

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

СИНТЕЗ АЛГОРИТМА ФИЛЬТРАЦИИ НЕСТАЦИОНАРНОГО
ФОНА ПРИ АВИАЦИОННОМ НАБЛЮДЕНИИ

Выпускная квалификационная работа
студента 915 группы ФРТК

Слынько Юрия Вячеславовича

Заведующий кафедрой
доктор технических наук профессор

А.В. Меньшиков

Научный руководитель
кандидат технических наук, старший научный сотрудник

В.Н. Лагуткин

Рецензент
кандидат технических наук, старший научный сотрудник

А.П. Лукьянов

Москва 2003

Оглавление

1. Введение	3
2. Постановка задачи	4
3. Синтез алгоритма фильтрации фона при наблюдении с движущегося носителя	5
3.1 Особенности формирования фона	5
3.2 Принципы построения алгоритма	6
3.3 Создание карты фона	8
3.4 Определение высоты облачного слоя	13
3.5 Создание карты облаков	14
3.6 Обеление кадра и пространственная фильтрация	17
4. Анализ и результаты работы алгоритма	19
4.1 Анализ алгоритма создания карты фона	19
4.2 Анализ алгоритма определения высоты облачного слоя	21
4.3 Анализ алгоритма создания карты фона	22
4.4 Анализ комплексного итерационного алгоритма фильтрации	25
4.5 Анализ алгоритма пространственной фильтрации	27
4.6 Итоговые результаты работы алгоритма	28
5. Особенности программная реализация	29
6. Выводы	32
7. Литература	33

1. Введение

В обработке видео изображений очень распространена задача выделения полезных объектов из всего видеоряда, т.е. выделение полезных объектов на фоне шумов.

Если мы имеем дело с движущимся точечным или квазиточечным объектом, то такая задача называется задачей селекции движущихся целей. К сегодняшнему дню в этом направлении проделана большая работа. Хорошо исследована и решена задача для случая нормального белого шума. Тогда решение определяется отношением сигнал-шум, т.е. одним параметром. В случае не белого, т.е. коррелированного шума, задача несколько усложняется. Коррелированность позволяет уменьшить интенсивность шума с одновременным сведением его к белому, т.е. обелением. Тогда задача сводится к первой, но с «лучшим» отношением сигнал-шум. Такая задача требует более глубокого анализа, чем первая, и создания более сложных и изощрённых алгоритмов. Однако способы их создания достаточно хорошо теоретически изучены и описаны в литературе.

Самую большую трудность представляют задачи, в которых законы распределения вовсе неизвестны. В этом случае стараются описать неизвестные законы распределения функцией от неизвестных параметров, а затем по данным наблюдения определить эти параметры. Тогда удаётся свести задачу к предыдущей. При этом универсального способа определения таких параметров на сегодняшний день не существует. К тому же, в задачах обработки видеоизображений количество параметров зачастую может превышать количество информации, которая содержится в самом видеоизображении. Да и законы, которым подчиняется данная система, могут быть крайне сложными.

Ещё более осложняет проблему наличие разного рода разрывов функций, неограниченного возрастания их или их производных и т.д. Тогда вообще большинство полученных закономерностей и формул перестают быть справедливыми или выполняются лишь приближенно.

Именно такая картина имеет место при наблюдении с летящего самолёта Земной поверхности и облачного слоя. Неизвестными параметрами здесь являются карта рельефа Земной поверхности, карта облачного слоя со скоростями движения всех участков, законы рассеяния и излучения и т.д. При этом в нашем распоряжении лишь проекция на плоскость матрицы фотоприёмника. Также особо следует отметить сильную нелинейность задачи, связанную со сферичностью Земли. А если наблюдение ведётся в направлении горизонта, то и вовсе имеет место неограниченное возрастание многих производных. Дополнительные

проблемы создаёт существование разрывов функции яркости на границе раздела между морями, полями, лесами, облаками и т.д.

2. Постановка задачи

В данной работе рассмотрена задача обнаружения точечных движущихся объектов при наблюдении с летящего самолёта подстилающей поверхности.

Наблюдение ведётся с нескольких камер, направленных в разные области вблизи горизонта так, что при движении самолёта одна и та же область может появиться в поле зрения нескольких камер. Все камеры работают в одном и том же инфракрасном диапазоне.

При этом мешающий фон не только не является нормальным белым, но является ещё и нестационарным, т.е. движется относительно самолёта. Чтобы формализовать отличие такого фона от полезных объектов, будем считать эти объекты:

1. Достаточно малыми. Действительно, размер пикселя изображения на поверхности Земли больше сотни метров, а размер летающих объектов едва ли превышает эту величину.
2. Движущихся по законам, отличным от законов движения подстилающей поверхности (имеется в виду движения относительно видео сенсора). Причём, здесь важно только проекция этого движения на плоскость матрицы фотоприёмника, а движение вдоль луча зрения не учитывается.

Только объекты, удовлетворяющие этим требованиям, называются полезными.

При реализации алгоритма все параметры сенсора, как и параметры движения носителя, считались известными. Это следующие параметры:

1. высота полёта самолёта над уровнем моря
2. средняя высота поверхности Земли в области наблюдения
3. скорость полёта \vec{V}
4. количество видеокамер
5. угол между скоростью самолёта и оптической осью первой камеры β
6. угол между камерами

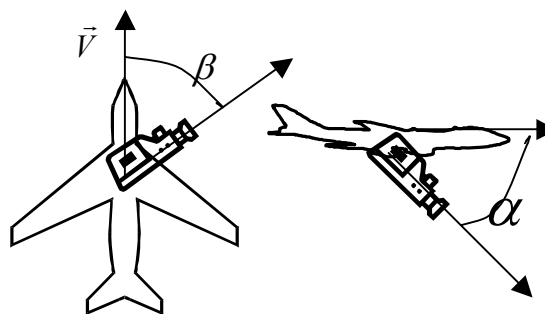


Рис.1

7. угол между горизонталью и оптической осью сенсора α
8. размер поля зрения в углах и в пикселях
9. темп получения кадров

Задачей данной работы было создание алгоритма удаления (фильтрации) мешающего фона, при этом полезные сигналы не должны искажаться. Входными данными этого алгоритма является последовательность кадров со всех камер, а выходом – изображение только полезных сигналов без мешающего фона.

3. Синтез алгоритма фильтрации фона при наблюдении с движущегося носителя

3.1 Особенности формирования фона

Работа алгоритма фильтрации исследовалась с помощью имитатора, позволяющего создать последовательность кадров в соответствии с физической моделью и с заданными характеристиками траектории движения носителя сенсора и параметрами самого сенсора. Также этот имитатор позволяет изменять внешние условия, такие, как освещённость, метеорологическую обстановку и т.д.

Основным уравнением для расчёта мощности излучения является уравнение переноса излучения:

$$\frac{dB}{ds} = -\alpha B + \sigma \int B(\vec{r}, \vec{l}, \lambda) x(\vec{r}, \gamma, \lambda) \frac{d\Omega}{4\pi} + \beta B_0, \text{ где}$$

- $B = B(\vec{r}, \vec{l}, \lambda)$ - спектральная плотность яркости (СПЯ), зависящая от координат точки наблюдения \vec{r} , направления наблюдения \vec{l} и длины волны λ .
- ds - дифференциал пути вдоль направления наблюдения.
- $\alpha = \alpha(\vec{r}, \lambda) = \sigma(\vec{r}, \lambda) + \beta(\vec{r}, \lambda)$ - коэффициент ослабления излучения.
- $\sigma(\vec{r}, \lambda), \beta(\vec{r}, \lambda)$ - коэффициенты рассеяния и поглощения излучения.
- $d\Omega$ - дифференциал пространственного угла.
- $x(\vec{r}, \gamma, \lambda)$ - индикатриса рассеяния (γ - угол рассеяния).

Принципиальная схема работы имитатора следующая:

1. Генерация рельефа Земной поверхности и коэффициента отражения.
2. Генерация рельефа облачного слоя и расчёт индикатрисы рассеяния для данных метеорологических параметров.
3. Расчёт мощности излучения на входном зрачке фотоприёмника на основе этих данных с учётом поглощения света в атмосфере
4. Расчёт кадра на выходе фотоприёмника с учётом функции рассеяния точки в оптической системе сенсора, собственных шумов матрицы фотоприёмника и конечных размеров пикселя.

3.2 Принципы построения алгоритма

Решение задачи обеления чаще всего можно свести к следующей схеме. Для каждого пикселя текущего кадра ищется оценка его яркости при условии, что соседние пиксели (пространственная корреляция) и/или пиксели из кадров, полученных в другое время (временная корреляция) имеют заданную яркость. Кадр из этих оценок будем называть карта фона. Обелённый кадр при этом получается вычитанием из текущего кадра карты фона.

Однако априорной плотности распределения яркости пикселей не существует, поэтому вместо неё используются различные параметрические модели распределений, что можно описать принципиальной формулой:

$$W_{ij}(t) = F_{ij}(t) - M_{ij}(t | F(t_n)..F(t_m), \vec{\gamma}(t, F(t_n)..F(t_m))), \text{ где}$$

- t_n и t_m – начальный и конечный моменты времени.
- $W_{ij}(t)$ – (i,j) пиксель обелённого кадра в момент времени t . При этом момент времени t необязательно является конечным моментом времени.
- $F_{ij}(t)$ – (i,j) пиксель исходного кадра в момент времени t .
- $M_{ij}(t | F(t_n)..F(t_m), \vec{\gamma}(F(t_k)..F(t_l)))$ – (i,j) пиксель карты фона в момент времени t . Он вычисляется на основе всех исходных кадров и вектора неизвестных параметров.
- $\vec{\gamma}(t, F(t_k)..F(t_l))$ - вектор неизвестных параметров в момент времени t , вычисленных на основе всех имеющихся исходных данных.

