

Алгоритм формирования видеопанорамы и его программная реализация с применением технологии CUDA

И.А. Кудинов^а, О.В. Павлов^а, И.С. Холопов^{а,б}, М.Ю. Храмов^а

^а Государственный Рязанский приборный завод, 390000, Семинарская, 32, Рязань, Россия

^б Рязанский государственный радиотехнический университет, 390005, Гагарина, 59/1, Рязань, Россия

Аннотация

Разработан и реализован с применением языка CUDA алгоритм формирования видеопанорамы по информации от пяти различных предварительно откалиброванных камер с перекрывающимися полями зрения. Для управляемого оператором окна интереса размером 1024×768 пикселей с частотой 50 кадров/с реализованы операции компенсации дисторсии, «сшивки» изображений на внутренней поверхности виртуальной сферы и сглаживания перепадов яркости (блендинга).

Ключевые слова: видеопанорама; калибровка камеры; компенсация дисторсии; сферическая панорама; окно интереса; инклинометр; блендинг; технология CUDA

1. Введение

Автоматическое формирование видеопанорам высокого разрешения по информации от камер с пересекающимися полями зрения является одним из актуальных направлений развития обзорных систем технического зрения. Навигация по панораме предполагает, как правило, наличие управляемого пользователем окна интереса (ОИ) с задаваемыми угловыми размерами поля зрения и разрешением. Такой подход является альтернативой применению обзорных оптико-механических систем, поскольку перемещающееся по видеопанораме ОИ [1] позволяет реализовать в авиационной и наземной технике концепцию так называемой прозрачной кабины [2, 3], а также предоставлять информацию о различных участках панорамы нескольким пользователям одновременно без механического перемещения видеокамер.

2. Постановка задачи

Панорамное изображение от N кадров с перекрывающимися (либо попарно перекрывающимися) полями зрения формируется путем нахождения соответствия между пиксельными координатами каждого кадра. Такое соответствие для кадров камер с номерами i и j , $i, j = 1..N$, $i \neq j$, определяется матрицей гомографии \mathbf{H}_{ij} размерности 3×3 [4]:

$$\mathbf{x}_i = \mathbf{H}_{ij} \mathbf{x}_j, \quad (1)$$

где матричное преобразование (1) выполняет пересчет однородных координат пикселей изображения j -й камеры в систему координат i -й камеры, $\mathbf{x}_j = [u_j, v_j, 1]^T$ и $\mathbf{x}_i = [u_i, v_i, 1]^T$ – однородные пиксельные координаты i -го и j -го изображений, (u, v) – координаты пикселя, расположенного на пересечении u -й строки и v -го столбца.

Существует несколько подходов к формированию панорам. При отсутствии априорной информации оценка матрицы гомографии \mathbf{H}_{ij} основана на выделении m особых точек (ОТ), описании их окрестностей с использованием дескрипторов и автоматическом поиске соответствий. Оценка матрицы $\mathbf{H} = [[h_1, h_2, h_3]^T, [h_4, h_5, h_6]^T, [h_7, h_8, 1]^T]$ в этом случае выполняется по псевдорешению (в смысле минимума квадрата ошибки) переопределенной системы уравнений для $m \geq 4$ пар соответствий пиксельных координат (так называемых инляеров):

$$\mathbf{A}_{(2m \times 9)} \mathbf{h}_{(9 \times 1)} = \mathbf{0}_{(2m \times 1)}, \quad (2)$$

$$\mathbf{A}_{(2 \times 9)} = \begin{bmatrix} -u_j & -v_j & -1 & 0 & 0 & 0 & u_i u_j & u_i v_j & u_i \\ 0 & 0 & 0 & -u_j & -v_j & -1 & v_i u_j & v_i v_j & v_i \end{bmatrix}.$$

Псевдорешением (2) является последний вектор-столбец матрицы \mathbf{V} результата SVD-разложения матрицы $\mathbf{A}_{(2m \times 9)}$, соответствующий минимальному собственному числу Σ_{\min} :

$$\mathbf{A}_{(2m \times 9)} = \mathbf{U} \Sigma \mathbf{V}^T, \quad \mathbf{h} = \mathbf{V}^{<\Sigma_{\min}>}.$$

Платой за универсальность такого подхода к формированию панорамы является низкая эффективность поиска пар соответствий ОТ на однородных поверхностях (дорожное покрытие, трава, пашня, лес, водная гладь, небо) и в сложных условиях наблюдения (осадки, низкая освещенность, контровый свет).

