

Идентификация линий по некалиброванным видам городской обстановки¹

В.А. Бобков, Ю.И. Роньшин, А.П. Кудряшов

Аннотация. Предложен метод сопоставления линий по трем некалиброванным изображениям (видам) сцен городской обстановки. Описаны методика и алгоритмы получения полной калибровки камер по изображениям с учетом присущей архитектурным сценам параллельности/ортогональности линий. Получены оценки эффективности предложенного подхода для изображений модельных и реальных сцен.

Введение

Настоящая работа является развитием ранее опубликованной авторами работы [1], где решалась задача сопоставления прямолинейных отрезков для калиброванных изображений с применением трифокальной геометрии и методики корреляционного сравнения пиксельных образов с целью последующей пространственной реконструкции сцены. Рассматриваются сцены городской обстановки, на векторизованных перспективных видах которых идентифицируются отрезки-ребра зданий. Поскольку предположение об априори известной полной калибровке камер (т.е. положений, ориентации и фокусов камер) ограничивает возможности практического применения разработанных средств в решении конечной задачи - реконструкции сцены по фотоизображениям, в настоящей статье делается акцент на получении полной калибровки исходных изображений.

В последние годы исследователями в области компьютерного зрения много внимания было уделено решению задачи калибровки в контексте проблемы реконструкции простран-

ственных сцен по некалиброванным изображениям. С обзором по этой теме можно ознакомиться, например, в [2]. Основопологающей концепцией в алгоритмах self-calibration является использование инварианта absolute conic. Учет дополнительных ограничений, связанных со спецификой сцен, позволяет упростить конструирование алгоритмов вычисляющих absolute conic и, соответственно, матрицы калибровки камеры. Например, в работах [3,4] существенно используется параллельность и ортогональность линий, присущая архитектурным сценам. В этом случае решения строятся на использовании т.н. vanishing points (точек схода), отражающих перспективные искажения семейств параллельных линий в плоскости изображения. В указанных работах решаются задачи вычисления аффинных измерений по одному виду, вычисления матрицы калибровки на основе absolute conic с использованием vanishing points и реконструкция отдельного здания по двум видам.

Предлагаемые в данной работе методика и алгоритмы калибровки также построены на использовании параллельности/ортогональности

¹ Работа выполнена при финансовой поддержке Президиума РАН (Программа фундаментальных исследований №14)

линий-ребер зданий с минимальным интерактивным привлечением оператора, обеспечивающего ввод дополнительной геометрической информации. Однако в отличие от вышеупомянутых работ, где рассматриваются локальные сцены с реконструкцией отдельных граней, предлагаемый подход ориентирован на обработку протяженных сцен и основывается на идее идентификации и восстановления ребер зданий с последующим построением полигональной модели сцены. Предлагается оригинальный метод определения полной калибровки камер и матриц преобразования, связывающих мировую систему координат сцены с системой координат камеры. Эффективность ранее предложенного авторами «трифокального алгоритма» сопоставления линий повышена, главным образом, за счет выявления и обработки ситуаций, когда на одном из трех изображений отсутствует образ наблюдаемого отрезка. Для оценки эффективности предложенной методики используются как модельные, так и реальные сцены. В целом разработанная методика, отталкиваясь от известных идей, в том числе, изложенных в работах [3-5], базируется на оригинальных алгоритмах, особенностью которых является попытка

минимизировать влияние на конечный результат ошибок геометрии, неизбежно присущих векторизации исходных изображений.

1. Определение калибровки камер

В качестве исходных данных рассматривается последовательность снимков (видов) городской сцены, представленной множеством зданий. Снимки сделаны с перекрытием, таким образом, чтобы для каждой пары соседних изображений нашлась одна точка-особенность (угол здания, окна), видимая на обоих изображениях. Каждый вид отвекторизован (автоматически или с привлечением оператора), т.е. ребра стен зданий и окон представлены отрезками прямых. Отрезки идентифицируются номерами. Требуется найти матрицы преобразований из некоторой единой евклидовой мировой системы координат (МСК) в системы координат камер (СКК).

Возможны три типа входных данных и, соответственно – три режима работы:

- а) фокус (фокусное расстояние) известен для каждого вида;
- б) фокус – одинаковый для всех видов, но неизвестный (при съемке zoom не используется, автофокусировка выключена);
- в) для каждого вида – свой неизвестный фокус.

