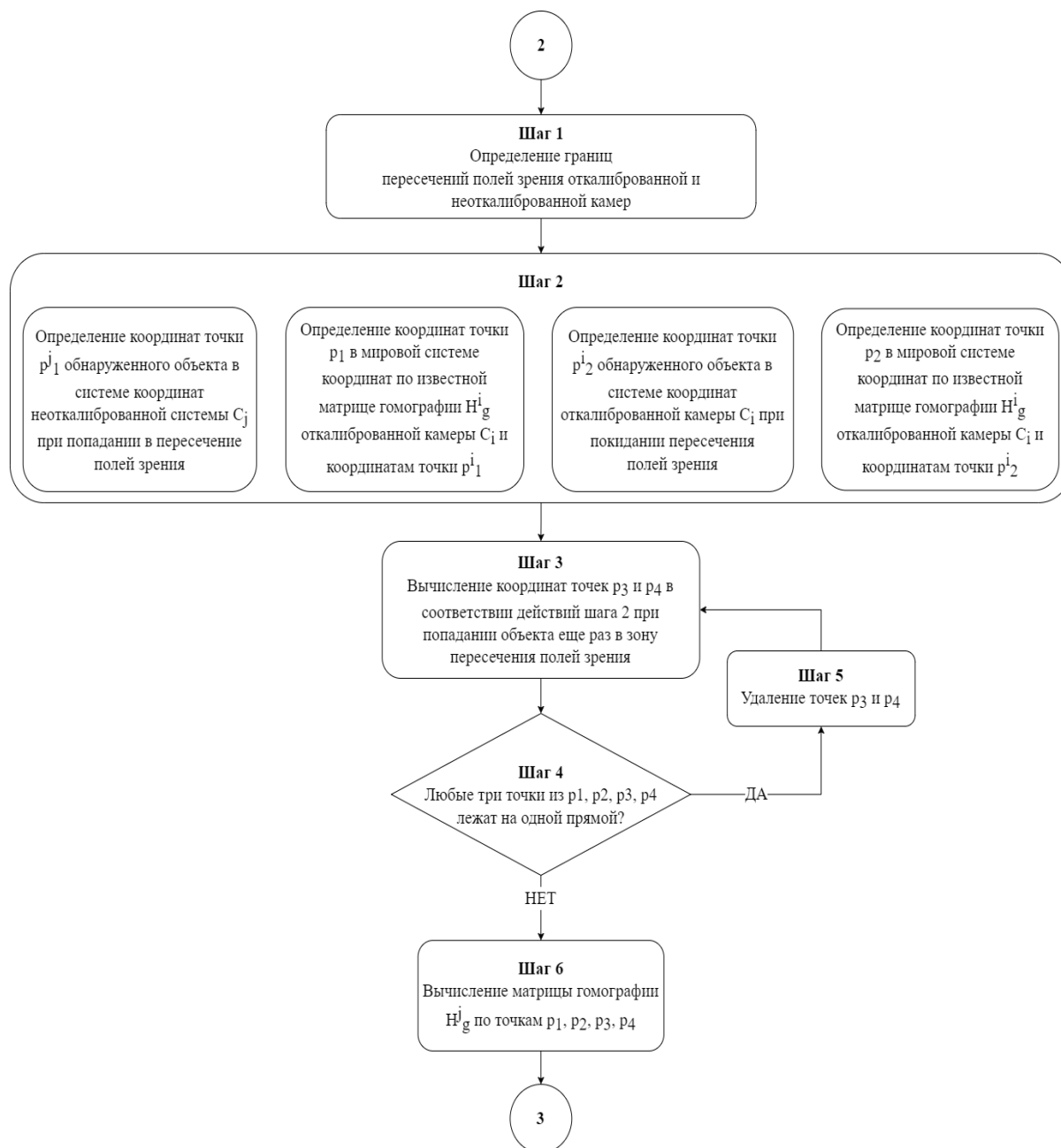


Рис. 3. Поиск опорных точек в пересечении полей зрения двух камер для вычисления матрицы гомографии неоткалиброванной камеры

Fig. 3. Search for reference points in the intersection of the fields of view of two cameras to calculate the homography matrix of the uncalibrated camera