При этом необходимо внимательно следить, чтобы в модель не входил сам полезный сигнал и он не «принимался» за мешающий фон. Считается, что полезным сигналом является точечный объект, смещающийся относительно других объектов (объекты размером в несколько пикселей считаются принадлежащими подстилающей поверхности). Такими

объектами с достаточной точностью являются все объекты, чьи угловые размеры меньше разрешающей способности сенсора, а яркость недостаточна для того, чтобы было существенно «размывание» объекта за счёт дифракции и аберрации.

Главное используемое приближение состоит в том, что при вычислении оценки яркости пикселя используются только яркости соседних по пространству и по времени пикселей. При этом полагается, что эти яркости физически неизменны, а не стационарность процесса связана только с движением летательного аппарата (т.е. не учитываются ни изменения освещённости, ни ветер и т.д.).

Также, при реализации алгоритма предполагалось, что шум не удовлетворяет ни одному из условий на полезный сигнал, а не только какому-либо, как должно было бы следовать из отрицания понятия полезный сигнал. Т.е., если в видеоряде появятся изображения объектов, удовлетворяющих только одному из этих условий (например, маленькие облака, или зависший в воздухе вертолёт), алгоритм может дать неадекватный результат. Но, поскольку алгоритм использует на всех этапах усреднение (в том или ином смысле), можно ожидать, что несоответствие наблюдаемой ситуации принятой модели не сильно скажется на результатах.

Если положить для текущего кадра время $t_0=0$, то учитываются только кадры из «временного окна» со временами $t=-t_{pr}.. t_{nx}$, где t_{pr} и t_{nx} – параметры от 0 до 10.

Таким образом, основой алгоритма является создание карты фона на основе кадров из «временного окна». Эту задачу можно условно разделить на две подзадачи: определение рельефа подстилающей поверхности и облаков, и создание собственно карты фона на основе этого рельефа. Здесь существенны две проблемы. Во-первых, количество параметров, описывающих рельеф земной поверхности и облака чрезвычайно велико, поэтому необходимо использовать приближённые модели, позволяющие уменьшить число этих параметров. Этими параметрами в данном алгоритме выбраны высота облачного слоя и карта облаков в приближении, что эти облака тонкие и плоские. Во-вторых, задача распадается на подзадачи лишь приближенно, и для получения удовлетворительного результата необходимо использовать сложную итерационную схему:

1. Вычисляется карта фона, считая Землю идеальной сферой.
2. По карте фона определяется высота облачного слоя.
3. Определяется карта облачного слоя, т.е. определяется, какие пиксели кадра соответствуют облачному слою, а какие – поверхности Земли. При этом считается, что облака плоские и тонкие.

4. По полученной карте облачного слоя вновь вычисляется карта фона с рельефа, которая и используется для обеления кадра.
5. Обелённый кадр подвергается пространственной фильтрации.

Дальнейшее продолжение итерационного процесса, т.е. определение мелкого рельефа облаков или поверхности Земли, вряд ли может быть целесообразным, поскольку даже определение карты облаков даёт заметные ошибки, в то время как ошибки в определении рельефа не сильно сказываются на качестве фильтрации.

Таким образом, основная формула создания обелённого кадра следующая:

$$W_{ij}(t) = F_{ij}(t) - M_{ij}(t | F(-t_{pr})..F(t_{nx}), h(t, F(-t_{pr})..F(t_{nx})), C(t, h, F(-t_{pr})..F(t_{nx}))), \text{ где}$$

- $h(t, F(-t_{pr})..F(t_{nx}))$ - высота облачного слоя.
- $C(t, h, F(-t_{pr})..F(t_{nx}))$ - карта облаков на момент времени t .

3.3 Создание карты фона

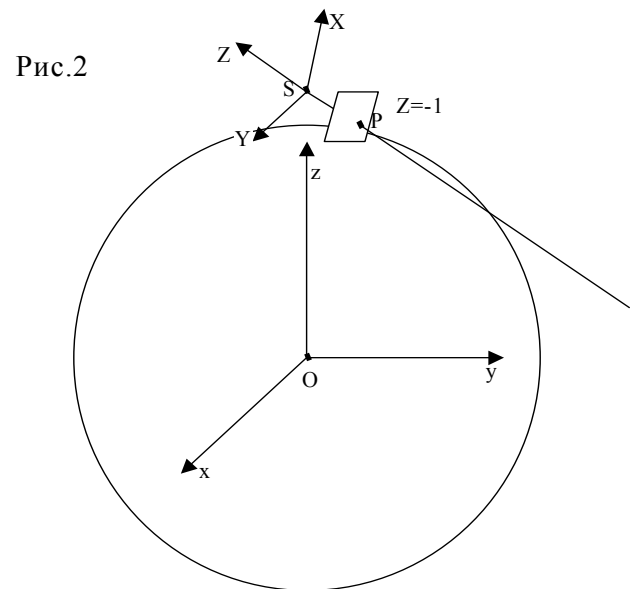
Точное вычисление карты фона является крайне трудоёмкой задачей, поэтому используется следующая схема приближённого вычисления.

При этом особое внимание следует уделять тому, что в данной задаче ведётся наблюдение пригоризонтной части поверхности Земли, и задача создание карты фона сильно не линейна, вплоть до неограниченного возрастания производных некоторых функций (например, площади поверхности Земли, соответствующей одному пикселю изображения).

Т.к. параметры движения самолёта известны, то для каждого кадра «временного окна» для каждой точки можно вычислить координаты этой же точки земной поверхности в текущем кадре. При этом необходимо знать высоты этих точек. В первом приближении они считаются одинаковыми (приближение идеальной сферы).

Схема пересчета точек из одного кадра в другой следующая:

1. Выбирается система координат с осью OZ, направленной в вдоль оптической оси сенсора, а оси OX и OY вдоль соответствующих осей матрицы



- фотоприёмника. Тогда координаты пикселя в плоскости $Z=-l$ (точка P) находятся умножением координат пикселя в кадре на угловой размер пикселя.
2. Координаты этой точки пересчитываются в систему отсчёта, связанную с Землёй, умножением вектора координат на матрицу поворота вокруг оси Oх на угол наклона камеры.
 3. Находится точка пересечения луча SP со сферой радиуса (R_3+H) , где R_3 – радиус Земли, а H – высота точки над Земной поверхностью. Эта задача сводится к решению квадратного уравнения, и если оно не имеет решений, то точка соответствует космическому пространству, и её координаты никак не изменяются от кадра к кадру. Таким образом, мы получаем координаты точки на Земной поверхности.
 4. Производится поворот системы координат в соответствии со скоростью летательного аппарата, (такой поворот производится матрицей, являющейся произведением трёх матриц поворота вокруг разных осей).
 5. Новые координаты точки на Земной поверхности проецируются на плоскость матрицы фотоприёмника для нахождения новых координат пикселя в кадре.
 6. Также учтена возможность того, что точка может переместиться из поля зрения одной камеры в поле зрения другой камеры. Пересчёт координат из одной камеры в другую осуществляется поворотом системы OXYZ вокруг оси OX.

При пересчёте координат получаются не целочисленные значения, а для создания карты фона необходимо вычислить яркость точек, соответствующих целым значениям координат пикселей в кадре. При этом не учитывается изменение области, с которой излучение попадает в данный пиксель, и не учитывается изменение яркости в соответствии с диаграммой рассеяния. Это приближенно соответствует Ламбертовскому рассеиванию. Тогда яркость в пикселях с целочисленными координатами вычисляется как интерполяция или аппроксимация яркостей с нецелочисленными координатами. При этом используется не реальные координаты точек, а координаты их изображений на матрице фотоприёмника. Во многом оба подхода эквивалентны, т.к. в малой окрестности каждой точки поверхность Земли можно считать плоской, а тогда все линейные алгоритмы вообще эквивалентны, а нелинейные близки друг к другу. К тому же использование других координат эквивалентно использованию других способов интерполяции/аппроксимации, а как показало изучение, все они дают приблизительно один и тот же результат.

При проведении интерполяции или аппроксимации для каждой точки данного кадра выбирается небольшая окрестность (размером от 1 до 3 пикселей). В эту окрестность попадают точки из других кадров, по которым и проводится интерполяция или аппроксимация. При этом координаты точек в такой окрестности квазислучайны и в силу нелинейности задачи различны для различных точек, поэтому все вычисления надо проводить для каждой точки отдельно.

Были реализованы следующие алгоритмы интерполяции/аппроксимации:

- *Кусочно-постоянная интерполяция.* Яркость пикселя приравнивается к яркости ближайшей к ней точки, что как раз соответствует приближению функции яркости кусочно-постоянной функцией.

- *Линейная интерполяция.* В этом алгоритме выбирается треугольник минимального размера, включающий нужную точку, и яркость этой точки вычисляется как линейная интерполяция по треугольнику. Треугольник выбирается следующим образом. Перебираются все возможные треугольники из точек, удалённых не более чем на полпикселя.