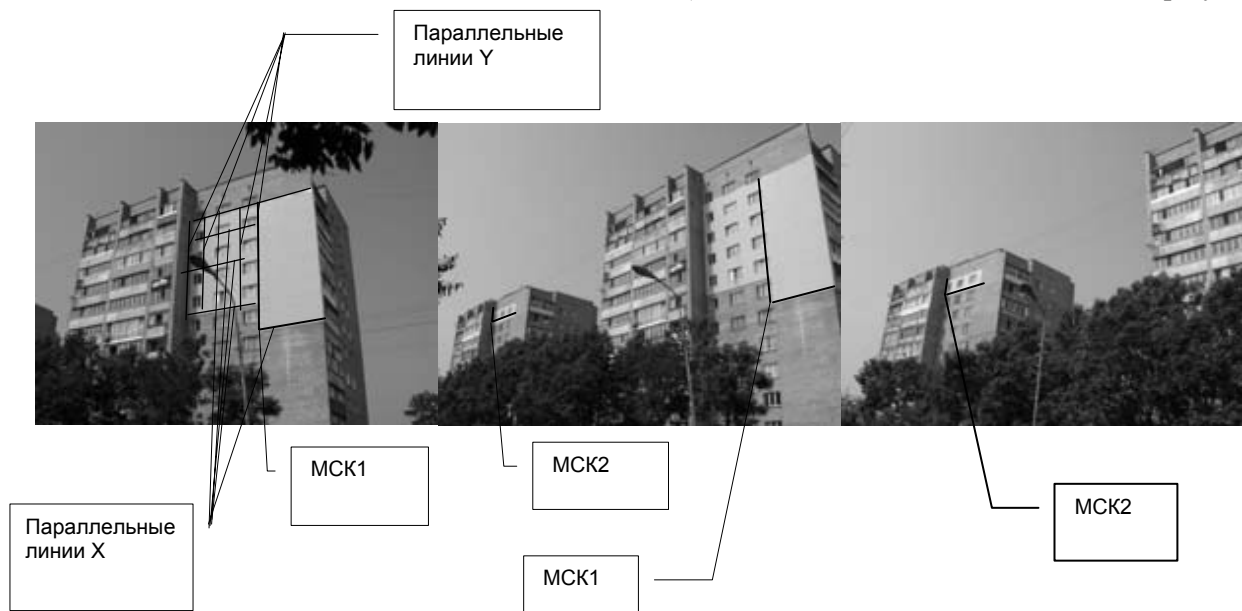


Рис. 1. Точки-особенности, задающие начала локальных МСК, и семейства параллельных линий на исходных изображениях, определяющие *vanishing points*

Если фокус f известен, он задаётся. При неизвестных фокусах требуется также найти фокусы для каждого вида. Далее будем рассматривать случай (с), как наиболее общий.

Решение. Решение ищем для тройки соседних видов. Привяжем единую МСК к некоторому углу (здания, окна) на снимке 1. Плоскость стены здания (*reference plane*), содержащей этот угол, является плоскостью XY этой МСК. Одно из рёбер этой стены соответствует оси X. Если для некоторой тройки выполняется оговоренное выше условие: угол (МСК1) виден на снимках 1 и 2, но не виден на снимке 3, то для снимка 2 определяется дополнительно матрица преобразования из другого угла (МСК2), видимого также на снимке 3 (Рис.1). Матрица преобразования из МСК1 в СКК3 вычисляется по формуле:

$$H_{13} = H_{12}H_{22}^{-1}H_{23}$$

где H_{12} – преобразование из МСК1 в СКК2,

H_{22} – преобразование из МСК2 в СКК2,

H_{23} – преобразование из МСК2 в СКК3.

Матрицы зависят от фокусов. При известных фокусах матрицы определяются независимо для каждого вида (п. 1.1). При неизвестных фокусах для каждой тройки соседних видов решается задача нелинейной оптимизации (раздел «Определение фокусов и матриц преобразования для тройки видов»). После того как фокусы получены, задача решается так же, как для режима (а).