Для нахождения четырех опорных точек в области пересечения полей зрения камер сначала определяются границы пересечений полей зрения камер с помощью алгоритма P. Lehel [9]. Координаты первой точки в мировой системе координат вычисляются по матри-

це гомографии откалиброванной камеры при попадании объекта в поля зрения откалиброванной и неоткалиброванной камер (т.е. при первом появлении ID на изображениях двух камер), а координаты второй – при покидании объектом пересечения полей зрения камер. Аналогично вычисляются координаты третьей и четвертой опорных точек. Далее выполняется алгоритм проективного преобразования для уточнения матрицы гомографии калибруемой камеры, что повышает точность автоматической калибровки камер. После выполнения автоматической калибровки камеры S_j координаты сопровождаемого объекта в рамках полей зрения камер системы видеонаблюдения вычисляются по известным матрицам гомографии и сохраняются в карту перемещений.

Полученные координаты используются для уточнения матриц гомографии следующим образом. Когда объект попадает в поле зрения двух камер, вычисляются его координаты p_5^i и p_5^j в пересечении полей зрения камер S_i и S_j в мировой системе координат. Значения координат, как правило, не совпадают из-за погрешности вычисления матрицы гомографии автоматически калибруемой камеры и погрешности работы детектора движущегося объекта. Эти значения далее используются для уточнения матрицы гомографии калибруемой камеры. Для камер S_i и S_j на основе координат точек p_1, p_2, p_3, p_4 в мировой системе координат и в системе координат камеры строится матрица системы A для каждой из камер. Матрицу A в блочной форме можно представить так (4):

$$A = \begin{bmatrix} Z & O & W_1 \\ O & Z & W_2 \end{bmatrix}, \quad (4)$$

$$\text{где } Z = \begin{bmatrix} x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 \\ x'_4 & y'_4 & 1 \end{bmatrix}, O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, W_1 = \begin{bmatrix} -x_1x'_1 & -x_1y'_1 \\ -x_2x'_2 & -x_2y'_2 \\ -x_3x'_3 & -x_3y'_3 \\ -x_4x'_4 & -x_4y'_4 \end{bmatrix}, W_2 = \begin{bmatrix} -y_1x'_1 & -y_1y'_1 \\ -y_2x'_2 & -y_2y'_2 \\ -y_3x'_3 & -y_3y'_3 \\ -y_4x'_4 & -y_4y'_4 \end{bmatrix}, x'_{pj}, y'_{pj}$$

– координаты по оси X и Y точки p_j в мировой системе координат; x_{pj}, y_{pj} – координаты по оси X и Y точки p_j в системе координат камеры.

Используя матрицу A , построим систему линейных алгебраических уравнений (СЛАУ), преобразованную к стандартной форме записи следующего вида:

$$A * G_H = B, \quad (5)$$

где A – матрица системы камеры, $G_H = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})^T$ – вектор-столбец элементов матрицы гомографии, подлежащих определению, $B = (x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)^T$ – вектор-столбец координат опорных точек из пересечения полей зрения двух камер, записанных в указанном порядке $(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)$.

Выполним переход к виду нормальной СЛАУ, помножив части выражений (5) на матрицу A^T :

$$(A^T * A) * G_H = A^T * B. \quad (6)$$

Переход от (5) к (6) эквивалентен поиску преобразования матрицы гомографии H для минимизации евклидова расстояния между $H*x$ и соответствующей ей точкой x' . При решении СЛАУ методом квадратного корня для 5 точек необходимое условие минимума (7):

$$\frac{\partial F(G_H)}{\partial G_H} = 0, \partial F(G_H) = \sum_{i=1}^k ((x_i - \phi_1(x_i, x'_i, y'_i))^2 + (y_i - \phi_2(y_i, x'_i, y'_i))^2), \quad (7)$$

$$\phi_1 = x'_i h_{11} + y'_i h_{12} + h_{13} - x_i x'_i h_{31} - x_i y'_i h_{32},$$

$$\phi_2 = x'_i h_{21} + y'_i h_{22} + h_{23} - y_i x'_i h_{31} - y_i y'_i h_{32},$$

где x'_i, y'_i – координаты по оси X и Y точки p_i в мировой системе координат; x_i, y_i – координаты по оси X и Y точки p_i в системе координат автоматически откалиброванной ка-