Проверяется, принадлежит ли данная точка каждому из этих треугольников. Это так, если луч, проведённый из этой точки, пересекает стороны треугольника ровно один раз (при этом луч можно выбирать для простоты горизонтально, а далее посмотреть, пересекает ли он каждую из сторон треугольника). Из всех треугольников, включающих данную точку, выбирается минимальный по критерию

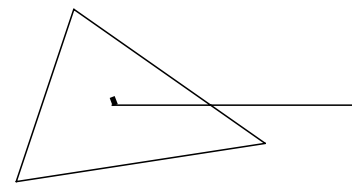


Рис.3

$$J = \sum_{i=1}^3 S_i^2, \text{ где}$$

S_i – расстояние от вершины треугольника до данной точки. Внутри этого треугольника функция интерполируется линейно функцией $b = Ax + By + C$. Коэффициенты находятся из решение методом Гаусса системы уравнений

$$b_i = Ax_i + By_i + C, \quad i = 1 \div 3, \text{ где}$$

b_i, x_i, y_i – значения яркости и координат каждой из вершин.

- *Вычисление среднего.* Яркость точки вычисляется как среднее значение яркостей ближайших точек. Реализовано два варианта – когда берутся только точки, удалённые от данной не более чем на 0.5 пикселей, и не более, чем на 1.5 пикселей.
- *Линейная аппроксимация.* В данном алгоритме используется модель фона, согласно которой яркость в окрестности данной точки есть линейная функция координат $b=a_x x+a_y y+a$. При этом коэффициенты вычисляются методом наименьших квадратов, т.е. решая методом Гаусса систему уравнений

$$\sum_{i=1}^N \begin{bmatrix} x_i^2 & x_i y_i & x_i \\ x_i y_i & y_i^2 & y_i \\ x_i & y_i & N \end{bmatrix} \times \begin{bmatrix} a_x \\ a_y \\ a \end{bmatrix} = \sum_{i=1}^N \begin{bmatrix} x_i b_i \\ y_i b_i \\ b_i \end{bmatrix}, \text{ где } N \text{ – количество точек.}$$

- *Полиномиальная интерполяция.* Яркость данной точки вычисляется путём полиномиальной интерполяции яркостей ближайших точек. При этом степень полинома выбирается так, чтобы количество точек было не меньше количества степеней свободы полинома (в двумерном случае количество степеней свободы равно $p=(n+1)(n+2)/2$, где n – степень полинома). Тогда коэффициенты полинома находятся решением системы, аналогичной системе, возникающей при линейной интерполяции, но с количеством уравнений, равным p .
- *Полиномиальная аппроксимация.* Функция яркости аппроксимируется полиномом заданной степени, по которому вычисляется яркость данной точки. Коэффициенты полинома ищутся методом наименьших квадратов аналогично случаю линейной аппроксимации.
- *Билинейная аппроксимация.* Функция яркости аппроксимируется билинейной функцией, т.е. $b=Axy+Bx+Cy+I$, линейной по каждой из координат. Коэффициенты в этой функции вычисляются фильтром Калмана. Здесь координаты отсчитываются от искомой точки, которая, соответственно, имеет координаты $(0,0)$. Тогда вектором состояния считается величина

$$\mathbf{X} = (b_{0.5,0.5}, b_{0.5,-0.5}, b_{-0.5,0.5}, b_{-0.5,-0.5})^T, \text{ где } b_{x,y} \text{ – яркость в точке } (x,y).$$

Тогда билинейно интерполированная яркость в точке (x,y) будет равна

$$b=a(x,y)^T \mathbf{X},$$

$$a(x,y)=((0.5+x)(0.5+y), (0.5+x)(0.5-y), (0.5-x)(0.5+y), (0.5-x)(0.5-y))^T.$$

Тогда на k -ом шаге в соответствии с формулами для фильтра Калмана производятся операции:

$$K=C(k-1) a(x_k,y_k) / (1+a^T(x_k,y_k) C(k-1) a(x_k,y_k))$$

$$X(k)=X(k-1)+K (b_k-a^T(x_k,y_k) X(k))$$

$$C(k)=C(k-1)-K a^T(x_k,y_k) C(k-1).$$

Начальными условиями здесь являются

$$X(0)=(0,0,0,0)^T, C(0)=diag(10^{12}, 10^{12}, 10^{12}, 10^{12}).$$

Когда эта итерационная процедура проведена для каждой точки, по конечному значению $X(N)$ билинейной интерполяцией вычисляется яркость искомой точки.

При этом необходимо учитывать то, что для некоторых точек, находящихся недалеко от горизонта, все точки из других кадров располагаются точно вдоль линии, и задача вырождается в одномерную, т.е. необходимо иметь одномерные аналоги всех представленных алгоритмов.

Также, если точка является изображением космического пространства, то используется только алгоритм вычисления среднего значения, т.к. тогда никаких смещений подстилающей поверхности (которой являются звёзды и атмосфера Земли на просвет) нет и задача строго стационарна.

Первый алгоритм показывает хорошие результаты только когда ближайшая точка достаточно (<0.1 пикселя) близко к нужной точке. Остальные же показывают примерно одинаковые результаты. Существование предела качества связано, видимо, с тем, что функция яркости подстилающей поверхности кусочно-непрерывна и имеет многочисленные разрывы, которые с трудом поддаются аппроксимации. И СКО остаточного фона связано, прежде всего, с плохой фильтрацией разрывов, т.е. качество алгоритмов ограничено не точностью заложенных в них моделей, а принципиальным ограничением точности оценки, обусловленным разрывностью функции яркости фона.

Для повышения качества можно использовать комбинацию нескольких алгоритмов, при этом в каждой точке выбирается тот алгоритм, который даёт наименьшую разность яркостей с данным кадром. Особенно эффективно использовать комбинацию кусочно-постоянной интерполяции с каким-нибудь другим алгоритмом.

Используя карту фона, построенную на основе модели сферической поверхности можно построить карту рельефа подстилающей поверхности. Особо важно выделить облака, т.к. их высоты достаточно велики для того, чтобы алгоритмы создания карты фона работали

абсолютно неадекватно. При этом область считается однослойной, а рельеф облаков и поверхности Земли не принимается во внимание. При этом не происходит существенной потери качества, т.к. высоты влияют только на пересчёт координат, а ошибка в высоте ~500м даёт несущественную ошибку в определении координат, причём многие алгоритмы аппроксимации/интерполяции практически не используют точные значения координат. Однако же при больших высотах, особенно при большой толщине облачного слоя эта модель может быть неудовлетворительной.

3.4 Определение высоты облачного слоя

Первым этапом определения рельефа является определение высоты облачного слоя. Это делается путём сравнения реального кадра с картами фона, созданными для различных высот подстилающей поверхности над уровнем моря. При этом карта фона создаётся в приближении идеальной сферы.

Т.к. создание карт фона для большого количества высот требует больших вычислительных затрат, то используется итерационный алгоритм:

1. Вычисляется высота с точностью 1км, что требует создание карт фона от высоты поверхности Земли на данной территории, (она считается известной из географических источников) до высоты полёта самолёта с интервалом в 1км.
2. В окрестности этой высоты вычисляются карты фона с интервалов высот в 100м.

Дальнейшее уточнение высоты нецелесообразно по нескольким причинам. Во-первых, толщина облачного слоя вряд ли может быть меньше ста метров, и тогда теряется сам смысл понятие высоты облачного слоя для более высоких точностей. Во-вторых, повышения качества для больших точностей исчезающе мало, а вычислительные затраты весьма велики. Также, для увеличения скорости работы алгоритма можно вычислять высоту облачного слоя не для каждого кадра, а каждого N-ого кадра, где N надо подбирать из физических соображений. Когда карты фона вычислены для каждой из высот, вычисляется кадр корреляции между каждой из них и текущим кадром по формуле

$$C_{kl} = \sum_{\substack{i=k-dx \\ j=l-dy}}^{\substack{i=k+dx \\ j=l+dy}} (M_{ij} - m_{kl})(F_{ij} - f_{kl}), \text{ где}$$

- C_{kl} – точка кадра корреляции.
- M_{ij} – точка кадра карты фона.

- F_{ij} – точка текущего кадра.
- dx и dy – размеры области, для которой вычисляется корреляция (могут быть от 5 до 25 пикселей, оптимальные ~15).

- $m_{kl} = \sum_{\substack{i=k-dx \\ j=l-dy}}^{i=k+dx \\ j=l+dy} M_{ij}$ и $f_{kl} = \sum_{\substack{i=k-dx \\ j=l-dy}}^{i=k+dx \\ j=l+dy} F_{ij}$ -средние значения яркостей карты фона и текущего кадра в данной области.

Такие кадры корреляции создаются для каждой высоты. Далее создаётся карта высот – для каждой точки текущего кадра из всех кадров корреляции выбирается кадр, в котором значение коэффициента корреляции для данного пикселя максимально, и полагается, что высота данного пикселя соответствует максимуму коэффициента корреляции. По карте высот строится гистограмма количества пикселей с данной высотой, максимум которой соответствует искомой высоте облачного слоя.

Несмотря на то, что при работе данного алгоритма создавалась карта высот, использовать её в качестве карты облаков нельзя, поскольку, несмотря на кажущуюся работоспособность данного алгоритма, карта облаков получается с недопустимым количеством ошибок, однако, высота облачного слоя при этом определяется вполне надёжно.