1.1. Определение матрицы преобразования при известном фокусе

Базовая геометрия одного вида, которая определяет геометрическую взаимосвязь перспективной камеры и наблюдаемой в пространстве плоскости (стена здания) через точки *vanishing points*, представлена на Рис. 2. Для определения матриц преобразования для каждого вида интерактивно задаются:

- два семейства параллельных линий, принадлежащих *reference plane* (стена здания) – $\{l_j\}_1, \{l_j\}_2$ (Рис.1);

отрезок l_{01} на изображении соответствует задаваемому оператором отрезку на оси X системы координат МСК1;

d - длина задаваемого отрезка на оси X (метрика на местности).

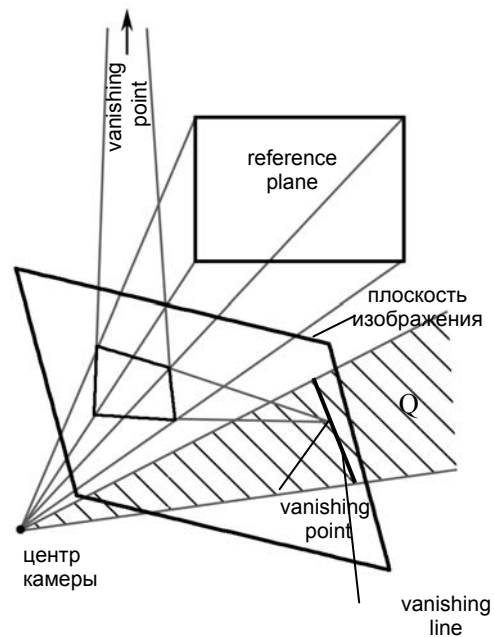


Рис. 2. Базовая геометрия одного вида

Vanishing line образуется пересечением плоскости изображения и плоскости Q, проходящей через центр камеры и параллельной *reference plane*. Vanishing point образуется пересечением в плоскости изображения проекции семейства параллельных линий, принадлежащих *reference plane*. На рисунке показаны проекции двух взаимно перпендикулярных пар параллельных линий на *reference plane*, образующих в плоскости изображения две vanishing points, которые лежат на vanishing line.

В режимах (b) и (с) для каждого вида задаются:

- множества точек-особенностей – $\{p_k\}$ (идентифицируются номерами);
- для каждой точки соответствующие ей точки на других видах – $\{p_l\}_k$,

где p_l – пара – $\{\langle \text{номер вида} \rangle, \langle \text{номер точки} \rangle\}$.

Требуется определить H – матрицу преобразования из МСК в СКК.

Алгоритм:

1. Определяется n – вектор нормали *reference plane* в СКК.

1.1. Вычисляем vp_1 – *vanishing point* для $\{l_j\}_1$.

Из-за ошибок векторизации параллельные линии на изображении в общем случае не будут пересекаться в одной точке. Для более корректного устранения этих ошибок применим алгоритм, предложенный в [4]. В соответствии с ним выполняется оптимизация функции двух переменных – x_1, u_1 – (координат vp_1). Начальное приближение

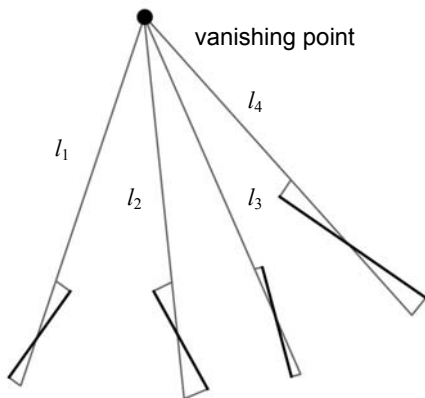


Рис. 3. Vanishing point определяется как пересечение линий l_i , которые минимизируют расстояния от конечных точек образов параллельных отрезков до указанных линий

– среднее от точек пересечения всевозможных пар отрезков из $\{l_j\}_1$. Целевая функция – сумма расстояний от концов отрезков l_j до линии, определённой точками vp_1 и серединой отрезка l_j (Рис3).

1.2. Аналогично вычисляем vp_2 – *vanishing point* для $\{l_j\}_2$.

1.3. Тогда \mathbf{n} определяется как вектор нормали плоскости, заданной 3D-точками $(0,0,0)$, (x_1, y_1, f) , (x_2, y_2, f) , т.е. плоскости Q (Рис.2). Здесь мы используем факт параллельности плоскостей *reference plane* и Q.

2. Вычисление матрицы H.