При наличии априорных сведений о взаимном положении систем координат камер, получаемых в ходе предварительной калибровки, матрица гомографии может быть оценена по формуле:

$$\mathbf{H}_{ij} = \mathbf{K}_i[\mathbf{R}_{ij} - \mathbf{t}_{ij}\mathbf{n}^T/d]\mathbf{K}_j^{-1}, \quad (3)$$

где \mathbf{K}_i и \mathbf{K}_j – матрицы внутренних параметров, $i, j = 1..N$, \mathbf{R}_{ij} и \mathbf{t}_{ij} – соответственно матрица поворота и вектор параллельного переноса системы координат камеры j относительно i , d – длина перпендикуляра до плоскости съемки с нормалью \mathbf{n} относительно системы координат опорной камеры (в рассматриваемом случае – с номером i). Если расстояние до наблюдаемых объектов велико ($\mathbf{t}_{ij} \ll d$), то исходя из (3) справедливо приближенное равенство:

$$\mathbf{H}_{ij} \approx \mathbf{K}_i\mathbf{R}_{ij}\mathbf{K}_j^{-1}. \quad (4)$$

При объединении информации от нескольких камер целесообразно формировать результирующее панорамное изображение не на плоскости в соответствии с (1), а на виртуальной поверхности равномерной кривизны: как правило, сферы или цилиндра единичного радиуса [5], что позволяет работать с нормированными однородными пиксельными и пространственными координатами. Геометрическая постановка задачи объединения кадров $N = 3$ камер с пересекающимися полями зрения на поверхности сферы (система координат камеры с номером 0 принята за опорную) показана на рис. 1. Предполагается, что по аналогии с (4) при $\mathbf{t}_{ij} \ll d$ несовпадением нодальных точек объективов камер можно пренебречь и свести их в точку центра сферы O .

На рис. 1 символом \mathbf{M} обозначены пространственные однородные координаты точечного объекта, изображение которого проецируется на поверхность сферы в точку с пространственными координатами $\mathbf{M}_{\text{сф}}$, $\|\mathbf{M}_{\text{сф}}\| = 1$, в системе координат сферы $OX_{\text{сф}}Y_{\text{сф}}Z_{\text{сф}}$, а символами \mathbf{x}_0 и \mathbf{x}_2 – пиксельные координаты точек с изображением $\mathbf{M}_{\text{сф}}$ на кадрах камер с номерами 0 и 2 соответственно.

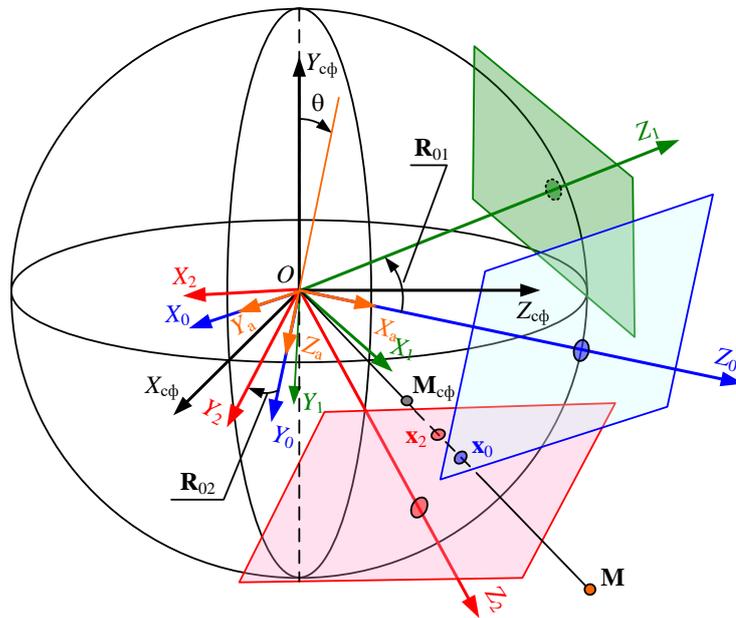


Рис. 1. Виртуальная сфера единичного радиуса для формирования панорамы на ее поверхности.