3.5 Создание карты облаков

После вычисления высоты облачного слоя создаётся карта облаков, т.е. кадр, определяющий для каждого пикселя текущего кадра, принадлежит тот облачному слою или Земной поверхности. Было разработано несколько алгоритмов для создания такой карты. Во всех в них используется карта фона для двух высот – для высоты облачного слоя и поверхности Земли.

Однако создание этих карт фона необходимо проводить с учётом более сложной ситуации, чем при сферической поверхности. Необходимо принимать особые меры, чтобы случаи затенения не сказывались существенно на качестве получаемых карт фона. При этом особую проблему составляет то, что облака перемещаются на фоне Земли при изменении угла зрения, т.е. точки поверхности Земли, видные на одном кадре могут быть не видны на другом.

Для решения этой проблемы алгоритмы создания карты фона используются с некоторыми модификациями. Все они заключаются в ограничении размеров временного окна. Т.к. облака и просветы между ними, зачастую, достаточно протяжённые, то можно

ожидать, что точка затеняется или, наоборот, находится вне затенения, достаточно долго. Т.е. если данная точка принадлежит поверхности Земли, то она не затеняется либо во все последующие моменты времени внутри «временного окна», либо во все предыдущие. Тогда при создании карты она можно использовать один и тот же алгоритм, но работающий сначала только со всеми предыдущими кадрами, а потом со всеми последующими. А дальше можно для каждой точки выбрать лучших из алгоритмов. Однако, иногда размеры облаков или просветов малы, и за «временное окно» точка может несколько раз выходить из тени и попадать туда снова. Тогда необходимо вычислять карте фона лишь по ближайшим по времени кадрам. При этом не следует забывать, что чем больше кадров используется при создании карты фона, тем больше её точность.

Для наилучшего результата необходимо использовать различные алгоритмы, модифицированные всеми перечисленными способами, а также и вовсе не модифицированные, а потом для каждой точки выбрать лучший из них путём сравнения с текущим кадром. Далее полученные таким образом карты фона для двух высот используются собственно в алгоритмах создания карты фона.

Таких алгоритмов несколько:

- *Проверка на лучшее совпадение.* Создаётся кадр с разностями между яркостями текущего кадра и карты фона и производится свёртка этого кадра с прямоугольником заданного размера, т.е. производятся следующие вычисления:

$$D_{kl} = \left| \sum_{\substack{i=k-w \\ j=l-w}}^{\substack{i=k+w \\ j=l+w}} (F_{ij} - M_{ij}) \right|, \text{ где}$$

- D_{kl} – пиксель кадра яркости.
- w – величина окна (от 1 до 5).

При этом точка определяется как принадлежащая облакам, если для неё свёртка разности для высоты облаков меньше, чем для высоты поверхности Земли, т.е. если её окрестность более похожа на облака, чем на поверхность Земли. При этом не учитываются точки, для которых это значение недостаточно мало по сравнению с яркостью точки в кадре, т.к. иначе слишком много точек на границе раздела между облаками и Землёй будут идентифицированы неправильно. Этот алгоритм достаточно хорошо обрабатывает облака, однако существует два недостатка. Для крупных облаков в центре появляются ошибки,

связанные с тем, что облака достаточно монотонны и любая область в них достаточно хорошо согласуется с любой другой внутренней областью. Также существенны ошибки на границах раздела.

- *Проверка на лучшую корреляцию.* Считаются кадры корреляции между двумя картами фона и текущим кадром по формуле, аналогичной той, что использовалась в алгоритме определения высоты облаков (при этом размер области, по которой считается корреляция, значительно меньше, чем при определении высоты облаков). Далее создаётся промежуточная карта облаков, в которую попадают точки, для которых лучший из коэффициентов корреляции достаточно мал по сравнению с квадратом яркости этой точки в текущем кадре, т.е., фактически, всё точки, находящиеся достаточно далеко от границы между облаками и поверхностью Земли. Потом на этой промежуточной карте облаков выбираются связанные области, и для каждой из них определяется их принадлежность к облачному слою методом, аналогичным первому алгоритму. Этот алгоритм хорошо выявляет лишь внутренние области больших облаков, совершенно «не замечая» при этом мелких.
- *Проверка на принадлежность поверхности Земли.* Текущий кадр обеляется на основе карты фона для высоты поверхности Земли. И точка считается принадлежащей облакам, если яркость пикселя обелённого кадра недостаточно мала по сравнению с яркостью пикселя исходного кадра, т.е. если она плохо отфильтрована. Этот алгоритм также как и первый не обрабатывает внутренние области облаков, однако достаточно хорошо выявляет их границы.
- *Проверка на принадлежность облакам.* Текущий кадр обеляется на основе карты фона для высоты облаков. И точка считается принадлежащей облакам, если яркость пикселя обелённого кадра мала по сравнению с СКО этого кадра. Этот алгоритм выявляет далеко не все точки, принадлежащие облакам, однако в качестве дополнения к остальным может быть весьма полезен.

Эти алгоритмы имеют довольно замысловатую структуру, однако упростить её, как показывает опыт, нельзя. Все алгоритмы, основанные, например, просто на корреляции между областями (без дополнительных ухищрений), неработоспособны.

Однако ни один из этих алгоритмов по отдельности не показывает удовлетворительных результатов. Для достижения хорошего качества необходимо

объединить результаты их работы, предварительно удалив с карт облаков все мелкие детали. Также удобно работать в приближении, что физически облака не сдвигаются от кадра к кадру и не деформируются. Тогда карты облаков для разных моментов времени одинаковы с точностью до изменения угла зрения. Однако, изменение кадра при изменении угла зрения известно (и использовалось уже при создании карты фона), и можно сравнивать текущую карту облаков с картами облаков в предыдущие моменты. И если данная точка была признана как принадлежащая облакам достаточное число раз в этих кадрах (включая текущий), то она признаётся принадлежащей облакам и в текущем кадре.

В результате получается достаточно качественная карта облаков. Однако существует существенный недостаток. Во всех алгоритмах создания карты фона используются различные параметры, значения которых удалось определить лишь на основе качества работы на тестовых последовательностях. И при изменении условий наблюдения, возможно, потребуется изменить некоторые из этих параметров.

3.6 Обеление кадра и пространственная фильтрация

На основе данной облаков фона создаёт карта фона с учётом высоты каждой точки со всеми возможными затенениями. Т.е. для точек с поверхности Земли при создании карты фона используются только те точки с других кадров, в которых данная точка не была закрыта облаками.

При этом необходимо учитывать возможные ошибки в алгоритме создания карты облаков. Они имеют место на границе облаков, но, т.к. граница перемещается в поле зрения со скоростью, отличной от скорости перемещения точек на поверхности Земли, имеет место эффект размазывания ошибок в яркости карты фона по большой площади. Для избежания этого можно в алгоритме создания карты фона при пересчёте координат точек с одного кадра на другой игнорировать точки, находящиеся вблизи границы облаков. При этом из поля зрения пропадает некоторая часть полезной площади, но улучшается качество алгоритма фильтрации.

Созданная таким образом карта фона используется для обеления кадра. Кадр обелается достаточно качественно в точках непрерывности функции яркости и не на границе раздела между облаками и поверхностью Земли. Для борьбы с этими остаточными помехами целесообразно использовать пространственную фильтрацию, т.к. они сильно коррелированы в пространстве. При пространственной фильтрации яркость пикселя обелённого кадр

получается вычитанием из пикселя исходного кадра линейной комбинации яркостей соседних пикселей по формуле

$$b_{i,j} = \sum_{n=-p}^p \sum_{m=-p}^p b_{i+n,j+m} H_{n,m}$$

Использовались следующие фильтры:

- Матрица 3x3 порядка 0

$$H_{nm} = \frac{1}{8} \begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix}$$

- Крест 3x3 порядка 0

$$H_{nm} = \frac{1}{4} \begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix}$$

- Матрица 3x3 порядка 2

$$H_{nm} = \frac{1}{4} \begin{vmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{vmatrix}$$

- Матрица 5x5 порядка 0

$$H_{nm} = \frac{1}{24} \begin{vmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{vmatrix}$$

- Матрица 5x5 порядка 2

$$H_{nm} = \frac{1}{1036} \begin{vmatrix} 91 & -14 & -49 & -14 & 91 \\ -14 & -119 & -154 & -119 & -14 \\ -49 & -154 & 1036 & -154 & -49 \\ -14 & -119 & -154 & -119 & -14 \\ 91 & -14 & -49 & -14 & 91 \end{vmatrix}$$

- Матрица 5x5 порядка 4

$$H_{nm} = \frac{1}{684} \begin{vmatrix} -51 & 99 & -96 & 99 & -51 \\ 99 & 24 & -246 & 24 & 99 \\ -96 & -246 & 684 & -246 & -96 \\ 99 & 24 & -246 & 24 & 99 \\ -51 & 99 & -96 & 99 & -51 \end{vmatrix}$$

Наилучшие результаты показал крестовой фильтр нулевого порядка. Это связано с тем, что остаточные помехи представляют собой в основном линии толщиной в 1 пиксель, и использование более сложных фильтров приводит лишь к их размазыванию.

4. Анализ и результаты работы алгоритма

Т.к. различных параметров в алгоритме достаточно много, каждый шаг работы алгоритма исследовался отдельно.

При исследовании использовались следующие параметры:

1. Размер исходных кадров 100x200 пикселей.
2. Изображение получалось с 4-х камер.
3. Яркость представлена в Вт/см² в формате double.
4. Размер «временного окна» - $t_{pr}=t_{nx}=6$.