2.1. Рассмотрим отрезок $\mathbf{p1p2}$ вдоль оси X выбранной МСК, где $\mathbf{p1}$ и $\mathbf{p2}$ – начало и конец ребра, а $\mathbf{p1}$ является началом МСК. Определим концевые 3D-точки отрезка как (Рис.4):

$$\mathbf{p1} = t_1 \mathbf{v1}; \quad \mathbf{p2} = t_2 \mathbf{v2},$$

где $\mathbf{v1}$ – 3D-точка начала отрезка l на экране с координатами (beg_lx, beg_ly, f) ,

$\mathbf{v2}$ – 3D-точка конца отрезка l на экране с координатами (end_lx, end_ly, f) .

Теперь получим выражения для $t1$ и $t2$ (привязка к системе координат камеры).

Пусть:

\mathbf{n} – вектор нормали плоскости,

$\mathbf{v1}$ – точка 1 на экране (в 3D),

$\mathbf{v2}$ – точка 2 на экране (в 3D),

L – длина на местности между точками $p1$ и $p2$.

Для всех точек плоскости выполняется:

$(\mathbf{n}, \mathbf{p1}) = h$, где h – расстояние до плоскости;

$(\mathbf{n}, \mathbf{p2}) = h$.

Пусть $\mathbf{p1} = t1 * \mathbf{v1}$ и $\mathbf{p2} = t2 * \mathbf{v2}$, а $\|\mathbf{p2} - \mathbf{p1}\| = L$.

Тогда

$$t2 = (\mathbf{n}, \mathbf{v1}) * t1 / (\mathbf{n}, \mathbf{v2}); \quad |t1| = L |(\mathbf{n}, \mathbf{v2})| / \|(\mathbf{n}, \mathbf{v1})\mathbf{v2} - (\mathbf{n}, \mathbf{v2})\mathbf{v1}\|$$

2.2. Определим МСК в СКК:

Начало – точка $\mathbf{p1}$,

Орт X (\mathbf{Ox}) – вектор $(\mathbf{p2} - \mathbf{p1}) / \|\mathbf{p2} - \mathbf{p1}\|$,

Орт Z (\mathbf{Oz}) – вектор \mathbf{n} ,

Орт Y (\mathbf{Oy}) – достраивается.

2.3. Для этого задания строится матрица H.

$$H = \begin{pmatrix} \mathbf{Ox}, x & \mathbf{Ox}, y & \mathbf{Ox}, z & 0 \\ \mathbf{Oy}, x & \mathbf{Oy}, y & \mathbf{Oy}, z & 0 \\ \mathbf{Oz}, x & \mathbf{Oz}, y & \mathbf{Oz}, z & 0 \\ \mathbf{p1}, x & \mathbf{p1}, y & \mathbf{p1}, z & 1 \end{pmatrix}$$

Элементами матрицы H являются компоненты соответствующих векторов. Например, \mathbf{Ox}, y – компонента y вектора \mathbf{Ox} .

1.2. Определение фокусов и матриц преобразования для тройки видов

Если семейства $\{l_j\}_1, \{l_j\}_2$ представляют взаимно ортогональные линии, то несложно показать, что фокус можно вычислить по формуле:

$$f = \text{sqrt}(-(\mathbf{vp}_1, \mathbf{vp}_2)) \quad (1)$$

Вывод (Рис.5):

$(\mathbf{v1}, \mathbf{v2}) = 0$; $\mathbf{vp}_1 = (x_1, y_1)$ и $\mathbf{vp}_2 = (x_2, y_2)$ – лежат в плоскости экрана.

$\mathbf{v1} = (x_1, y_1, f)$ и $\mathbf{v2} = (x_2, y_2, f)$ т.к. \mathbf{f} – направлен по оси z СКК.

