

В.Ю. Сдобников, Б.Д. Савчинский

## Оценивание положения точек схода на изображениях городских кварталов

Описано вычисление положения точек пересечения проекций параллельных линий на плоскости изображения, полученного с помощью фотокамеры, что позволяет оценить поворот и внутренние параметры камеры, избавиться от перспективных искажений и решать многие другие задачи. Предложена постановка задачи нахождения точек схода как задачи самообучения и приведен эффективный алгоритм ее решения.

The vanishing points detection on the city block images is described. Vanishing points have a number of possible usages, some of which are camera parameters estimation and images rectification, thus their detection is an important computer vision problem. The novelty of the paper is the formalization of the vanishing points detection problem as the problem of unsupervised learning. An effective algorithm of its solution is suggested which is based on the well-known EM algorithm.

Обчислено положення точок перетину проєкцій паралельних прямих на площині зображення, отриманого за допомогою фотокамери, що дозволяє оцінити поворот та внутрішні параметри камери, позбутись перспективних спотворень тощо. Запропоновано постановку задачі обчислення точок сходу як задачі самонавчання та наведено ефективний алгоритм її розв'язання.

**Введение.** Прямые, параллельные в трехмерном пространстве перестают быть параллельными при их центральном проектировании на плоскость изображения (как это происходит в традиционных фотокамерах [1]) и пересекаются в одной точке. Точку пересечения этих прямых принято называть точкой схода.

Знание точек схода и соответствующих им направлений в пространстве позволяет решать такие прикладные задачи компьютерного зрения, как:

- оценка поворота и внутренних параметров камеры [1] по единственному изображению;
- ректификация изображения, избавление от перспективных искажений [1];
- анализ положения объектов в пространстве;
- частичная трехмерная реконструкция сцены по единственному снимку и другие.

Поскольку задача нахождения точек схода важна в компьютерном зрении и давно исследуется, существует значительное количество подходов к ее решению [2–9].

Большинство известных алгоритмов состоят из трех стадий: выделение прямых (или их фрагментов) на изображении, разбиение выделенных прямых на классы, соответствующие точкам схода, и оценка точек схода по прямым (их фрагментам), попавшим в каждый из клас-

сов. Для выделения прямых, как правило, используются преобразование Хафа [10] или детектором краев Кэнни [11]. Выделенные прямые разбиваются на классы, к каждому из которых относятся прямые, параллельные в пространстве (и, соответственно, задающие точку схода на изображении). Затем по прямым, попавшим в каждый из классов, оценивают точки схода. При оценке пользуются различными критериями, наиболее распространенный из которых – минимизация суммы квадратов отклонений прямых из класса от оцениваемой точки.

При таком подходе вторая и третья стадии связаны: разбиение на классы зависит от расположения точек схода, а оценки для точек схода зависят от разбиения выделенных прямых на классы. Для преодоления этой трудности в существующих методах для разбиения прямых на классы применяют алгоритм *RANSAC* [12], результатом которого является принадлежность каждой из выделенных прямых к одному из классов (горизонтальные и вертикальные прямые на рис. 1). Затем оценивают точки схода.

Алгоритм *RANSAC* для разделения выделенных прямых на классы выглядит следующим образом:

1. задается порог  $t$  и число итераций  $N$ , повторяется  $N$  раз следующая процедура:

а) случайным образом из выделенных на изображении прямых выбираются две различные, подсчитывается точка их пересечения  $P$ ;

б) подсчитывается количество прямых  $k$ , проходящих ближе порога  $t$  к найденной точке пересечения  $P$ ;

2. Выбирается такая точка  $P^*$ , для которой число  $k$  максимально. Считается, что к классу относятся те прямые, которые проходят ближе порога  $t$  к точке  $P^*$  (горизонтальные прямые на рис. 1).