меры, $i = 1, \dots, k$, $(h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})$ – элементы матрицы гомографии камеры. Тогда элементы матрицы гомографии G_H находятся из уравнения (6):

$$\begin{pmatrix} Z^T Z & 0 & W_1 \\ 0 & Z^T Z & W_2 \\ W_1^T * Z & W_2^T * Z & W_1^T * W_1 + W_2^T * W_2 \end{pmatrix} * G_H = \begin{pmatrix} Z^T B_1 \\ Z^T B_2 \\ W_1^T * B_1 + W_2^T * B_2 \end{pmatrix}. \quad (8)$$

После выполнения операций уточнения и построения новой матрицы гомографии калибруемой камеры вновь вычисляем координаты пяти точек в мировой системе. Проводим вычисление евклидова расстояния между всеми парами точек и удаляем ту пару, расстояние между образами которых оказалось наибольшим, после чего из карты перемещений выбираем для вычислений новую пару точек и повторяем алгоритм уточнения матрицы гомографии. Реализуем данный алгоритм, пока евклидово расстояние для всех точек не будет меньше или равно требуемой погрешности. Алгоритм уточнения матрицы гомографии калибруемой j -й камеры приведен на рис. 4.

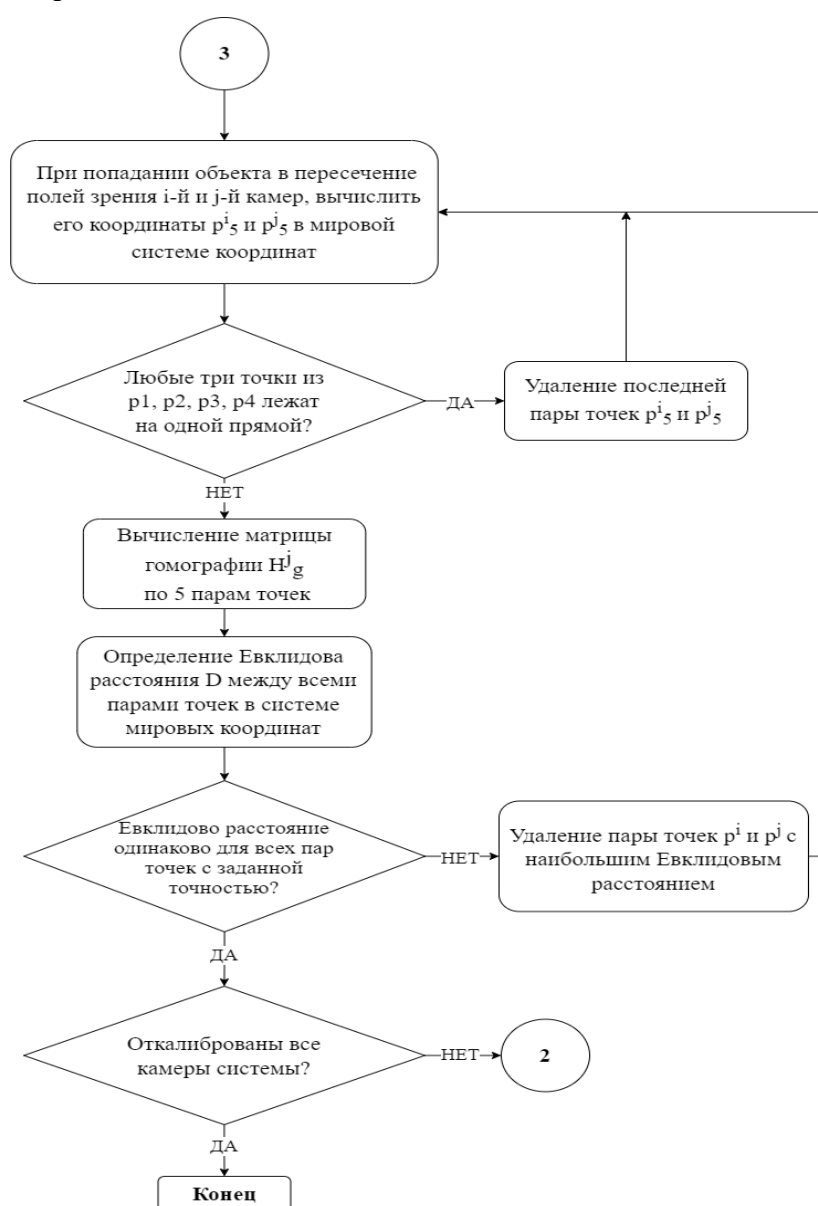


Рис. 4. Алгоритм уточнения матрицы гомографии автоматически калибруемой камеры

Fig. 4. Algorithm for refining the homography matrix of an automatically calibrated camera

Алгоритмы автоматической калибровки очередной камеры системы и уточнения ее матрицы гомографии с целью повышения точности автоматической калибровки проводятся для каждой пары откалиброванной и неоткалиброванной камер системы до тех пор, пока все камеры не будут откалиброваны.