4.1 Анализ алгоритма создания карты фона

Для исследования алгоритма создания карты фона использовались последовательности без облачного слоя. При этом моделировалась как сферическая Земля, так и небольшой рельеф с перепадом высот до 600 метров. Кадр этой последовательности и типичный результат его обработки представлен на рисунке 4.

Основным критерием качества работы алгоритмов является коэффициент подавления фона, т.е. отношение СКО исходного кадра к СКО обелённого. В таблице 1 представлены результаты работы всех алгоритмов.

Таблица 1	Идеальная сферическая поверхность	Рельеф ~600 м
Кусочно-постоянная интерполяция	12 ÷ 14	12 ÷ 14

Вычисление среднего	17÷19	15÷17
Линейная интерполяция	16÷17	14÷17
Линейная аппроксимация	18÷19	17÷18
Полиномиальная интерполяция	12÷13	11÷12
Полиномиальная аппроксимация	12÷14	12÷13
Билинейная аппроксимация	17÷18	17÷18
Комбинация кусочно-постоянной интерполяции и линейной аппроксимации	20÷22	20÷22

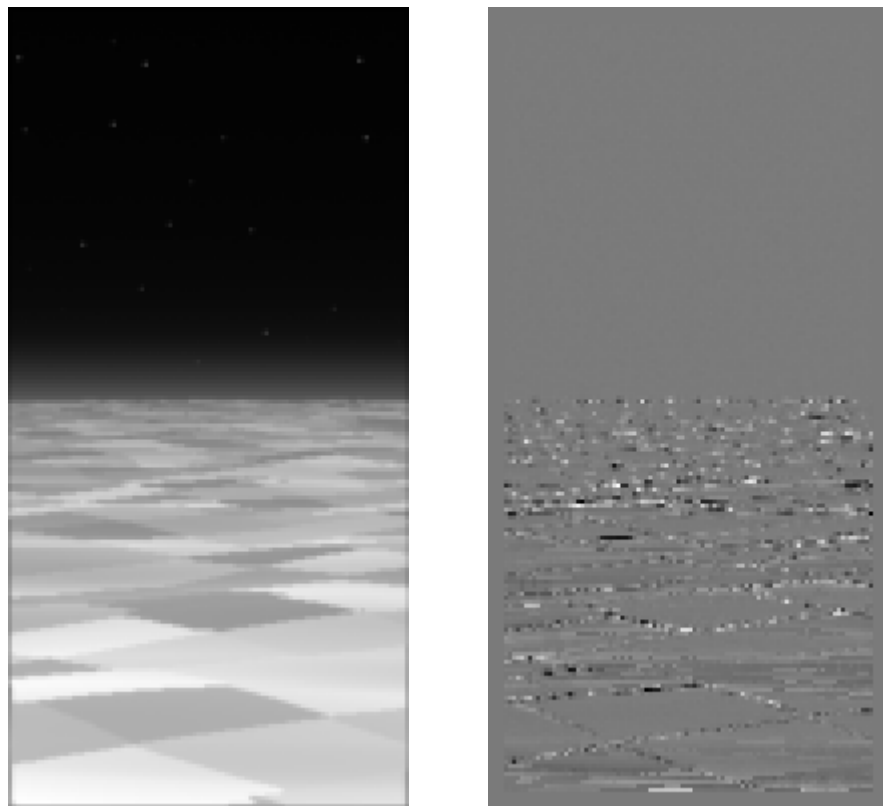


Рис.4

Из таблицы ясно, что все алгоритмы показывают близкие результаты, что можно объяснить существованием теоретического предела, связанного с большим количеством разрывов функции яркости. Из Рис.4 видно, что максимальная интенсивность остаточных помех находится вблизи линий разрыва.

Самым качественным алгоритмом является комбинация кусочно-постоянной интерполяции и линейной аппроксимации. Он в основном и используется на последующих шагах работы алгоритма.

4.2 Анализ алгоритма определения высоты облачного слоя

Для исследования точности определения высоты облачного слоя использовались две последовательности с идентичной картой облаков, но с различными вариантами освещения. Кадры из этих последовательностей представлены на Рис.5.

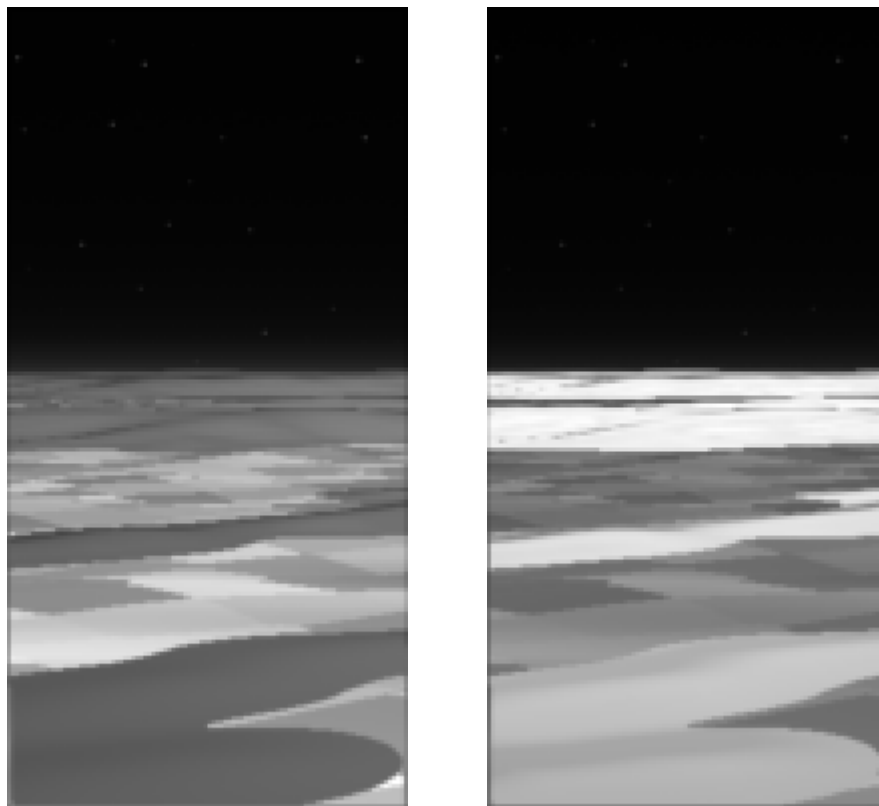


Рис.5

Высота облачного слоя – 6 км при высоте полёта – 9 км, толщина около 1 км. Гистограммы распределения точек по высотам для первого (точность – 1км) и второго (точность – 100м) шагов работы итерационного алгоритма определения высоты облачного слоя представлены на Рис.6.

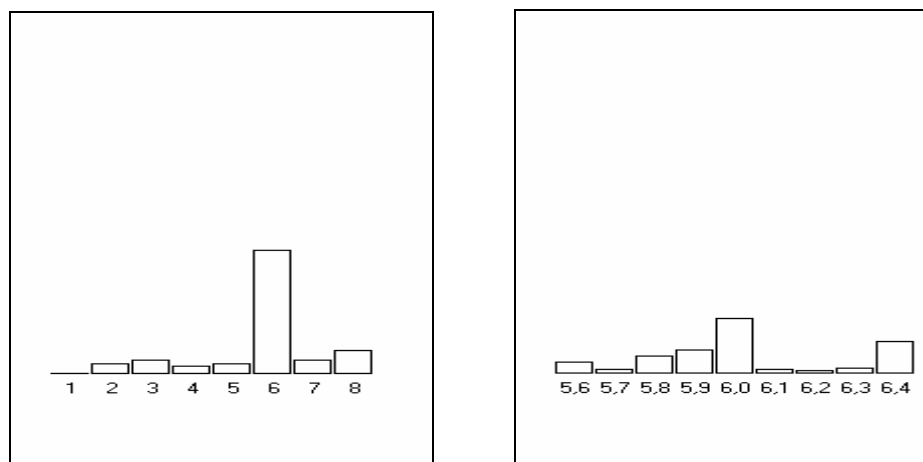


Рис.6

Как уже отмечалось, получаемая в этом алгоритме карта высот крайне плохо соответствует действительности, что может навести на подозрение о плохом качестве работы данного алгоритма. Однако во всех протестированных последовательностях гистограммы имели такой же вид, т.е. высота не только определяется, но и определяется устойчиво с заданной точностью.

Рельеф земной поверхности вплоть до 1 км вообще никак не сказывается на работе данного алгоритма.

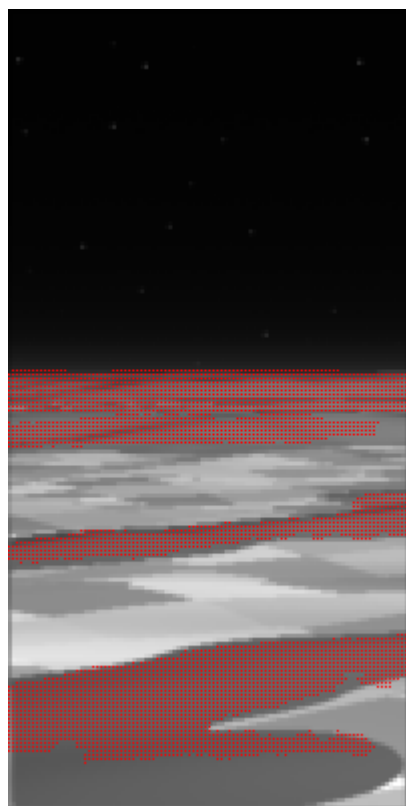
4.3 Анализ алгоритма создания карты фона

Алгоритм создания карты облачного слоя тестировался на тех же последовательностях. При этом исследовалось как качество работы отдельных алгоритмов, так и всей их совокупности.