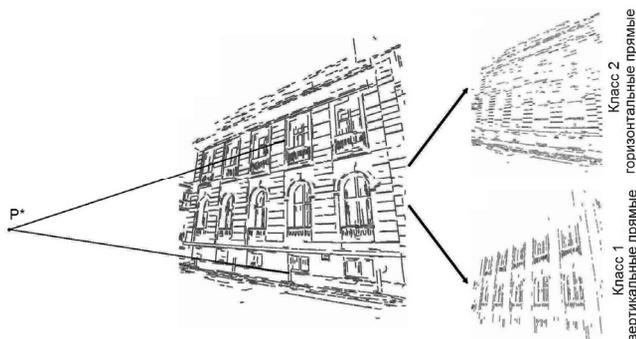


Рис. 1. Разбиение выделенных на изображении прямых на классы с помощью алгоритма RANSAC

Для разделения на несколько классов  $m$  описанный алгоритм запускается  $m$  раз, каждый последующий раз на прямых, не попавших ни в один из классов на предыдущих запусках. Все оставшиеся после  $m$  запусков прямые считаются выбросами и не влияют на оценку точек схода. Таким образом, этот подход позволяет отсеивать шумовые прямые в том случае, когда их часть в выделенных прямых невелика.

Основным недостатком описанного подхода есть то, что разделение на классы и оценка точек схода происходят не одновременно, что приводит к ошибкам (рис. 2). Пример иллюстрирует две пары фрагментов прямых, выбранных на двух разных итерациях стадии 1,а алгоритма RANSAC, и соответствующие этим парам классы прямых. Результатом работы алгоритма будут прямые, проходящие возле точки пересечения синих фрагментов (рис. 2 внизу справа), так как их больше (хотя они и не параллельны в пространстве) чем тех, которые проходят близко к точке пересечения красных

фрагментов (и являются параллельными, вертикальными в пространстве).

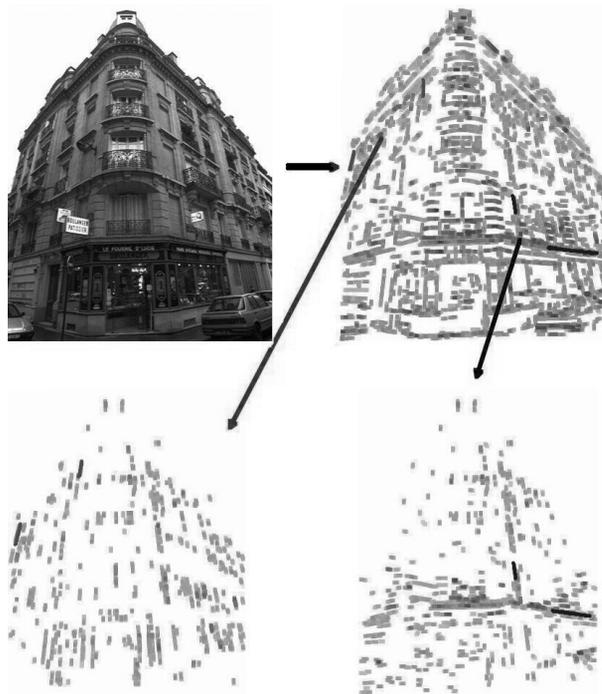


Рис. 2. Ошибка при разбиении выделенных на изображении прямых на классы с помощью алгоритма RANSAC

Итак, предлагается подход, лишенный этого недостатка, где разделение выделенных прямых на классы и поиск точек схода происходит одновременно, исключая возможность возникновения описанной ошибки разбиения. Входными данными для алгоритма самообучения [13] является выборка прямых, выделенных на изображении с помощью преобразования Хафа [10]. Задача состоит в разделении представленной выборки на классы, каждый из которых соответствует подмножеству параллельных в трехмерном пространстве прямых, и оценке координат точек пересечения (точек схода) прямых, входящих в каждый из классов. Предложенный метод учитывает шумовые прямые, выделенные на изображении (кроме прямых, выделяемых на плоскости фасадов зданий и соответствующих точкам схода, существуют прямые, не соответствующие никаким из точек схода – границы деревьев, машин, провода, люди, соседние здания и т.д.). Также разделение происходит на все классы одновременно, что значительно умень-

шает вероятность заведомо неправильного отнесения прямых одного класса к другому.

### **Постановка задачи поиска точек схода как задачи самообучения**

#### **Выделение прямых на изображении**

Первый этап решения задачи заключается в выделении на входном изображении множества прямых. Для этого используется преобразование Хафа [10], что позволяет находить параметрически заданные кривые на изображении, в частности прямые. Прямая

$$x \cdot \cos \theta + y \cdot \sin \theta - \rho = 0 \quad (1)$$

на плоскости изображения задается углом наклона  $\theta$  и расстоянием  $\rho$  от начала координат до прямой.