Корректное формирование сферической панорамы предполагает наличие информации об угловом отклонении оптических осей камер относительно плоскости горизонта и в данной работе решается установкой инклинометра на основе трехосного MEMS акселерометра, с которым связана система координат $OX_aY_aZ_a$, на камеру, выбранную опорной. Тогда крен φ и тангаж θ ее системы координат $OX_0Y_0Z_0$ оцениваются по формулам [6]:

$$\varphi = \text{atan2}(a_y, a_z), \quad \theta = \text{atan2}[-a_x, (a_y^2 + a_z^2)^{0.5}], \quad (5)$$

где $[a_x, a_y, a_z]^T$ – вектор показаний акселерометра с учетом его калибровки [7].

Выбор пикселей в областях, где луч виртуальной сферы $\mathbf{OM}_{\text{сф}}$ пересекает несколько кадров, может быть реализован по нескольким критериям: например, максимуму расстояния от пикселя до линии пересечения плоскостей изображений камер или минимуму длины вектора нормированных пиксельных координат. В работе с целью минимизации вычислений реализован критерий выбора по минимуму углового отклонения пикселя от оптической оси камеры, поскольку для таких пикселей, как правило, меньше ошибка коррекции дисторсии:

$$\min_i [(\mathbf{R}_{0i}\mathbf{R}_{\varphi\theta})^{\langle 3 \rangle} \cdot \mathbf{M}_{\text{сф}}], \quad (6)$$

$$\mathbf{R}_{\varphi\theta} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \quad (7)$$

где $\mathbf{R}_{\phi\theta}$ матрица поворота системы координат опорной камеры относительно плоскости горизонта, \mathbf{R}_{0i} – матрица поворота i -й камеры относительно опорной, оцениваемая во время калибровки, $\langle k \rangle$ – k -й столбец матрицы, а символом « \cdot » обозначено скалярное произведение векторов.

Для реализации функции скользящего ОИ размером $(W \times H)$ пикселей с угловыми размерами по горизонтали и вертикали $\Delta\phi_w$ и $\Delta\phi_h$ соответственно вводится кватернион $\mathbf{q}_{\text{виз}}$ [8], задающий азимут α и угол места β линии визирования:

$$\mathbf{q}_{\text{виз}} = [\cos(\alpha/2)\cos(\beta/2), \cos(\alpha/2)\sin(\beta/2), \sin(\alpha/2)\cos(\beta/2), \sin(\alpha/2)\sin(\beta/2)]^T. \quad (8)$$

Каждому пикселю ОИ с координатами (u, v) соответствует точка $\mathbf{M}_{uv} = [x_{uv}, y_{uv}, z_{uv}]^T / \|[x_{uv}, y_{uv}, z_{uv}]^T\|$ на сфере (см. рис. 2), определяемая радиус-вектором с соответствующим кватернионом

$$\mathbf{q}_{uv} = \mathbf{q}_{\text{виз}}\mathbf{q}_{uv0}, \quad (9)$$

где из геометрических построений рис. 2 начальному ($\alpha = \beta = 0, z_{uv} = 1$) положению пикселей ОИ (u, v) соответствуют кватернионы

$$\begin{aligned} \mathbf{q}_{uv0} &= [\cos(\alpha_u/2)\cos(\beta_v/2), \cos(\alpha_u/2)\sin(\beta_v/2), \sin(\alpha_u/2)\cos(\beta_v/2), \sin(\alpha_u/2)\sin(\beta_v/2)]^T, \\ \alpha_u &= \arctg(x_{uv}), \quad \beta_v = \arcsin[y_{uv}/(x_{uv}^2 + y_{uv}^2 + 1)^{0.5}], \\ x_{uv} &= (2u/W - 1)\text{tg}(\Delta\phi_w/2), \quad y_{uv} = -(2v/H - 1)\text{tg}(\Delta\phi_h/2). \end{aligned} \quad (10)$$

Кватернион \mathbf{q}_{uv} позволяет определить координаты точки \mathbf{M}_{uv} в системе координат сферы [8]:

$$\mathbf{M}_{uv} = [2(q_x q_z + q_w q_y), 2(q_y q_z - q_w q_x), q_w^2 + q_z^2 - (q_x^2 + q_y^2)]^T, \quad (11)$$

где q_w и $[q_x, q_y, q_z]^T$ – соответственно скалярная и векторная части кватерниона \mathbf{q}_{uv} .