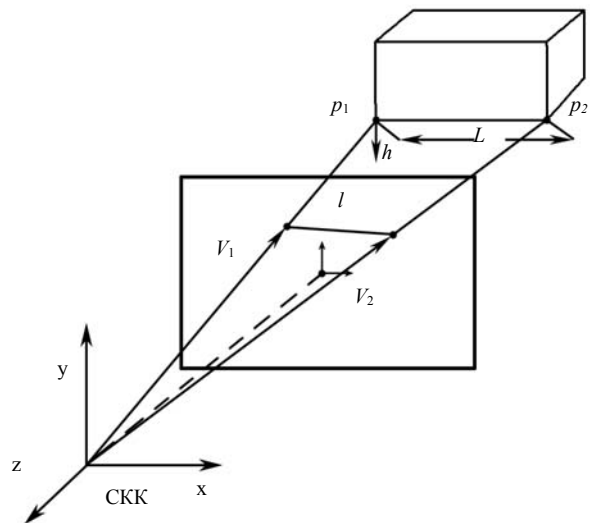


Рис. 4. Вычисление орта X МСК

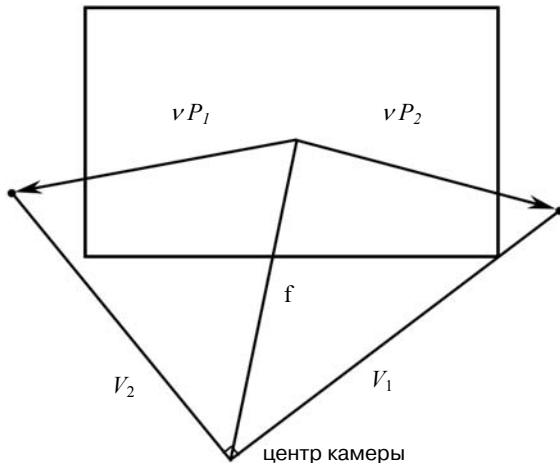


Рис. 5. Вычисление фокуса по двум взаимно перпендикулярным семействам параллельных линий

Тогда $(v_1, v_2) = (vp_1, vp_2) + f * f$, и, соответственно, $f = \sqrt{-(vp_1, vp_2)}$.

Таким образом, фокусы зависят только от *vanishing points*. Задача их нахождения решается оптимизацией функции, в которой переменными являются координаты *vanishing points* для каждого вида. Начальные приближения – результаты п.п. 1.1, 1.2 раздела «Определение матрицы преобразования при заданном фокусе». Целевая функция вычисляется следующим образом. Для очередного приближения *vanishing points*:

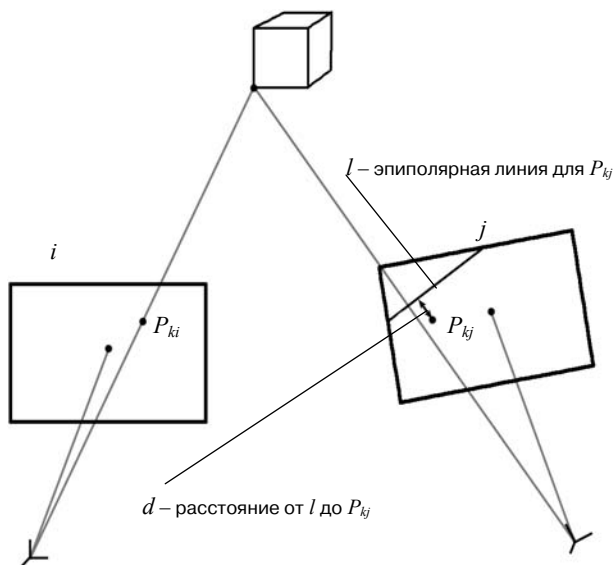


Рис. 6. Вычисление суммарной репроекционной ошибки для заданных контрольных точек

1. Для каждого вида вычисляются фокусы f_i по формуле (1) и определяются матрицы H_i (f_i) по алгоритму, описанному в разделе 1.1.

2. Для каждого вида i и для каждой точки из $\{r_k\}$, если есть соответствующая точка на виде j , по матрицам H_i, H_j вычисляется фундаментальная матрица F_{ij} и на виде j строится эпиполярная линия, соответствующая точке r_k (Рис.6).

3. Вычисляется расстояние от точки, соответствующей точке r_k , до эпиполярной линии.

4. Значение целевой функции равно сумме этих расстояний.

Таким образом, для найденных в результате оптимизации *vanishing points* получаем окончательную полную калибровку камер и искомые матрицы преобразования.