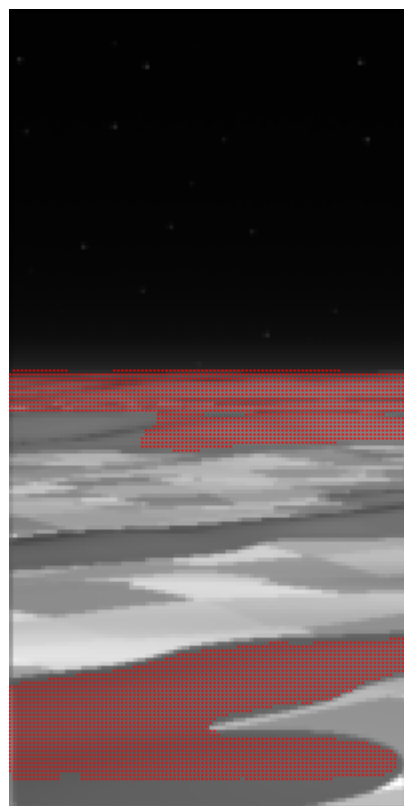
Типичные результаты работы отдельных алгоритмов показаны на Рис.7. Красным показана получившаяся карта облаков. Как видно ни один из этих алгоритмов не показывает идеальных результатов.

Алгоритм проверки на лучшее совпадение (Рис.7а) работает лучше всего, однако плохо обрабатывает края облаков. Также существенно, что в некоторых случаях он пропускает центральные части больших облаков, а иногда выходит за границы облаков на поверхность Земли.

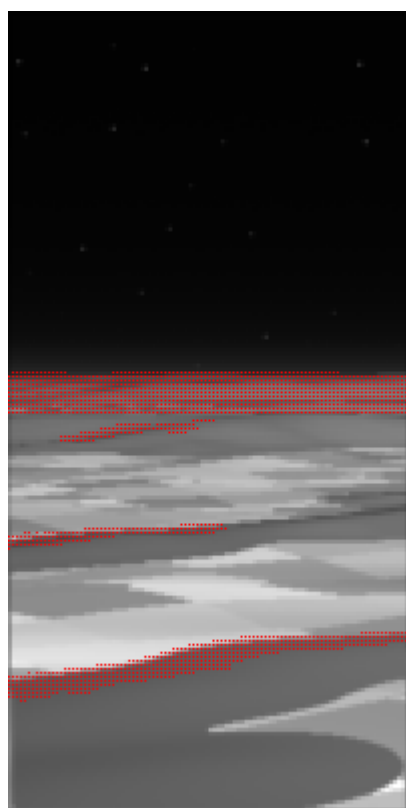
Алгоритм проверки на лучшую корреляцию (Рис.7б) пропускает мелкие облака, при этом с крупными работает очень хорошо и устойчиво.



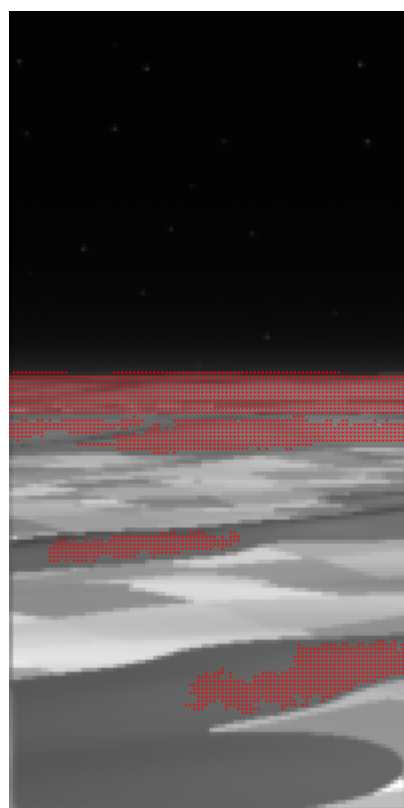
а



б



в



г

Рис.7

Алгоритм проверки на принадлежность Земле (Рис.7в) вообще трудно назвать алгоритмом по составлению карты облачного слоя, однако он бывает полезен вблизи границ облаков, учитывая, что оба предыдущих алгоритма не вполне удовлетворительно работало с ними. Эта особенность его работы связана с тем, что облака имеют достаточно равномерную яркость и вдали от границ фильтруются зачастую качественно, в то время, как разрывы на поверхности Земли по-прежнему недостаточно хорошо обрабатываются.

Алгоритм на проверку принадлежности к облакам (Рис.7г) с большой степенью детализации обрабатывает пригоризонтные области.

Комбинация всех этих алгоритмов, с усреднением по времени даёт уже гораздо более качественный результат, представленный на Рис.8

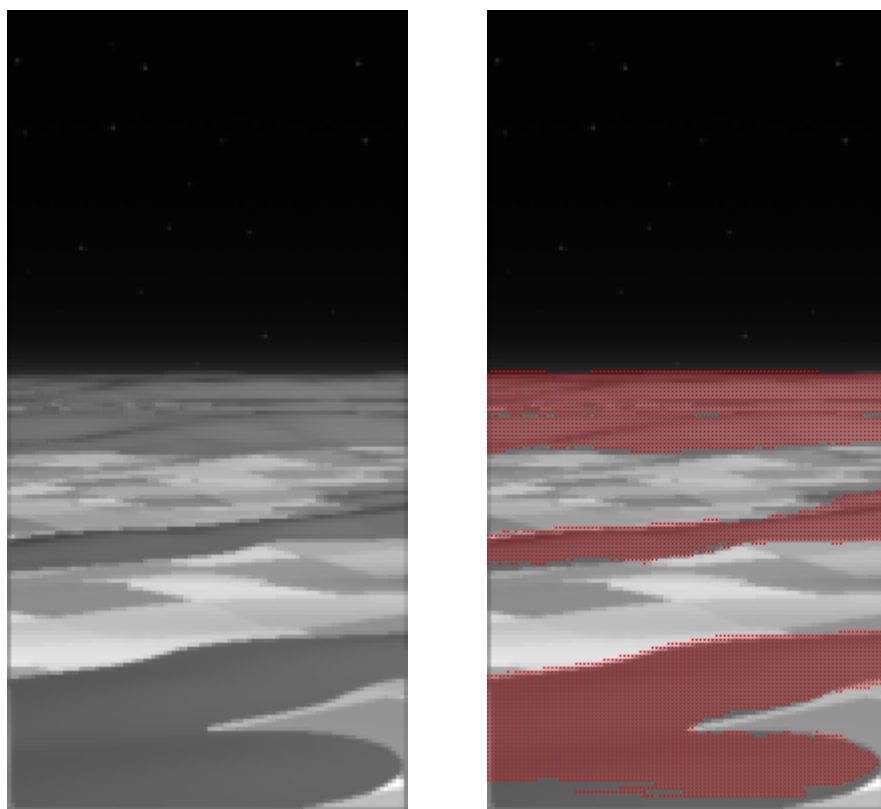


Рис.8

Однако следует заметить, что и здесь существуют ошибки. Во-первых, иногда граница облаков вычисляется с ошибками ~ 1 пикселя. Также в нижней части кадра существует область, которая за счёт краевых эффектов практически не обрабатывается.

Особую проблему в алгоритме определения карты облаков составляет настройка этих алгоритмов, т.е. задание параметров, используемых в этих алгоритмах. Никакой аналитической оценки этих параметров получить не удалось. Поэтому единственной возможностью их определения является обучение с учителем.

Однако удалось добиться того, что они безразмерны и не зависят от простых преобразований входящих кадров. При этом оптимальные параметры практически одинаковы для всех исследованных последовательностей, что позволяет надеяться на их применимость в достаточно широких условиях.

4.4 Анализ комплексного итерационного алгоритма фильтрации

Следующим шагом алгоритма является создание карты фона с учётом рельефа, и собственно обеление кадра. Здесь существуют два подхода. Помимо прямого создания карты фона можно сначала исключить из рассмотрения точки, лежащие на границе облаков. Это приведёт к тому, что потеряется какая-то часть кадра, однако при этом улучшится качество фильтрации оставшейся части.

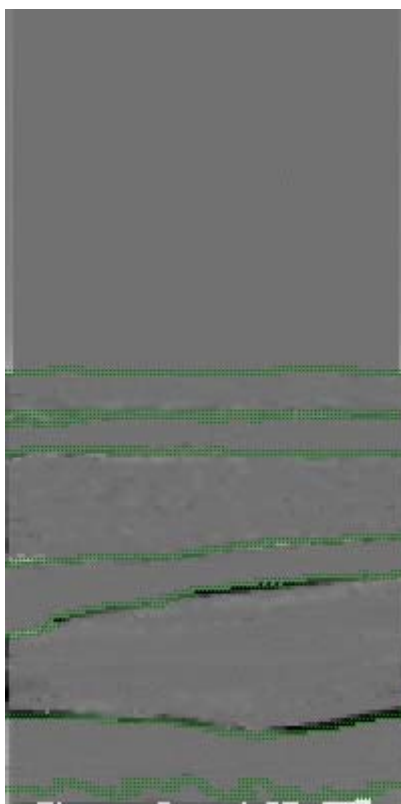
Результаты работы представлены на Рис.9 и в таблице 2.