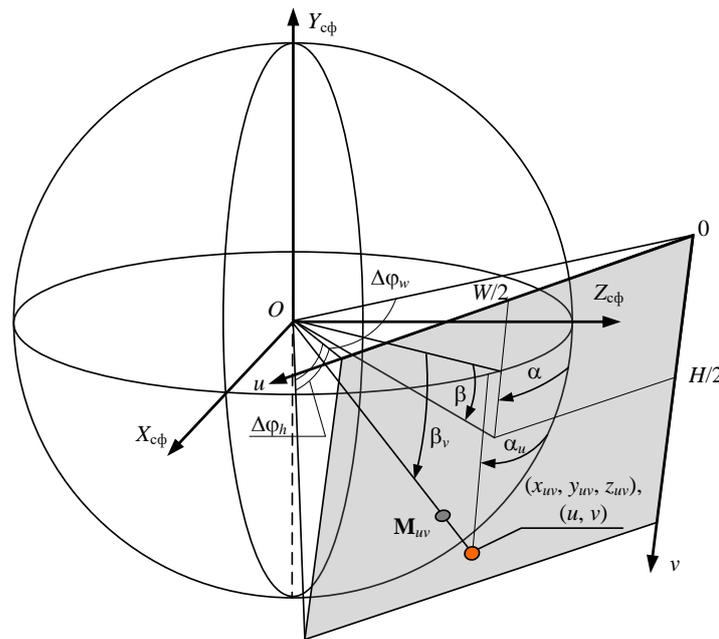


Рис. 2. Связь угловых и пространственных координат точки \mathbf{M}_{uv} на поверхности виртуальной сферы.

На плоскости кадра каждой i -й камеры пиксельные координаты точек \mathbf{x}_{uvi} , соответствующих точке \mathbf{M}_{uv} , определяются (с учетом ориентации относительно плоскости горизонта) через матрицу проекции [4]:

$$\mathbf{x}_{uvi} = \mathbf{K}_i[\mathbf{R}_{0i}\mathbf{R}_{\phi\theta} \mid \mathbf{0}]\mathbf{M}_{uv}, \quad (12)$$

где $\mathbf{0} = [0, 0, 0]^T$, а выбор пикселя, переносимого с кадра i -й камеры на ОИ, производится по критерию (6).

Для камер с широкоугольными объективами необходима компенсация дисторсии: для нормированных координат

$$\mathbf{x}_{ni} = \mathbf{K}_i^{-1}\mathbf{x}_{uvi} \quad (13)$$

по модели Брауна – Конради [9] (для уменьшения объема вычислений в работе используются только первые два коэффициента радиальной дисторсии) оцениваются нормированные координаты без дисторсии, по ним путем умножения на матрицу внутренних параметров \mathbf{K}_i вычисляются пиксельные координаты

$$\mathbf{x}_{\text{корр}i} = \mathbf{K}_i \mathbf{x}_{\text{н}i} \quad (14)$$

и, поскольку они являются дробными, выполняется интерполяция значения яркости (в работе используется билинейная интерполяция [10]).

Поскольку сцены каждой из камер различны, то в режиме автоматического выбора времени экспонирования средняя яркость кадров, составляющих панораму, оказывается различной. Поэтому после составления панорамы (либо заполнения ОИ) дополнительно необходимо выполнение процедуры сглаживания перепадов яркости – блендинга. В работе для этого применяется подход, аналогичный [11].

3. Алгоритм формирования видеопанорамы с управляемым ОИ и его программная реализация

Алгоритм формирования изображения на ОИ содержит следующие этапы.

1. Инициализация: вычисление кватернионов \mathbf{q}_{iv0} по (10).

Основной цикл:

2. Оценка текущего углового положения опорной камеры по (5) и матрицы поворота $\mathbf{R}_{\varphi\theta}$ по (7).
3. Вычисление кватерниона поворота для заданного углового положения на панораме $\mathbf{q}_{\text{виз}}$ по (8) и кватернионов \mathbf{q}_{iv} по (9).
4. Заполнение ОИ пикселями с камер (с компенсацией дисторсии) в соответствии с (11)-(14) по критерию (6).
5. Выполнение процедуры блендинга (опциональное).