2. Идентификация линий по трем видам

На втором этапе выполняется сопоставление отрезков по трем видам для калиброванных камер (полученных на первом этапе). Сопоставление осуществляется с помощью «трифокального» алгоритма, подробное описание начальной версии которого приведено в [1]. Алгоритм основывается на построении интегральной оценки сходства отрезков, учитывающей их геометрическую и текстурную близость. Геометрическая близость оценивается с помощью трифокального тензора, а текстурная – применением кросс-корреляции с учетом гомографии. Применение пороговой фильтрации по нескольким критериям и ранжировка кандидатов по интегральным оценкам дает решение, которое, в принципе, носит вероятностный характер. С учетом недостатков, присущих первой версии алгоритма, в настоящей работе была выполнена модификация алгоритма, направленная на повышение числа идентифицированных линий и уменьшение количества ошибочно сопоставленных отрезков. В частности, выполненная модификация позволила обрабатывать ситуации, когда на одном из видов отсутствует образ наблюдаемого отрезка. Рассмотрим две основные доработки, которые позволили повысить эффективность алгоритма в целом.

Первая связана с учетом смежности (связности) отрезков, сохраняющейся на каждом из видов. Отрезки считаются смежными, если

имеют общую концевую точку. После построения интегральных оценок для каждого отрезка первого вида сохраняется ранжированный массив троек i,j,k – кандидатов на верное решение по сопоставлению отрезков. Здесь i – номер отрезка на первом виде, j,k – номера, соответственно, на втором и третьем виде. Обозначим через in смежный отрезок с i на первом виде, соответственно через jn смежный отрезок с j на втором виде и через kn смежный отрезок с k на третьем виде. Также введем обозначение \rightarrow для указания сходства отрезков на двух видах и обозначение \approx для указания смежности отрезков одного вида. Тогда для правильно определенной тройки i,j,k справедливо:

если $i \approx jn \rightarrow jn$ и $i \rightarrow j \approx jn^*$, то $jn = jn^*$, что, собственно, отражает тот факт, что образы смежных отрезков должны быть тоже смежными. Аналогично с учетом третьего вида:

если $i \approx jn \rightarrow kn$ и $i \rightarrow k \approx kn^*$, то $kn = kn^*$.

При успешном выполнении указанных проверок найденная тройка считается верной.

Вторая доработка заключается в реализации так называемого обратного прохода для отбраковки неправильных троек. В основном она помогает обрабатывать ситуации, когда образ наблюдаемого отрезка отсутствует на втором/третьем виде. Обратный проход обеспечи-

вается циклической перестановкой видов: $(1,2,3)$, $(2,3,1)$, $(3,1,2)$. Для каждой из указанных последовательностей видов полностью отрабатывается алгоритм сопоставления, давая в итоге найденную тройку i,j,k . Для правильно сопоставленной тройки во всех случаях должен получиться один и тот же результат.

3. Результаты экспериментов

Для оценки эффективности предложенного подхода были проведены две серии вычислительных экспериментов. В первой рассматривалась модельная сцена городской обстановки, в которой здания генерировались средствами воксельной графики по реальным данным кадастрового плана с нанесением искусственной текстуры. Входной информацией для работы метода в целом (калибровка плюс сопоставление линий) являлись векторизованные растровые виды, полученные программным образом. На Рис. 7 представлены растровые изображения (виды), а на Рис. 8 соответствующие им векторные изображения, полученные после векторизации. Чтобы оценить эффективность этапа калибровки, вначале был проведен эксперимент с использованием известной калибровки (той, которая устанавливалась при программном

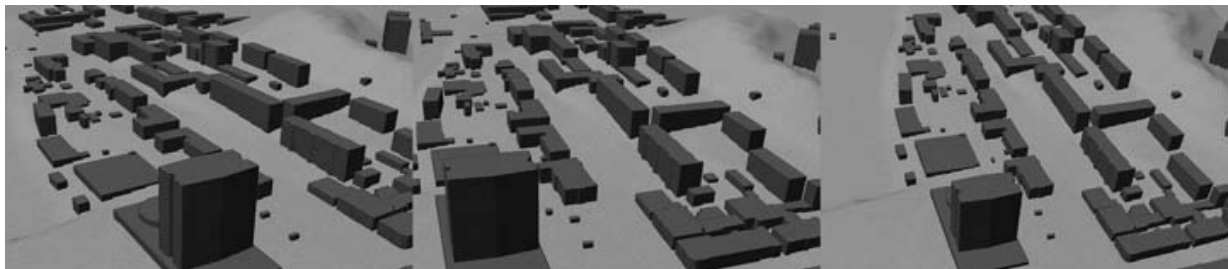


Рис. 7. Виды модельной городской сцены.