Мы использовали модификацию классического преобразования Хафа, описанную в [14]. В этой модификации считается, что каждая точка изображения лежит лишь на одной, перпендикулярной направлению градиента яркости изображения в данной точке, прямой. Также каждая точка учитывается с весом, равным модулю градиента яркости изображения в ней. Результат преобразования Хафа – множество точек в пространстве параметров  $(\theta, \rho)$ , каждая из которых соответствует выделенной на изображении прямой. В такой модификации алгоритма каждой точке пространства параметров приписано значение ее интенсивности, пропорциональное количеству точек на изображении, лежащих на соответствующей прямой. Для оценки положения точек схода используется только определенное, наперед заданное количество точек пространства параметров (прямых на изображении) с наибольшими значениями интенсивностей. Множество прямых, соответствующих таким наиболее интенсивным точкам, обозначим  $\{\omega_i = (\theta_i, \rho_i) \mid i = 1, \dots, n\}$ .

Считаем, что каждая из выделенных на изображении прямых лежит в одном из  $m$  классов. Множество классов обозначим  $K = \{1, \dots, m\}$ . Каждый из классов  $k = 1, \dots, m-1$ , содержит прямые, отвечающие одной, определенной точке схода. В классе  $m$  лежат все остальные прямые (этот класс отвечает шумовым прямым).

Рассмотрим ситуацию, при которой на изображении присутствует группа прямых, пересекающихся в одной точке  $(x_0, y_0)$ . Тогда, согласно уравнению (1), точки в пространстве параметров, соответствующие этим прямым, после преобразования Хафа будут лежать на синусоиде, период которой фиксирован, а амплитуда и фаза задается параметрами  $(x_0, y_0)$ . С другой стороны, найденные параметры синусоиды в пространстве Хафа соответствуют точке схода на изображении, т.е. поиск точек схода на изображении соответствует поиску параметров синусоид в пространстве преобразования Хафа.

Формальная постановка задачи разбиения прямых на классы (точек в пространстве параметров на синусоиды) и оценки точек схода (параметров синусоид) как задачи самообучения описана в следующем подразделе.

#### **Постановка задачи разбиения прямых на классы и оценивания точек схода**

Пусть  $\Omega$  ( $\omega_i \in \Omega, i = 1, \dots, n$ ) – множество всех возможных прямых на входном изображении (точек в пространстве преобразования Хафа), а  $K$  – множество классов прямых. На декартовом произведении  $\Omega \times K$  будем считать заданным с точностью до некоторых параметров распределение вероятностей  $p_{\Omega K}$ . Величина

$p_{\Omega K}(\omega, k; a_k) = p_K(k) \cdot p_{\Omega K}(\omega, a_k)$ ,  $\omega \in \Omega, k \in K$  определяет совместную вероятность прямой  $\omega$  и вероятность ее принадлежности к классу  $k$  в зависимости от значения параметра  $a_k$ . Величина  $p_K(k)$  определяет априорную вероятность класса  $k$ , а  $p_{\Omega K}(\omega; a_k)$  – вероятность прямой  $\omega$  при условии ее принадлежности к классу  $k$ . Вероятность совокупности  $\omega_1, \dots, \omega_n \in \Omega$  выделенных на изображении прямых равна:

$$\begin{aligned} l(p_K(k), a_k \mid k \in K) &= \\ &= \prod_{i=1}^n \sum_{k \in K} p_K(k) \cdot p_{\Omega K}(\omega_i; a_k). \end{aligned} \quad (2)$$

Совокупность величин  $M = (p_K(k), a_k \mid k \in K)$  назовем *моделью*.

Задача разбиения совокупности  $\omega_1, \dots, \omega_n$  выделенных на изображении линий на классы и оценка соответствующих точек схода заключается в поиске такой модели  $M^*$ , которая максимизирует вероятность этой совокупности прямых (2):

$$M^* = \arg \max_M \left( \prod_{i=1}^n \sum_{k \in K} p_K(k) \cdot p_{\Omega|K}(w_i; a_k) \right). \quad (3)$$

Укажем теперь, какой вид имеет распределение  $p_{\Omega|K}(\omega; a_k)$ .