а



б



в

Рис.9

На Рис.9а показан исходный кадр, на Рис.9б – результат его обработки без исключения границ облаков, а на Рис.9в – с исключением этих границ (границы указаны зелёным).

Таблица 2	Без удаления границ облаков	С удалением границ облаков
Кусочно-постоянная интерполяция	10÷12	11÷12
Вычисление среднего	15÷17	16÷17
Линейная интерполяция	16÷18	17÷18
Линейная аппроксимация	16÷17	17÷19
Полиномиальная интерполяция	10÷12	10÷13

Полиномиальная аппроксимация	10÷12	11÷13
Билинейная аппроксимация	16÷18	18÷19
Комбинация кусочно-постоянной интерполяции и линейной аппроксимации	16÷18	18÷20

Как видно, качество фильтрации падает не сильно по сравнению с идеальной сферической Землёй. Причём существенная часть отметок возникает именно на границе облаков, которую можно исключить из рассмотрения.

4.5 Анализ алгоритма пространственной фильтрации

Последним этапом алгоритма является пространственная фильтрация. Она применяется к уже обелённому кадру. Результаты её работы показаны на Рис.10 и в таблице 3 (здесь указан коэффициент подавления фона по отношению к уже обелённому кадру).

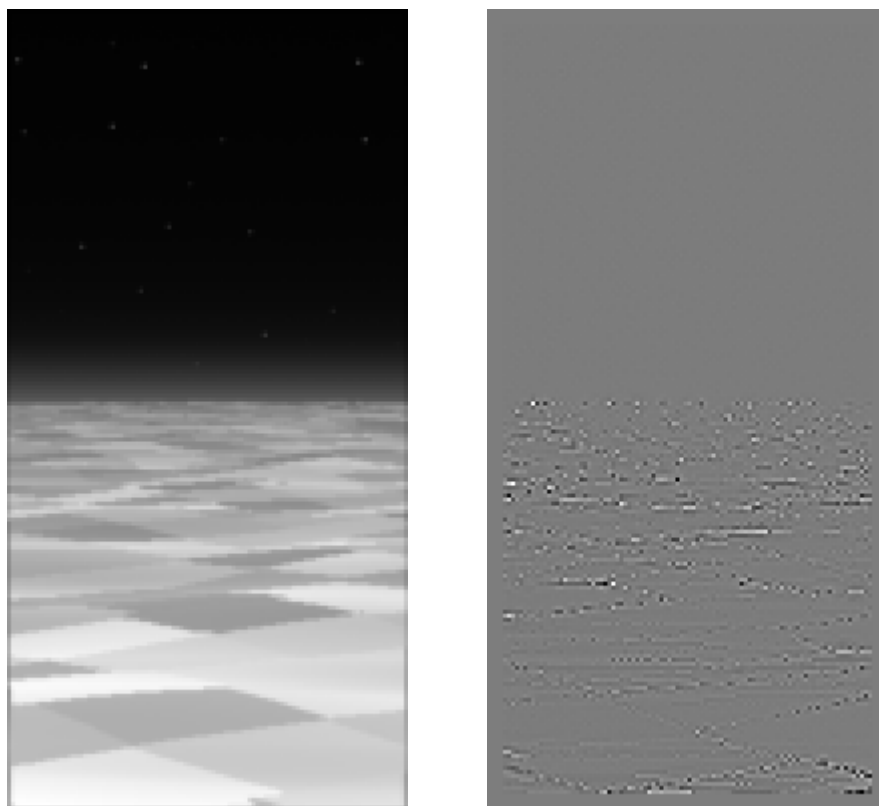


Рис.10

Таблица 3	Без облаков	С облаками
Матрица 3x3 порядка 0	1.3	1.4
Крест 3x3 порядка 0	1.8	2.0
Матрица 3x3 порядка 2	1.02	1.05
Матрица 5x5 порядка 0	1.2	1.1
Матрица 5x5 порядка 2	1.05	0.9
Матрица 5x5 порядка 4	0.9	1.1

Помимо собственно подавления шума существенно также снижение корреляции остаточного шума по пространству и по времени.

4.6 Итоговые результаты работы алгоритма

В результате анализа наилучшие результаты показал комплексный алгоритм, в котором при создании карты фона используется комбинация кусочно-постоянной интерполяции и линейной аппроксимации с игнорированием границ облаков, при создании карты

Последовательность	Отношение СКО исходного кадра к СКО обелённого
Без облаков и без рельефа	35÷40
Без облаков и с рельефом ~600м	33÷38
С тёмными облаками	26÷31
Со светлыми облаками	26÷31
Таблица 4	

облаков – все 4 представленных алгоритма с последующим усреднением по времени, в

качестве алгоритма пространственной фильтрации – схема «Крест» нулевого порядка. Результаты работы этого алгоритма на различных тестовых последовательностях представлены в таблице 4.

Однако не следует забывать, что этот вариант алгоритма был выбран для некоторых конкретных последовательностей, в иных же случаях другие варианты алгоритмов могут показать наилучшие результаты.

5. Особенности программная реализация

Программная реализация представленного алгоритма была выполнена я языке Delphi версии 5.0 под операционную систему Windows. При этом интерфейс программы (Рис.11) позволяет видеть результаты работы как отдельных блоков алгоритма, там и всего алгоритма в целом.

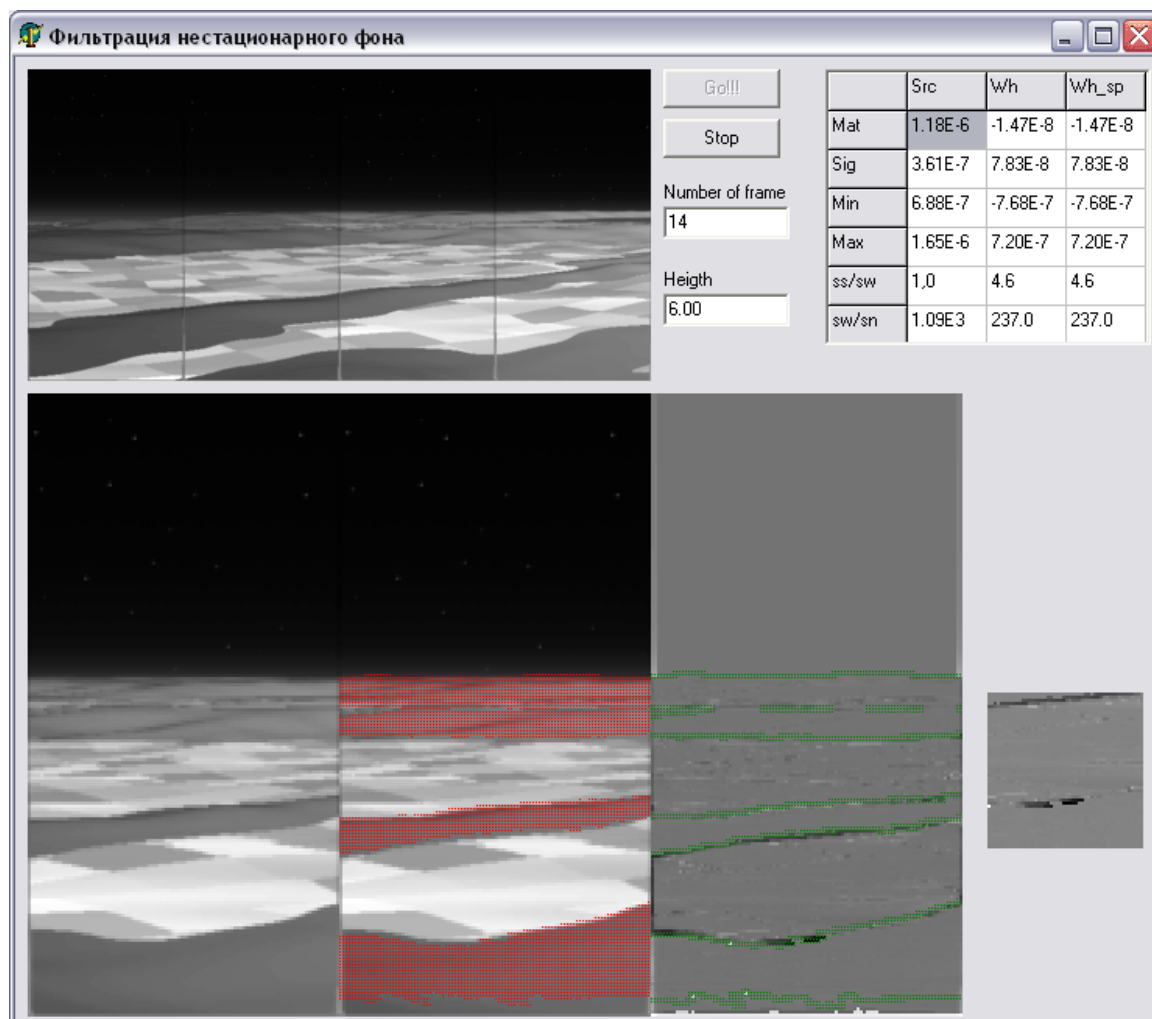


Рис.11

Блок схема программы представлена на Рис.12.