Поскольку для каждого пикселя ОИ обработка является однородной, это позволяет применить процедуру распараллеливания вычислений. В макете системы формирования видеопанорамы на основе ПЭВМ для реализации шагов алгоритма 1, 4 и 5 задействованы ресурсы видеокарты NVIDIA: весь объем вычислений с применением технологии CUDA и языка CUDA C разделен на 3072 параллельно выполняемых блока (64 по горизонтали и 48 – по вертикали) по 256 нитей в каждом (по 16 нитей по горизонтали и вертикали). При этом индексы пикселей ОИ (u, v) в соответствии с [12] представляются в виде:

$$u = \text{blockIdx.y} * \text{blockDim.y} + \text{threadIdx.y}; \quad v = \text{blockIdx.x} * \text{blockDim.x} + \text{threadIdx.x}; \quad (15)$$

В (15) приняты следующие стандартные для CUDA обозначения: blockIdx – идентификатор (номер) блока, blockDim – размерность блока, threadIdx – идентификатор (номер) параллельно исполняемой нити, атрибуты x и y указывают ось системы координат двумерного евклидова пространства, вдоль которой отсчитывается идентификатор или размерность.

Поскольку копирование содержимого памяти CPU в память GPU и обратно является сравнительно медленным [12], то при реализации алгоритма минимизировано количество подобных операций: при инициализации в памяти GPU заполняются массивы с данными о параметрах камер (матрицы внутренних параметров и коэффициенты дисторсии объективов) и кватернионах \mathbf{q}_{iv0} (10) для пикселей ОИ. В основном цикле работы кадры с каждой камеры угловые координаты линии визирования (центра ОИ) и оцененные по сигналам MEMS датчика угловые координаты копируются в память GPU, далее выполняются этапы 3-5 алгоритма и результат (массив значений яркости в ОИ) копируется обратно в память CPU для вывода на монитор ПЭВМ.

4. Результаты эксперимента

Макет системы формирования видеопанорамы реализован на ПЭВМ с процессором Intel Core-i5 и видеокартой NVIDIA GeForce GTX 560 Ti (384 ядра) и состоит из закрепленных на кронштейне специальной формы (рис. 3) пяти цифровых видеокамер с интерфейсом GigE: двух камер Basler acA2000 разрешением 2048×1088 пикселей с объективами Cowa с фокусным расстоянием 5 мм и трех камер IDS 5240 RE разрешением 1280×1024 пикселя с объективами Computar с фокусным расстоянием 5 мм (две крайние камеры) и Navitar с фокусным расстоянием 8 мм (центральная опорная камера), отладочной платы с MEMS сенсором InvenSense MPU-9250, установленной на опорной камере, двух моторизованных платформ Standa 8MR190-2 для перемещения кронштейна с камерами в азимутальной и угломестной плоскостях и USB-джойстика Defender Cobra M5 для навигации по панораме.

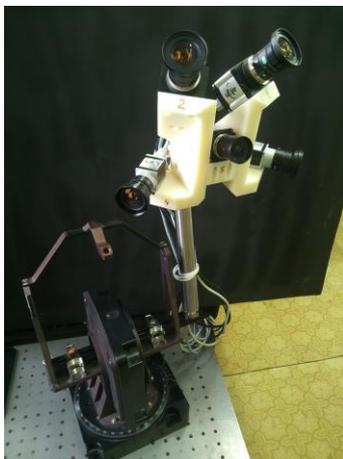


Рис. 3. Общий вид макета формирования видеопанорамы.

Макет позволяет формировать панораму разрешением 3600×2400 пикселей с угловыми размерами $180^\circ \times 120^\circ$ либо отображать ОИ с задаваемыми пользователем разрешением и угловыми размерами поля зрения.

Предварительная калибровка камер макета выполнялась с применением библиотек OpenCV [13]: по 40 кадрам тестового шаблона типа «шахматная доска», содержащего 12×11 клеток со стороной 3 см, оценивались матрицы внутренних параметров камер и коэффициенты дисторсии их объективов; затем по 40 кадрам тестового шаблона типа «шахматная доска», содержащего 9×6 клеток со стороной 3 см, оценивались матрицы поворота камер относительно опорной.

Влияние погрешности оценивания угловой ориентации опорной камеры относительно плоскости горизонта на геометрию панорамы показано на рис. 4. Как видно из рисунка, ошибка оценивания тангажа приводит к некорректному отображению горизонта (положение экватора сферической панорамы показано оранжевой линией).