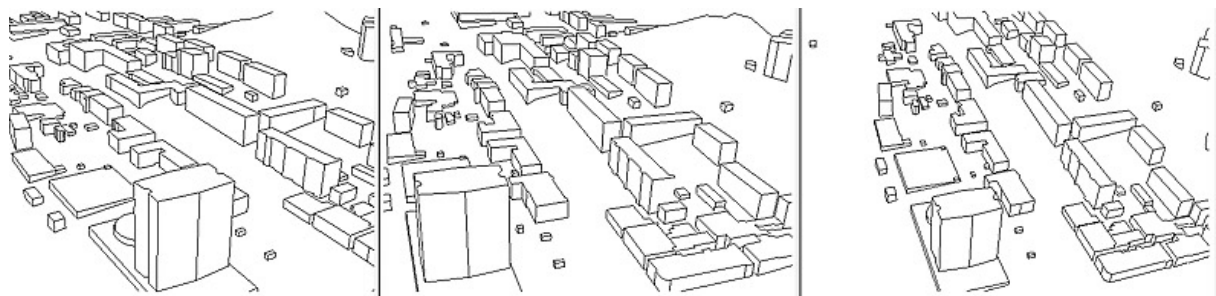


Рис. 8. Векторизованные виды городской сцены

получении видов – для приведенных видов фокусное расстояние задавалось равным 50 см). Во втором эксперименте калибровка считалась неизвестной и вычислялась описанным выше методом, а «трифокальный» алгоритм работал уже с вычисленным фокусом и вычисленными матрицами преобразования. Это позволило получить косвенную оценку эффективности калибровки за счет сравнения полученных результатов сопоставления линий в первом и во втором случае. Результат работы «трифокального» алгоритма в первом эксперименте приведен в первой строке Табл. 1., а во втором эксперименте – во второй строке. В таблице приведена также интегральная количественная оценка эффективности сопоставления линий, учитывающая как количество сопоставленных алгоритмом отрезков, так и количество ошибочных решений. Оценка вычислялась по формуле: $E = (n-w)(m-w)/nm$, где n – число потенциальных троек (определяется оператором), m – число определенных троек, w – число неправильных троек. Как видно из таблицы, ошибка в вычислении калибровки несколько ухудшила результат сопоставления – количество сопоставленных уменьшилось на 28 отрезков, а количество ошибочно сопоставленных увеличилось на 14. Погрешность в калибровке можно оценить и по полученному значению фокуса для каждого вида: $f_1 = 50.4172$, $f_2 = 50.3573$, $f_3 = 49.5906$ (правильное значение 50 см). Следует также отметить, что анализ ошибочных решений в первом эксперименте пока-

зал, что в половине случаев из 22 ошибка сопоставления связана с явной ошибкой векторизации или с совпадением образа искомого отрезка на втором и третьем виде с другим отрезком.

Во второй серии экспериментов использовалась реальная сцена, для которой были получены три некалиброванные изображения (Рис. 9). Эксперименты проводились для разного числа контрольных точек p_k (п. 1.2.), используемых на этапе калибровки для формирования целевой функции, оценивающей суммарную репроекционную ошибку. В принципе, результаты экспериментов подтвердили предположение о том, что увеличение количества контрольных точек повысит точность калибровки. Вместе с тем, как видно из таблицы, эксперимент с 9-ю контрольными точками дал результат лучше, чем с 12 и 25 точками, что, по-видимому, можно объяснить удачной случайной подборкой контрольных точек, приведшей к «хорошо» вычисленной фундаментальной матрице.

Параллельные вычисления. Оценка вычислительных затрат в проведенных экспериментах показала, что наиболее трудоемкой частью всей вычислительной схемы является работа алгоритма по сопоставлению отрезков на трех видах. Например, для приведенной модельной сцены это время порядка 3-х минут. Учитывая, что для реальных протяженных сцен трудоемкость будет выше и что в конечной версии системы предполагается обработка «длинной» последовательности изображений (видов),

Табл. 1. Результаты вычислительных экспериментов для модельной и реальной сцены

Сцена	Число контрольных точек на этапе калибровки	Всего отрезков в сцене	Количество сопоставленных отрезков	Количество ошибочно сопоставленных отрезков	Эффективность
Модельная сцена с известной калибровкой		522	490	22	0.856
Модельная сцена с неизвестной калибровкой	7	522	462	36	0.752
Реальная сцена	6	94	70	5	0.642
Реальная сцена	9	94	90	0	0.957
Реальная сцена	12	94	86	3	0.852
Реальная сцена	25	94	84	1	0.872