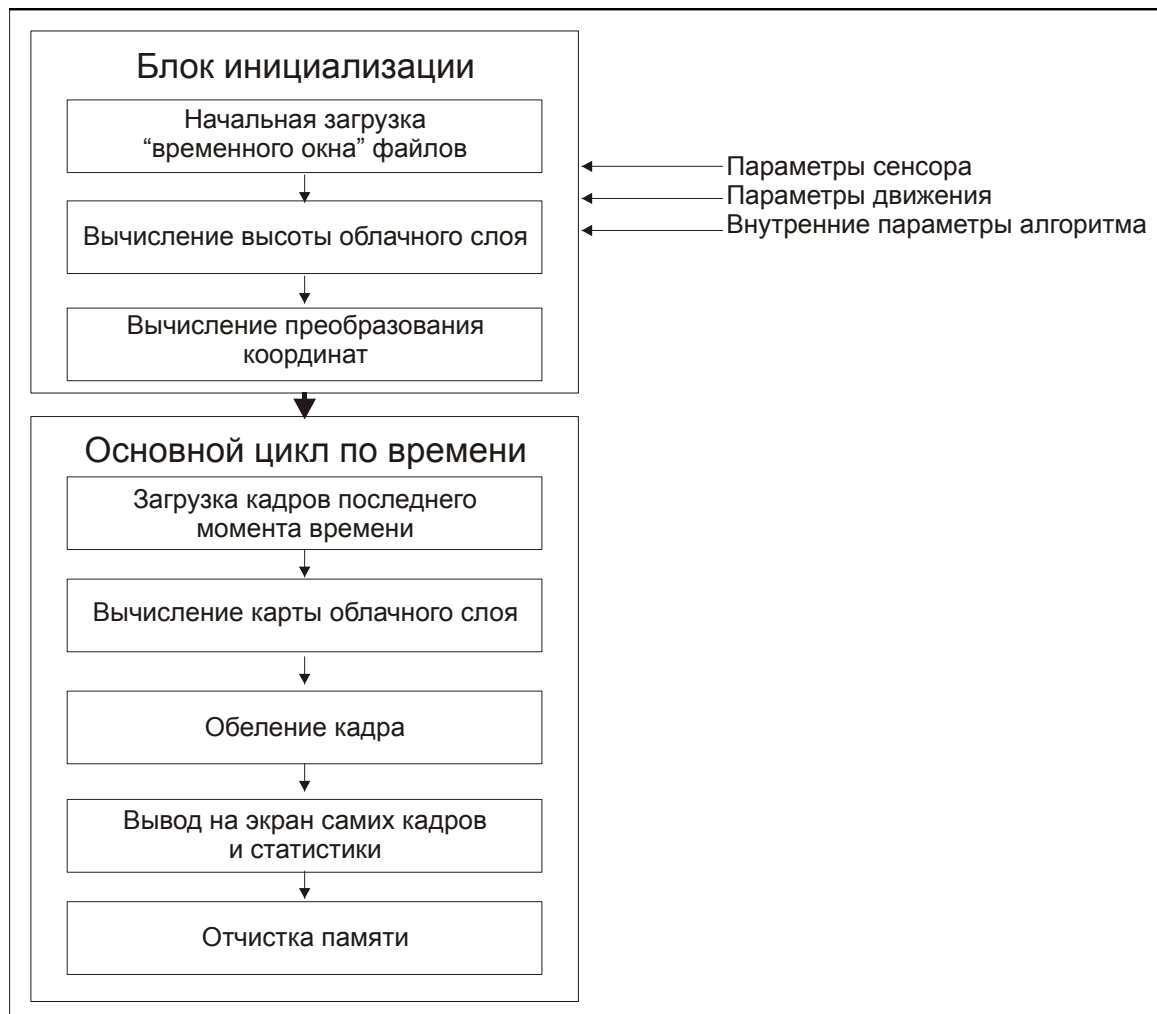


Рис.12

При создании программы особое внимание уделялось скорости работы. Для оптимизации алгоритма были предприняты следующие шаги, позволившие уменьшить время выполнения на порядки:

1. Пересчёт координат можно проводить лишь один раз, т.к. он не меняется со временем. При этом необходимо помимо координат сохранять также номер кадра и координату пикселя в кадре, для которого эти преобразования были выполнены. Тогда для каждого момента времени надо лишь выбрать нужные яркости из новых кадров, а не производить все вычисления заново. Это преобразование также используется во многих других блоках алгоритма без дополнительных вычислений.
2. После пересчёта координат получается набор точек с квазислучайными координатами, которые никак не упорядочены. Тогда поиск нужной точки

среде всех может потребовать очень больших вычислительных затрат. Для ускорения работы точки сортируются в двумерный массив, размеры которого совпадают с размерами кадра. Элемент массива представляет собой список всех точек, координаты которых лежат в окрестности данной точки. Тогда все алгоритмы интерполяции и аппроксимации для каждой точки исходного кадра работают только с одним элементом такого массива (размер которого менее 30 точек), а не со всеми точками (которых более 10^5).

Быстродействие алгоритма протестировано на машине со следующими характеристиками:

- процессор - Athlon 2100+
- память – 256М
- видео карта – GeForce4 MX 440 32М
- операционная система – Windows XP

Времена выполнения каждого из блоков алгоритма представлены в таблице 5.

Блок алгоритма	Время выполнения
Определение высоты облачного слоя	7.9 с
Пересчёт координат	2.7 с
Определение карты облаков	0.8 с
Обеление кадра	0.098 с
<u>Полное время блока инициализации</u>	10.1 с
<u>Полное время одной итерации цикла</u>	0.98 с
Таблица 5	

6. Выводы

Задачей являлась разработка алгоритма фильтрации фона для следующей системы. С самолёта, летящего на высоте 7-10 км, производится наблюдение на фоне подстилающей поверхности и облаков. На самолёте установлены несколько камер, наблюдающих поверхность Земли и облака в направлении горизонта. Наблюдение ведётся в инфракрасном диапазоне. Задачей такого наблюдения является обнаружение точечных движущихся объектов.

Особенностями данной задачи являются:

1. Нестационарность мешающего фона.
2. Нелинейность задачи, связанная со сферичностью Земли.
3. Сложная физическая картина фона. Учитывалось, что фон может составлять поверхность Земли с неизвестным рельефом, облачный слой неизвестной структуры и т.д.
4. Наличие разрывов функции яркости.
5. Наличие неограниченных производных некоторых функций, т.к. наблюдается области горизонта.

Был создан целый спектр похожих алгоритмов, дающих различные результаты для разных внешних условий, что позволяет выбрать лучший из них и обеспечивает устойчивость качества результата в разных ситуациях.

При этом были созданы различные алгоритмы, близкие к оптимальным, а существование одинакового предела качества их работы позволяет утверждать, что он ограничен постановкой задачи, а не оптимальностью самих алгоритмов.

Результаты работы алгоритма характеризуются следующими основными параметрами:

- для случая безоблачной атмосферы коэффициент подавления фона достигает 35 - 40,
- при наличии облаков – 25 – 30,
- при этом остаточный фон лишь в 3 – 5 раз превышает уровень шумов фотоприёмника.

Помимо базового комплексного алгоритма фильтрации были созданы следующие вспомогательные алгоритмы:

1. Алгоритм определения высоты облачного слоя исключительно по данным его наблюдения с летящего самолёта при полном отсутствии априорных данных о её величине.
2. Алгоритм построения карты облачного слоя по тем же данным.

Т.е. фактически реализованы алгоритмы построения трёхмерной картины по данным двумерных наблюдений. Особо следует подчеркнуть, что большинство существующих алгоритмов подобного типа работают при больших изменениях ракурса наблюдения и на основе изучения поведения реперных точек. Разработанные здесь алгоритмы работают при малых углах изменения ракурса и не могут основываться на поведении реперных строчек, т.к. основной задачей данного алгоритма является селекция движущихся точечных целей.

Данный алгоритм был программно реализован. Созданная программа позволяет исследовать работу алгоритма на всех шагах в удобном интерфейсе с выводом всей необходимой информации на экран. Скорость работы алгоритма оптимизирована настолько, что на современном персональном компьютере может работать в реальном времени.

7. Литература

1. **Стратонович Р.Л.** Принципы адаптивного приёма. – «Советское радио», 1973.
2. **Репин В.Г., Тартаковский Г.П.** Статистический синтез при априорной неопределённости и адаптация информационных систем. - «Советское радио», 1977.
3. **Браммер К., Зиффлинг Г.** Фильтр Калмана-Бьюси. –пер. с англ. «Наука», 1982.
4. **Фельдман Ю.И, Гидапсов Ю.Б., Гомзин В.Н.** Сопровождение движущихся целей. – «Советское радио», 1978.
5. **Федоренко Р.П.** Введение в вычислительную физику. – Изд. МФТИ, 1994.
6. **Бутиков Е.И.** Оптика. - «Высшая школа», 1986.
7. **Нейбургер М., Эдингер Д., Боннер У.** Познание окружающей нас атмосферы. – пер. с англ., «Знание», 1985.
8. **Карасик В.Е., Орлов В.М.** Лазерные системы видения. – Изд. МГТУ им. Баумана, 2001.
9. Справочник по инфракрасной технике под ред. Волф У., Цисис Г. – пер. с англ., «Мир», 1995.
10. **Daniilidis K.** Fixation simplifies 3D motion estimation, - Computer visio and image understanding, 1997 No.2.
11. **Jebara T., Azarbayejani A., Pentland A.** 3D structure from 2D motion, - IEEE Signal processing magazine, May 1999/