Рассмотрим произвольную совокупность параллельных в трехмерном пространстве линий. В случае когда плоскость проектирования камеры не параллельна им, проекции прямых на эту плоскость пересекаются в одной точке – точке схода этих прямых (вертикальные линии на рис. 3). Вследствие действия таких факторов, как неточность выделения прямых на изображении, искривлений, обусловленных оптической системой фотокамеры, и т.п., эти прямые не пересекаются в одной точке. Будем искать точку, минимизирующую сумму квадратов расстояний до всех прямых совокупности. Отклонение (расстояние, взятое со знаком) точки  $(x_0, y_0)$  от прямой, которая задается параметрами  $\theta$  и  $\rho$ , равно  $x \cdot \cos \theta + y \cdot \sin \theta - \rho$ . Будем считать распределение отклонения прямых совокупности от точки схода гауссовым. Таким образом, для класса  $k$ , соответствующего точке схода  $(x_k, y_k)$ , распределение вероятностей  $p_{\Omega|K}(\omega; a_k)$  принимает следующий вид:

$$p_{\Omega|K}(\omega; a_k) = \frac{1}{\sqrt{2\pi\sigma_k}} \cdot e^{-\frac{(x_k \cos \theta + y_k \sin \theta - \rho)^2}{2\sigma_k^2}}, \quad (4)$$

где  $\sigma_k$  – дисперсия гауссового распределения, а  $a_k = (x_k, y_k, \sigma_k)$ .

В случае когда плоскость проектирования камеры параллельна совокупности параллельных в пространстве прямых, их проекции на нее также параллельны (горизонтальные линии на рис. 3), и, следовательно, задаются только одним параметром – углом наклона  $\theta$ . Предположения, аналогичные описанным, приводят к гауссовому распределению с математическим ожиданием  $\theta_k$  угла наклона линий  $k$ -го класса:

$$p_{\Omega|K}(\omega; a_k) = \frac{1}{\sqrt{2\pi\sigma_k}} \cdot e^{-\frac{(\theta - \theta_k)^2}{2\sigma_k^2}}, \quad (5)$$

где  $\sigma_k$  – дисперсия гауссового распределения, а  $a_k = (\theta_k, \sigma_k)$ .

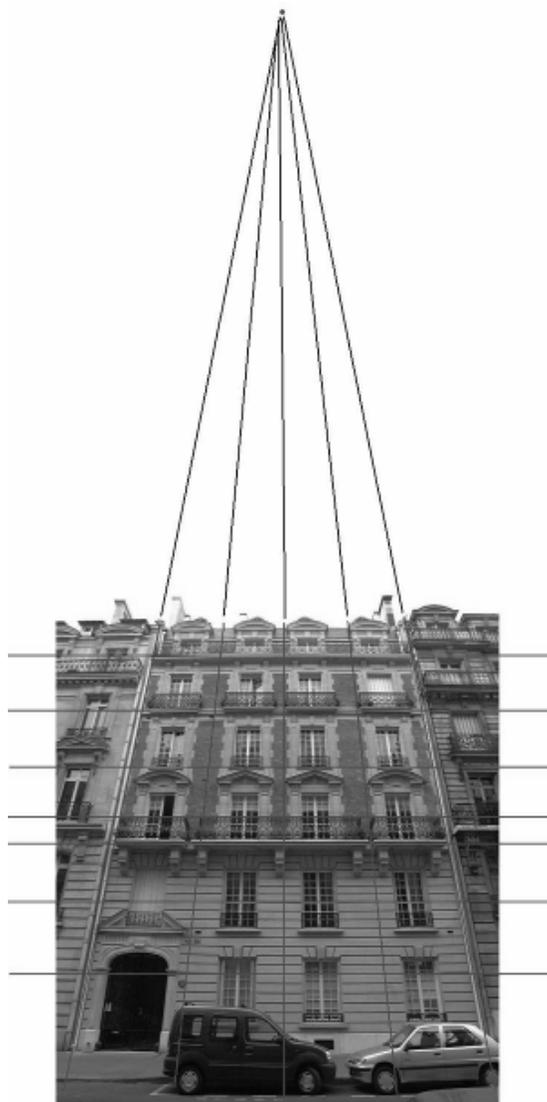


Рис. 3. Точки схода на изображении: конечная и на бесконечности (линии на изображении указывают на расположение точек схода)

В этом случае будем говорить, что соответствующая точка схода находится на бесконечности.