3.2. Алгоритм сшивки видео в многокамерных системах

При сшивке изображений с видеопотоков в единую панораму будем использовать цилиндрическую проекцию, так как прямоугольная система координат может существенно преувеличить перспективу по мере увеличения угла обзора, что приводит к видимому завалу объектов к границам кадра. Для проведения панорамной сшивки предлагается использовать комбинацию алгоритмов цветовой коррекции и градиентного смешивания, что минимизирует видимость шва между стыкуемыми изображениями. Для выбора оптимального алгоритма цветовой коррекции проведено исследование двух подходов – цветовой коррекции на основе опорного изображения и линейной цветовой коррекции. В результате выполнено их объединение для повышения точности результата. Алгоритм цветовой коррекции на основе опорного изображения представлен на рис. 5. На первом шаге выбирается опорное изображение, на 2-5 шагах остальные изображения с камер подстраиваются под его уровень распределения яркости.

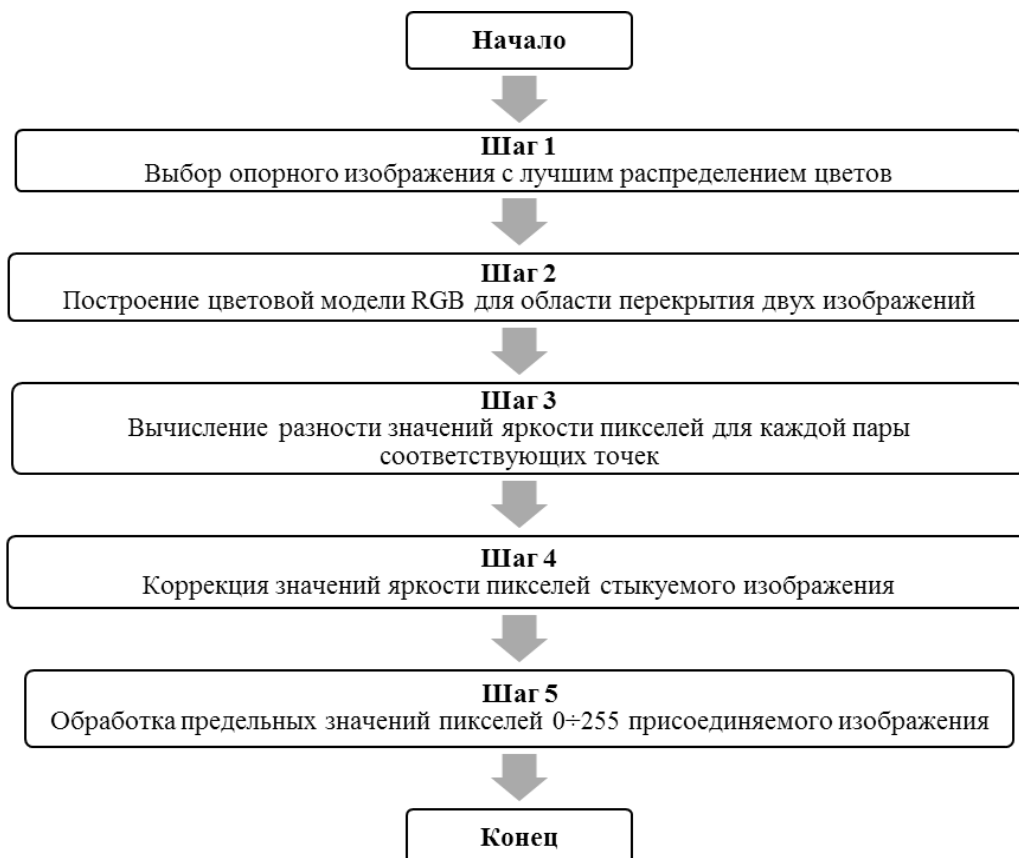


Рис. 5. Выполнение цветовой коррекции на основе опорного изображения

Fig. 5. Color correction based on the reference image

Шаг 1. При выборе опорного изображения, на основе которого будет проводиться коррекция цвета соседних, для каждого из поступающих изображений с видеопотоков камер рассчитывается распределение цветов в каждом из каналов RGB, и строится гистограмма распределения цветов. При построении гистограммы используем метод *cv2.calcHist* библио-