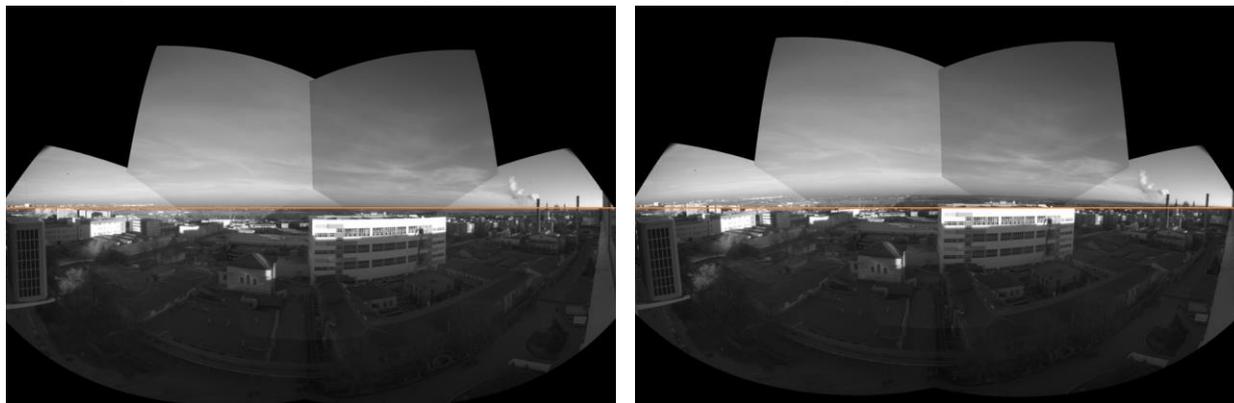


Рис. 4. Влияние погрешности оценки тангажа на результат формирования сферической панорамы (без блендинга): слева – тангаж опорной камеры совпадает с истинным, $\theta = 15^\circ$, справа – тангаж опорной камеры не совпадает с истинным, $\theta = 12^\circ$.

Результаты формирования ОИ разрешением 1024×768 пикселей и полем зрения $40^\circ \times 30^\circ$ с угловыми координатами линии визирования $\alpha = 15^\circ$ и $\beta = -5^\circ$ с блендингом и без приведены на рис. 5 и 6 соответственно. Затраты времени на формирование одного кадра в ОИ при реализации на CPU и GPU сведены в таблицу 1 (время копирования данных из памяти CPU в память GPU и обратно составляет 3,5 мс), а затраты времени при различных размерах ОИ – в таблицу 2.

Таблица 1. Время формирования ОИ 1024×768 пикселей, мс

	На CPU	На GPU с CUDA	Ускорение, раз
Без блендинга	114	6,5	17,5
С блендингом	272	17,2	15,8

Таблица 2. Время формирования ОИ на GPU с CUDA в зависимости от размера ОИ, мс

Разрешение ОИ, пикселей	800×600	1024×768	1280×1024
Без блендинга	5,1	6,5	9,7
С блендингом	12,9	17,2	23,4
Время копирования CPU ↔ GPU	2,9	3,5	3,7

На рис. 7 и 8 показаны результаты промежуточных вычислений для реализации блендинга: границы изображений камер в ОИ (рис. 7) и бинарная маска шириной 100 пикселей, по которой выполняется сглаживание яркости (рис. 8).



Рис. 5. ОИ без процедуры блендинга.



Рис. 6. ОИ после выполнения процедуры блендинга.

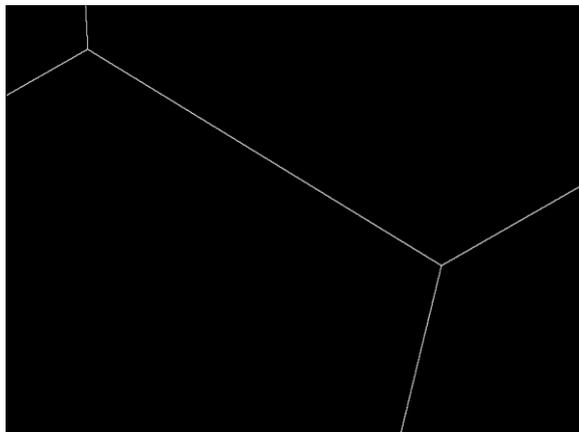


Рис. 7. Границы изображений камер в ОИ (бинарное изображение).