Кроме двух рассмотренных классов параллельных в пространстве линий, выделяется отдельный класс всех остальных прямых не параллельных в трехмерном пространстве. Назовем его классом шумовых линий. К этому классу

отнесем все прямые, не попавшие в первые  $m - 1$  классов. Распределение этих линий считаем равномерным на прямоугольнике  $\{(\theta, \rho) \mid \theta_{\min} < \theta < \theta_{\max}, \rho_{\min} < \rho < \rho_{\max}\}$ :

$$p_{\Omega K}(\omega) = \frac{1}{(\theta_{\max} - \theta_{\min}) \cdot (\rho_{\max} - \rho_{\min})}. \quad (6)$$

Это распределение задано полностью и не зависит от параметров.

При постановке задачи поиска модели, максимизирующей вероятность (2) совокупности  $\omega_1, \omega_2, \dots, \omega_n$  выделенных на изображении линий, будем считать известным количество классов, которые отвечают конечным (распределение вида (4)) и бесконечным (распределение (5)) точкам схода. Отдельный класс шумовых линий (распределение (6)) всегда один.

Поиск модели  $(p_K(k), a_k \mid k \in K)$  включает в себя в том числе и поиск точек схода, содержащихся в параметрах распределений  $a_k \mid k \in K$ .

### **Решение задачи разбиения прямых на классы и оценивания точек схода**

Для решения задачи 3 использован алгоритм самообучения [13], известный также как EM-алгоритм [15].

Алгоритм самообучения является итеративным. Он начинает работать с некоторой, пусть и неправильной, модели  $M^0 = (p_K^0(k), a_k^0 \mid k \in K)$ , на каждом шаге  $t$  строит новую модель  $M^t$  и для каждой прямой  $w_i$  из обучающей совокупности оценивает апостериорные вероятности  $\alpha^t(i, k)$  принадлежности ее к каждому из классов  $k \in K$  по следующим формулам:

$$\alpha^t(i, k) = \frac{p_K^t(k) \cdot p_{\Omega K}(\omega_i; a_k^t)}{\sum_{k' \in K} p_K^t(k') \cdot p_{\Omega K}(\omega_i; a_{k'}^t)}, \quad (7)$$

$$p_K^{t+1}(k) = \frac{\sum_{i=1}^n \alpha^t(i, k)}{\sum_{k' \in K} \sum_{i=1}^n \alpha^t(i, k')} = \frac{\sum_{i=1}^n \alpha_t(i, k)}{n}, \quad (8)$$

$$a_k^{t+1} = \arg \max_a \sum_{i=1}^n \alpha^t(i, k) \cdot \log p_{\Omega K}(w_i, a_k), \quad (9)$$

$k \in K$ .

Величины  $\alpha^t(i, k)$  и  $p_K^{t+1}(k)$  вычисляются непосредственно по формуле (7) и обозначают

апостериорные вероятности прямой  $\omega_i$  принадлежать классу  $k$ . Оценки  $p_K^{t+1}(k)$  априорных вероятностей классов также вычисляются непосредственно по формуле (8). Вычисление формулы (9) происходит методом наименьших квадратов. В результате в случае, когда положение точек схода задается распределением (4), происходит минимизация суммы взвешенных квадратов расстояний от точки до прямых соответствующего класса. Для точек, задаваемых распределением (5), минимизируется сумма взвешенных квадратов отклонений от угла наклона. Весами являются апостериорные вероятности  $\alpha^t(i, k)$  принадлежности прямых данному классу.

Об указанном алгоритме [13] известно, что: (i) он монотонно увеличивает (не уменьшает) значение функции правдоподобия  $l(p_K^t(k), a_k^t \mid k \in K)$  (см. (2)) входной совокупности линий; (ii) сходится в локальный минимум, зависящий от начальной модели  $M^0$ . Экспериментальные результаты применения алгоритма и задания начальной модели  $M^0$  рассматриваются в следующем разделе.