теки *OpenCV*. Для нахождения изображения с лучшим распределением цветов рассчитываем среднее значение яркости каждого из каналов RGB для всех полученных с камер сцены изображений, а также среднеквадратичное отклонение яркости от этих значений. Затем выбираем изображение, для которого среднеквадратичное отклонение наименьшее, и считаем его опорным.

Шаг 2. Область перекрытия двух изображений содержит характеристики распределения яркости этих двух изображений. Построим цветовую модель RGB двух изображений в области общего поля зрения камер, для чего определим значения яркости его R, G и B составляющих из RGB-гистограммы и рассчитаем средние значения серого цвета для каждой пары опорных точек M1 и M2 из области пересечения полей зрения камер (9):

$$M_k = \frac{1}{w \cdot h} \sum_{j=0}^h \sum_{i=0}^w \frac{R_k + G_k + B_k}{3}, \quad k = \overline{1,2}, \quad (9)$$

где w – ширина изображения, h – высота изображения.

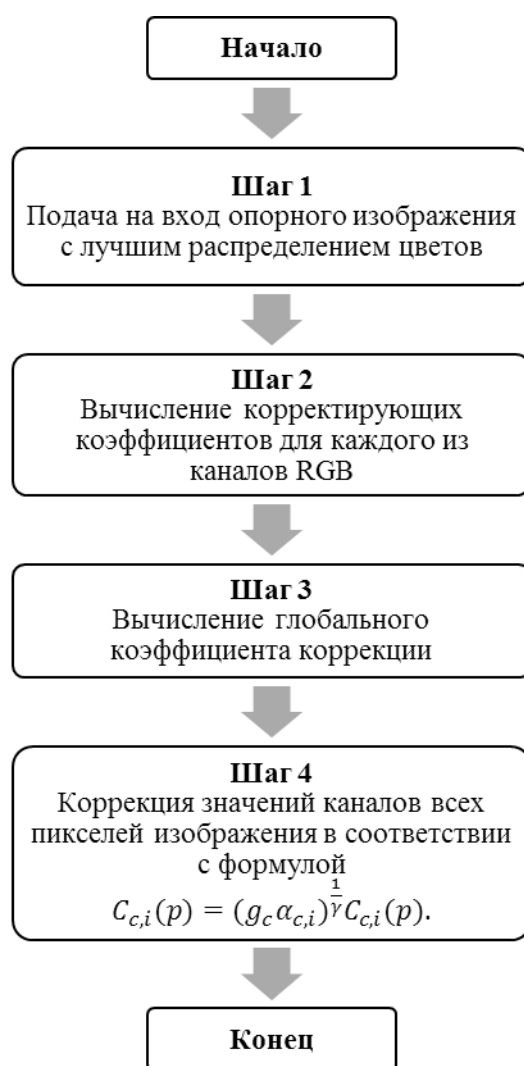


Рис. 6. Алгоритм линейной цветовой коррекции

Fig. 6. Linear color correction algorithm

Шаг 3. Для каждой пары соответствующих точек в области перекрытия двух изображений вычислим разность значений яркостей пикселей (10):

$$diff = M_1 - M_2. \quad (10)$$

Шаг 4. Значения яркости всех пикселей стыкуемого изображения корректируем на полученную величину (11):

$$Dst(i, j) = Res(i, j) + diff, \quad (11)$$

где $Res(i, j)$ – значение пикселей до коррекции, $Dst(i, j)$ – скорректированное значение пикселей.