Рис. 9. Виды реальной городской сцены

целесообразно повысить эффективность обработки за счет организации параллельных вычислений. Анализ алгоритма сопоставления показывает, что сравнительно легко применить принцип распараллеливания «по данным» к первой его части, связанной с однотипной обработкой для каждого отрезка первого вида – вычислением описанных выше потенциальных троек-кандидатов. При этом вычисления смежности отрезков и «обратных проходов» будут выполняться последовательно. Оценку эффективности можно получить следующим образом. Пусть T_1 будет время выполнения распараллеливаемого участка вычислений, q – доля накладных расходов, n – количество процессоров, а T_2 – время выполнения нераспараллеливаемой части алгоритма. Тогда время выполнения в последовательном варианте будет $T_{\text{послед}} = T_1 + T_2$, а в параллельном $T_{\text{паралл}} = T_1(1-q)/n + qT_1 + T_2$. Эффективность определим как $T/nT_{\text{паралл}}$. При таком варианте распараллеливания реализуемая эффективность получилась сравнительно небольшой, поскольку T_1 в рассматриваемом алгоритме составляет примерно половину от общего времени его работы.

Заключение

В работе представлен метод идентификации (сопоставления) линий зданий по трем векторизованным исходным некалиброванным фотоизображениям городской сцены. Внешняя и внутренняя калибровка камер, а также привязка всех видов к единой евклидовой мировой системе координат выполняется с помощью алгоритмов, использующих параллельность/ортогональность ребер зданий и оптимизацию репроекционных ошибок, с минимальным привлечением интерактивной работы оператора.

В настоящей версии программы точки-особенности, используемые на этапе оптимизации для уточнения vanishing points, задаются оператором. Планируемое в дальнейшем алгоритмическое определение точек-особенностей позволит упростить методику работы с программой и повысить точность калибровки. Сопоставление линий осуществляется оригинальным алгоритмом, построенным с применением эпиполярных ограничений (трифокальный тензор) и методики корреляционного сравнения растровых образов отрезков. Проведенные вычислительные эксперименты на модельных и реальных сценах позволяют говорить об эффективности и достаточной устойчивости предложенной методики к неизбежным ошибкам векторизации исходных изображений, а также о перспективности подхода для целей пространственной реконструкции городских сцен по изображениям. Дальнейшее развитие работы предполагает решение задачи полной реконструкции протяженной городской сцены по ограниченной последовательности некалиброванных изображений с использованием полученных результатов.

Литература

1. В.А. Бобков, Ю.И. Роньшин, А.П. Кудряшов. Сопоставление линий по трём видам пространственной сцены // Информационные технологии и вычислительные системы, №2, 2006, С. 71-78.
2. A. Fusiello. Uncalibrated Euclidean Reconstruction: A Review. IVC, 18(6-7):555–563, 2000.
3. A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. IJCV, 40(2); pages 123-148, 2000.
4. D. Leibowitz, A. Criminisi, and a. Zisserman. Creating architectural models from images. In Proceedings EUROGRAPHICS, pp. 39-50, 1999.
5. Schmid, C. and Zisserman, A., 1997. Automatic line matching across views. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 666–671.

Бобков Валерий Александрович. Родился в 1947 г. Окончил Московский физико-технический институт в 1971 году. Доктор технических наук. Автор более 60 научных работ. Область научных интересов: компьютерная графика, геоинформатика, обработка изображений, компьютерное зрение. Заведующий лабораторией машинной графики в Институте автоматизации и процессов управления ДВО РАН.

Роньшин Юрий Иванович. Родился в 1953 г. Окончил Ленинградский государственный университет в 1977 году. Автор около 20 научных работ. Специалист в области компьютерной графики и ее приложений, системного программирования, параллельных вычислительных систем. Ведущий инженер – программист Института автоматизации и процессов управления ДВО РАН.

Кудряшов Алексей Павлович. Родился в 1981 г. Окончил в 2004 году Дальневосточный технический университет г. Владивосток). Автор 2 научных работ. Область научных интересов: компьютерная графика, обработка изображений, компьютерное зрение. Аспирант, инженер - программист Института автоматизации и процессов управления ДВО РАН.