### **Результаты**

Экспериментальная проверка запрограммированного метода проводилась на базе данных изображений жилых кварталов Парижа, уменьшенные копии которых используются в интерактивной картографической интернет-системе [www.Mappy.com](http://www.Mappy.com). Использовалась выборка более чем из тысячи изображений этой базы. Процент правильно найденных точек схода (считаем, что результат неправильный, если хотя бы одна точка схода была найдена неправильно) зависит от типа изображения (типы изображений описаны далее) и в среднем составляет 95%.

Особенность статьи – наличие априорной информации о расположении точек схода на изображении, содержащейся в базе данных, а именно: известно – справа, слева или фронтально была сделана фотография. При использовании этой информации для задания начальных условий для алгоритма самообучения (например, для

изображений, сделанных слева, точка схода находится справа на изображении), алгоритм самообучения на практике сходится к нужным положениям точек схода. Примеры работы описанного метода показаны на рис. 4, и 5 (в нижнем ряду).



Рис. 4. Результаты работы алгоритма на изображениях с двумя точками схода

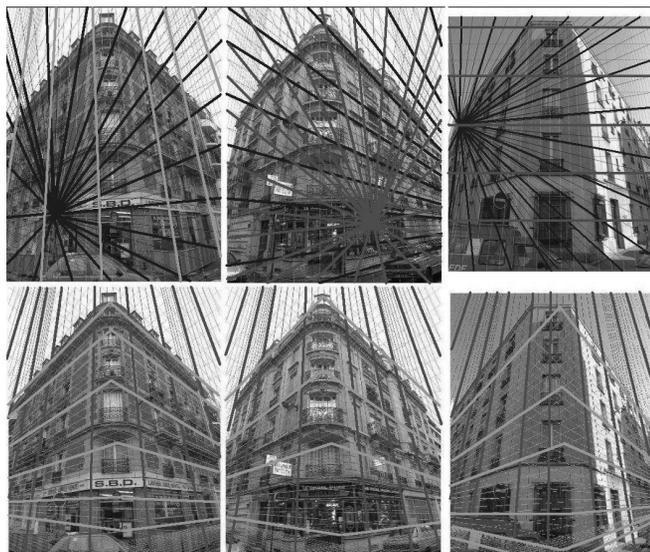


Рис. 5. Верхний ряд – результаты работы алгоритма [9]; нижний ряд – результаты работы метода, описанного в статье

Изображения тестовой выборки можно условно разделить на два типа. На изображениях первого типа содержится только один фасад (одна плоскость). Ей соответствуют две точки схода: вертикальная, в которой пересекаются линии, перпендикулярные поверхности земли, и горизонтальная, в которой пересекаются линии, параллельные земной поверхности и плоскости фасада. Примеры таких изображений представлены на рис. 4. На изображениях вто-

рого типа есть несколько фасадов, расположенных в разных плоскостях в пространстве. Каждой плоскости соответствует своя горизонтальная точка схода, вертикальная точка схода одинакова для всех фасадов. Количество горизонтальных точек схода на изображении такое же, как количество фасадов, расположенных в разных не параллельных плоскостях. Примеры изображений второго типа приведены на рис. 5.

Для изображений первого типа в базе данных указано расположение точки съемки относительно плоскости фасада, а именно: указано слева, справа или по центру относительно фасада находилась камера во время съемки. Левое и правое расположения камеры соответствуют конечным горизонтальным точкам схода (расположенным соответственно справа и слева на изображении). В этом случае мы использовали распределение вида (4) для соответствующего множества линий. Для фронтального расположения камеры точка схода – бесконечна, и соответствующее множество линий описывается распределением вида (5). Изображения второго типа, как правило, соответствуют угловым зданиям, для которых есть три точки схода: слева, справа и сверху. Количество, тип и начальные положения точек схода выбираются в соответствии с этой априорной информацией.

На изображениях второго типа расположены либо две фасадных плоскости, соответствующие правому и левому конечным расположениям точек схода, либо три фасадных плоскости с одной дополнительной точкой на бесконечности. В первом случае мы использовали три (для двух горизонтальных и вертикальной точек схода) распределения вида (4), а во втором вводился дополнительный класс линий, который описывается распределением вида (5). Для сопоставления этих двух случаев мы использовали карту города.