Шаг 5. Так как максимальное значение яркости пикселей не может превышать значения 255, при коррекции цвета по формуле (11) необходимо выполнить обработку предельных значений пикселей – 0 и 255 присоединяемого изображения.

$$Dst(i, j) = \begin{cases} 0 & Dst(i, j) \leq 0 \\ Dst(i, j) & 0 < Dst(i, j) < 255 \\ 255 & Dst(i, j) \geq 255 \end{cases} \quad (12)$$

Из-за неправильного выбора опорного изображения результат сшивки панорам может быть слишком светлым, что затрудняет стыковку с общей панорамой. Чтобы решить эту проблему, возникающую при использовании цветовой коррекции на основе опорного изображения, предлагается вычислить корректирующие коэффициенты для каждого из цветовых каналов R, G и B с использованием алгоритма линейной коррекции цвета.

Алгоритм линейной цветовой коррекции представлен на рис. 6 и в качестве входных данных использует опорное изображение, выбранное на шаге 1 работы алгоритма коррекции цвета. При работе алгоритма линейной цветовой коррекции используется последовательность изображений с n камер и зон их перекрытия S_i ($i = 1, \dots, n$), схематичное изображение перекрытий приведено на рис. 7.

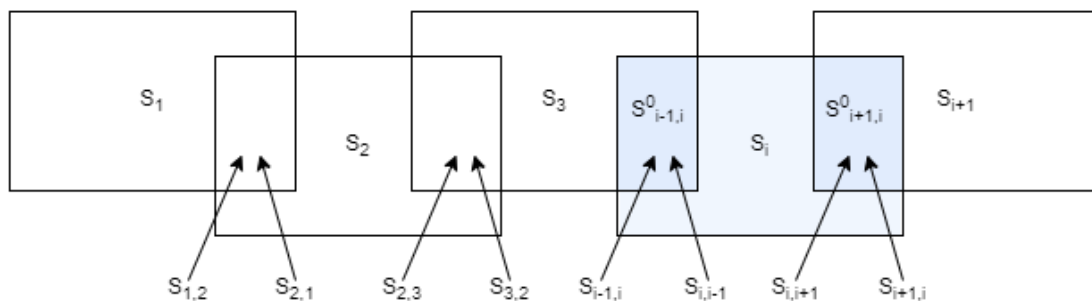


Рис. 7. Изображения с перекрывающимися областями в многокамерной системе

Fig. 7. Images with overlapping areas in a multi-camera system

Алгоритм линейной цветовой коррекции заключается в следующем.

Шаг 1. Обозначим перекрывающиеся области зрения камер относительно области зрения i -й камеры S_i ($i = 1, \dots, n$) с левой стороны – $S_{i,i-1}$, с правой – $S_{i,i+1}$. Обозначим $S_{i-1,i}^0$ и $S_{i,i-1}^0$ – области перекрытия, принадлежащие изображению S_i (рис. 7).

Для вычисления корректирующих коэффициентов $\alpha_R, \alpha_G, \alpha_B$ для каждого из каналов RGB используем линейную коррекцию, представленную в виде диагональной модели (13):

$$S_i = S_{i-1} * M, M = \begin{bmatrix} \alpha_R & 0 & 0 \\ 0 & \alpha_G & 0 \\ 0 & 0 & \alpha_B \end{bmatrix}, \quad (13)$$

где M – диагональная модель матрицы трансформации, $\alpha_R, \alpha_G, \alpha_B$ – коэффициенты цветовой коррекции R, G, B соответственно.

Вычисление корректирующих коэффициентов проводится следующим образом (14):

$$\alpha_c = \frac{\overline{C_{R,i-1}}}{\overline{C_{R,i}}}, \quad (14)$$

где $\overline{C_{c,i-1}}$, $\overline{C_{c,i}}$ – средние значения R, G, B для $S_{i-1,i}^0$ и $S_{i,i-1}^0$, $c \in \{R, G, B\}$, а α_c , $c \in \{R, G, B\}$ – среднее значение цвета в области пересечения полей зрения для каждого канала RGB.

Вычисляем локальный коэффициент коррекции $\alpha_{c,i}$ в линейном цветовом пространстве для изображения S_i по следующей формуле (15):

$$\alpha_{c,i} = \frac{\sum_p (C_{c,i-1}(p))^\gamma}{\sum_p (C_{c,i}(p))^\gamma}, \quad (15)$$

где $C_{c,i}(p)$ – значение цвета пикселя p цветового канала c изображения S_i^0 ; γ – гамма-коэффициент ($\gamma = 2,2$) [17]. Для эталонного изображения, относительно которого будут корректироваться последующие изображения, коэффициент коррекции устанавливаем равным 1.