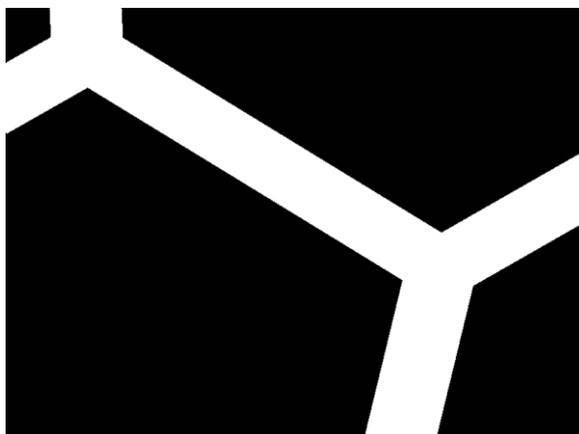


Рис. 8. Бинарная маска для процедуры блендинга, полученная по рис. 7.

Увеличение времени формирования кадра с блендингом более чем в 2 раза вызвано необходимостью выполнения вспомогательных процедур: поиска линий пересечения кадров камер в ОИ, определения зон для смешивания значений яркости и сглаживания изображений для оценивания низкочастотной составляющей яркости в кадре каждой камеры. Для уменьшения объема вычислений применяется процедура сглаживания с равновесным окном 8×8 пикселей и аккумулятором с вычислительной сложностью $O(n^2)$.

Для расстояний до объектов съемки $d > 70$ м и два-норм векторов параллельного переноса между системами координат i -й и опорной камер $\|t_{oi}\| < 15$ см гипотеза о совмещении нодалных точек их объективов обеспечивает ошибку «сшивки» изображений объектов на границах изображений не более 5 пикселей.

Результаты эксперимента показали, что использование ресурсов GPU позволяет сократить время формирования ОИ по сравнению с CPU до 16 раз.

5. Заключение

Разработанный и реализованный с применением технологии CUDA и распараллеливания вычислений на видеокарте алгоритм формирования видеопанорамы для окна интереса разрешением 1 Мп и угловыми размерами $40^\circ \times 30^\circ$ реализует с частотой 50 кадров/с навигацию по панорамному видеоизображению разрешением 8,2 Мп и угловыми размерами $180^\circ \times 120^\circ$.

Литература

- [1] DeNies M. Streaming panoramic video. Patent US 20020021353 A1, 2012.
- [2] AN/AAQ-37 Distributed Aperture System (DAS) for the F-35 — Access mode: <http://www.northropgrumman.com/Capabilities/ANAAQ37F35/Pages/default.aspx> (01.10.2016)
- [3] IronVision™ — Access mode: <http://elbitsystems.com/media/IronVision.pdf> (10.09.2016)
- [4] Hartley, R. Multiple View Geometry in Computer Vision / R. Hartley, A. Zisserman: 2nd edition. — Cambridge: Cambridge University Press, 2003. — 656 p.
- [5] Szeliski, R. Image alignment and stitching: a tutorial / R. Szeliski // Foundations and trends in computer graphics and vision. — 2006. — Vol. 2(1). — P. 1-104.
- [6] Tilt Sensing Using a Three-Axis Accelerometer — Access mode: http://www.freescale.com/files/sensors/doc/app_note/AN3461.pdf (11.04.2016)
- [7] Wang, L. Intelligent calibration method of low cost MEMS inertial measurement unit for an FPGA-based navigation system / L. Wang, F. Wang // International J. of Intelligent Engineering and Systems. — 2011, — Vol. 4(2). — P. 32-41.
- [8] Kuipers, J.B. Quaternions and Rotation Sequences / J.B. Kuipers. New Jersey: Princeton University. — 1998, 400 p.
- [9] Brown, D.C. Close-Range Camera Calibration / D.C. Brown // Photogrammetric Engineering. — 1971. — Vol. 37(8). — P. 855-866.
- [10] Parker, J.A. Comparison of interpolating methods for image resampling / J.A. Parker, R.V. Kenyon, D.E. Troxel // IEEE Trans. on Medical Imaging. — 1983. — Vol. 2(1). — P. 31-39.
- [11] Xiong, Y. Fast image stitching and editing for panorama painting on mobile phones / Y.Xiong, K. Pulli. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 13-18 June 2010. — P. 47-52.
- [12] Sanders, J. CUDA by Example / J. Sanders, E. Kandrot. — New York: Addison-Wesley, 2010. — 290 p.
- [13] Camera Calibration and 3D Reconstruction — Access mode: http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html (14.03.2015)