В таблице приведено сравнение двух подходов: обозначение *RANSAC* используется для алгоритма, описанного в [9], который считается одним из лучших в области, а обозначение *EM* – для алгоритма, описанного в статье. Из срав-

нения видно, что наш алгоритм сравним с алгоритмом *RANSAC* на простых изображениях с одним фасадом и значительно превосходит его на изображениях, содержащих несколько плоскостей фасадов. Объяснение эффективности нашего алгоритма в наличии априорной информации о примерном расположении точек схода. Такую информацию сложно эффективно использовать при подходах, базирующихся на алгоритме *RANSAC*.

На изображениях первого типа все точки схода находились с помощью предложенного метода правильно приблизительно на 97% изображений тестовой выборки, на изображениях второго типа – около 99%. В среднем 95% случаев отрабатываются правильно (изображений второго типа меньше).

Основные ошибки связаны с недостаточным количеством выделенных на фасаде линий и с неправильным определением типа точки схода.

Сравнение подходов, использующих алгоритмы *RANSAC* и EM

Тип изображения	Количество изображений	<i>RANSAC</i> , %	EM, %
С одним фасадом	920	99	97
Угловое здание	182	47	88

Существующие алгоритмы нахождения точек схода при наличии информации о расположении точек схода работают хуже. Проведено сравнение с одной из наилучших модификаций алгоритма *RANSAC* – алгоритмом Эндрю Циссермана [9]. В этом алгоритме используются сегменты прямых, найденных с помощью разбиения выделенных детектором границ Кэнни линий на прямые участки. В качестве расстояния от точки до сегмента используется минимальное из средних отклонений точек сегмента до линий, проходящих через точку, до которой подсчитывается расстояние. Точка схода оценивается как такая, которая минимизирует сумму определенных таким образом расстояний до сегментов класса (классы определяются с помощью алгоритма *RANSAC*).

Из-за проблем, описанных ранее, при количестве точек схода больше чем две, этот алгоритм часто оценивает точки схода неправильно

(рис. 5). При этом на изображениях с двумя точками схода ошибок практически не возникает.

**Заключение.** В статье предложен новый алгоритм поиска точек схода. Задача поиска точек схода сформулирована как задача обучения без учителя [13] и решена с помощью EM-алгоритма [13, 15]. Предложенный метод позволяет одновременно разделять выделенные линии на классы, отвечающие точкам схода, и оценивать точки схода по линиям, попавшим в каждый из классов. Отдельный класс равномерно распределенных линий введен для описания шумовых линий, возникающих на стадии выделения линий на изображении. Предложенный подход может быть эффективно использован в случаях, когда доступна априорная информация о количестве и примерном расположении точек схода. В этом случае он работает эффективнее широко используемых подходов на базе алгоритма *RANSAC* [12].

1. *Hartley R., Zisserman A.* Multiple View Geometry in Computer Vision. – United Kingdom, Cambridge University Press, 2003. – 656 p.
2. *Suttrop T., Bucher T.* Robust vanishing point estimation for driver assistance // Proc. of the 9th Intern. IEEE Conf. on Intel. Transp. Syst. (ITSC 2006), 2006. – P. 1550–1555.
3. *Learning class specific edges for vanishing point estimation / O. Barinova, A. Kuzmishkina, A. Vezhnevets et al.* // Proc. of Graphicon'2007, Moscow, Russia, June 2007. – P. 162–165.
4. *Almansa A., Desolneux A., Vamech S.* Vanishing point detection without any a priori information // Pat. Analysis and Machine Intel., April 2003. – **25**, N 4. – P. 502–507.
5. *Aguilera D.G., Gemez Lahoz J., Finat J.* Codes. A new method for vanishing points detection in 3D reconstruction from a single view // ISPRS Commission V, WG V/2. Mestre (Venice), 2005.
6. *Carsten Rother.* A new approach for vanishing point detection in architectural environments // Proc. of the British Machine Vision Conf. – 2002. – **20**. – P. 647–656.
7. *Vanishing point detection: representation analysis and new approaches / V. Cantoni, L. Lombardi, M. Porta et al.* // Proc. 11th Intern. Conf. on Volume, Image Analysis and Processing, Issue, 26–28 Sep. 2001. – P. 90–94.

Окончание на стр. 29