Шаг 2. Чтобы избежать перенасыщения или потери контраста, необходимо минимизировать величину каждого коэффициента коррекции. Для этого используем квадратичную функцию с параметром g_c , что позволяет избежать перенасыщения или потери контраста (16):

$$E = \min_{g_c} \sum_{i=1}^n (g_c \alpha_{c,i} - 1)^2, \quad (16)$$

где g_c – коэффициент глобальной компенсации для цветового канала c ; n – количество изображений; $\alpha_{c,i}$ – коэффициент коррекции канала c для i -го изображения.

Для поиска экстремума приравняем ее производную к нулю, $\frac{\partial E}{\partial g_c} = 0$, в результате получим оптимальное значение коэффициента глобальной компенсации (17):

$$g_c = \frac{\sum_{i=1}^n \alpha_{c,i}}{\sum_{i=1}^n \alpha_{c,i}^2}. \quad (17)$$

Шаг 3. С помощью коэффициентов коррекции выполняем компенсацию цвета для всего изображения S_i . Для этого проведем изменение коэффициента коррекции в соответствии с выражением следующего вида (18):

$$(\alpha C_{c,i}^\gamma)^\frac{1}{\gamma} = \alpha^\frac{1}{\gamma} C_{c,i}. \quad (18)$$

Шаг 4. Применив далее эту идею для g_c и α_c , с помощью линейной коррекции цвета изменим значения каналов всех пикселей изображения на определенное значение. Таким образом, итоговый метод цветокоррекции отдельно корректирует значения каналов всех пикселей изображения на некоторую величину согласно выражению (19):

$$C_{c,i}(p) = (g_c \alpha_{c,i})^\frac{1}{\gamma} C_{c,i}(p). \quad (19)$$

В результате применения алгоритма линейной коррекции цвета при сшивке изображений в панораму цветовой переход становится более плавным, чем при цветовой коррекции с помощью алгоритма на основе опорного изображения. Однако из-за сильной разницы в цвете между изображениями применение только цветовой коррекции бывает недостаточно для полного избавления от шва между стыкуемыми изображениями, поэтому дополнительно, после проведения цветовой линейной коррекции, предлагается проводить градиентное смешивание.

Исследования в области градиентного смешивания по оптимальному шву показали, что алгоритм линейного градиентного смешивания дает видимый эффект размытия [10], алгоритмы Пуассоновского и многополосного смешивания являются затратными по времени и вычислительным мощностям [11]. Лучшее решение данной проблемы дает алгоритм градиентного смешения n -ой степени, который вычисляет значение цветового канала на линии смешения путем нахождения среднего значения двух соответствующих значений цветовых каналов пересекающихся пикселей на двух изображениях. При удалении от линии смешения весовой коэффициент одного изображения увеличивается до 1, в то время как другой коэффициент уменьшается до 0 согласно выражению [12]:

$$C'_I(p) = \frac{d_1^n C_I(p) + d_2^n C_S(p)}{d_1^n + d_2^n}, \quad (20)$$

где $C'_I(p)$ – новое значение цвета пикселя p в полученном панорамном изображении I ; $C_I(p)$ – цвет пикселя p в исходном изображении I ; $C_S(p)$ – цвет пикселя p в перекрывающем изображении S ; d_1 и d_2 – расстояния от границы смешения между двумя изображениями до границы области смешения.

Для проведения градиентного смешивания в работе используется метод *GraphCutSeamFinder* библиотеки компьютерного зрения *OpenCV*. Для определения оптимального пути сшивки выполняется поиск пути минимальной стоимости на графе, узлами которого являются пиксели области перекрытия двух изображений, применяемого для создания полностью плавного перехода [13]. Работа по созданию панорамного кадра завершается нахождением максимально вписываемого прямоугольника, по которому производится обрезка панорамы.

Заключение

Разработан и реализован метод сшивки изображений с видеопотоков в панораму, представляющий собой комбинацию алгоритмов калибровки систем камер, цветовой коррекции входных изображений и градиентного смешивания, повышающий скорость построения панорамы, сокращающий вычислительные затраты, повышающий точность калибровки камер за счет автоматизации процесса и участия оператора только на начальном этапе калибровки двух первых камер. Представленный метод построения панорамы позволяет строить эффективные многокамерные системы видеонаблюдения и аналитики, работающие в режиме реального времени.

Для реализации алгоритмов использовался язык *Python* с использованием инструментов библиотеки компьютерного зрения *OpenCV*. Создан экспериментальный стенд, включающий пять камер высокого разрешения *Tantos* – 1280x1960 пикселей, передающих цветное видеоизображение, монитор, одноплатный компьютер *Raspberry PI 4B*, вычислительный модуль *Intel neural stick 2*, видеорегистратор, аппаратная платформа *Arduino Mega* на базе микроконтроллера *ATmega 2560*. Проведенные эксперименты показали, что для одного кадра общее время, затрачиваемое на прохождение алгоритма, составляет около 35 миллисекунд.

Библиографический список

1. Багиров, М.Б. Обработка больших данных с целью создания системы определения геопозиции по фотографиям местности // О.П. Тимофеева, М.М. Гордеев, А.Н. Райцын, А.А. Дмитриев, А.Д. Тряскова, М.Б. Багиров // Информационные системы и технологии ИСТ-2020: сборник материалов XXVI Международной научно-технической конференции. Нижний Новгород: НГТУ им. Р.Е. Алексеева. 2020. С. 716-721.
2. Agarwala, A. Interactive digital photomontage / A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen // ACM Trans. Graph, vol. 23, pp. 294-302, 2004.

3. **Abbas, Q.** Reduced Solution Set Shortest Path Problem: Capton Algoritm With Special Reference To Dijkstra's Algorithm / Abbas, Q., Hussain, Q., Zia, T. & Mansoor, A. // Malaysian Journal of Computer Science, pp. 175-187.
4. **Жуковская, А.Н.** Алгоритм соединения двух изображений при создании панорамного изображения / А.Н. Жуковская, М.Н. Фаворская // Решетневские чтения. 2018. С. 264-266.
5. **Khamiyev, I.**, Issa, D., Akhtar, Z. et al. Panoramic image generation using deep neural networks. Soft Comput (2023). DOI: 10.1007/s00500-023-08056-5
6. **Dasari, R.** Reference Image Based Color Correction for Multi-camera Panoramic High Resolution Imaging / Dasari, R.; Zhang, D.-Q.; Chen, C.W // In Proceedings of the 2016 13th Conference on Computer and Robot Vision (CRV), Victoria, BC, Canada, 1 June 2016, pp. 410-415.
7. **Козин, Е.В.** Фотограмметрия СПб: Университет ИТМО / Е.В. Козин, А.Г. Карманов, Н.А. Карманова. 2019. С. 142.
8. **Шабалин, А.С.** Трансферное обучение на основе архитектуры нейронной сети EfficientNetV2B0 в задаче построения 3D маски лица человека / А.С. Шабалин, И.М. Елисеев // Ученые записки УлГУ. Сер. Математика и информационные технологии. 2022. № 2. С. 123-129.
9. **P. Lehel**, E.E. Nemayed and A.A. Farag. Sensor Planning for a Trinocular Active Vision System // In Proceedings of IEEE Computer Society Conference of Computer Vision and Pattern Recognition, Fort Collins, CO, 1999, vol. 2, pp. 306-312.
10. **V. Rankov**, In Three-Dimensional and Multidimensional Microscopy / V. Rankov, R.J. Locke, R.J. Edens, P.R. Barberand B. Vojnovic // Image Acquisition and Processing XII, vol. 5701, pp. 190-199, 2005.
11. **Пахирка, А.И.** Создание панорамных аэрофотоснимков с использованием квадрокоптера / А.И. Пахирка, А.Г. Зотин, В.В. Буряченко // Программные продукты и системы. 2018. С. 362-367.
12. **Rajaram, M. Gowda.** Development of Image Stitching Using Feature Detection and Feature Matching Techniques // 2020 IEEE International Conference for Innovation in Technology (INOCON), pp. 1-7.
13. **V. Kwatra.** Graphcut textures: image and video synthesis using graph cuts / Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick // In ACM Transactions on Graphics (ToG), volume 22, pp. 277-286. ACM, 2003.

*Дата поступления
в редакцию: 06.07.2023*

*Дата принятия
к публикации: 01.11.2023*