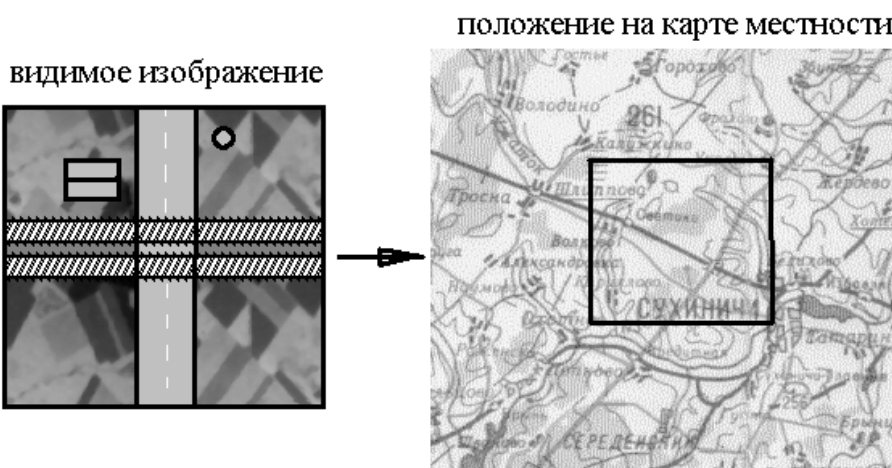
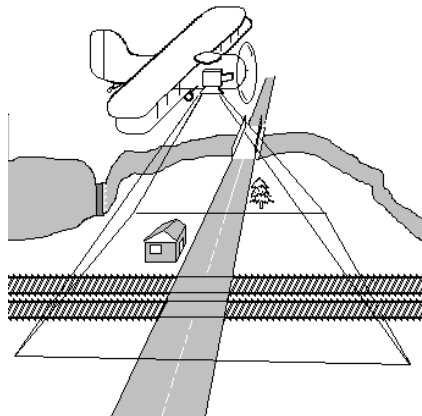


Лабораторная работа № 5

Поиск объекта методами Обобщенного преобразования Хафа.

Теория

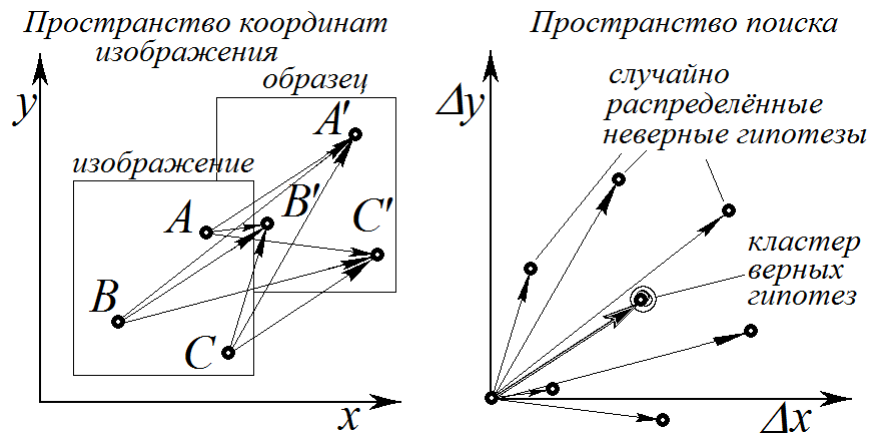
Рассмотрим ту же задачу, что и в предыдущей работе. Пусть имеется БПЛА, оснащенный видеокамерой. Сравнивая получаемое изображение с электронной бортовой картой, нужно определить положение кадра на карте.



В данной работе мы рассмотрим поиск положения кадра на карте с использованием обобщенного преобразования Хафа.

Обобщенное преобразование Хафа преобразует точки пространства изображения в новое пространство (пространство параметров поиска, ПП) таким образом, что все точки объекта будут стянуты в одну точку. Аналог яркости в этой точке называется *откликом* объекта, он равен количеству точек объекта на изображении, совпадающих с образцом, а координаты точки в ПП будут соответствовать координатам объекта на изображении. Тогда задача поиска объекта будет сведена к гораздо более простой задаче поиска максимального отклика в ПП.

Обобщенное преобразование Хафа можно описать следующим образом. Пусть имеется образец S , заданный в виде списка точек (A, B, C, \dots) . Это могут быть, например, точки границ объекта, положения углов, областей высокой кривизны и т.п. Пусть также имеется изображение I , также представленное в виде списка точек (A', B', C', \dots) . Эти точки могут принадлежать объекту, а могут и не принадлежать. Будем считать, что для каждой точки изображения можно указать точки объекта, которые могут ей соответствовать. Соответствие точек определяется по приближенному равенству их параметров – яркости, направления контура и радиуса его кривизны, и т.д.



Каждая пара соответствующих точек объекта и изображения создает одну гипотезу H_{ij} о совпадении этих точек, и соответственно, о положении объекта на изображении:

$$H_{ij} = (\Delta x, \Delta y) = (x_i - x'_j, y_i - y'_j).$$

Если перенести все возможные гипотезы в пространство поиска $H(\Delta x, \Delta y)$, то все верные гипотезы совпадут, образуя суммарный отклик $H^*(\Delta x^*, \Delta y^*)$, а неверные гипотезы будут распределены хаотически беспорядочно, не создавая существенных максимумов. Величина отклика H^* равна количеству совпавших точек, а координаты отклика $(\Delta x^*, \Delta y^*)$ будут равны искомому положению объекта на изображении (рис. 4.10).

Для вычисления положения максимальных откликов представим ПП в виде двумерного массива счетчиков $H[x, y]$. Будем последовательно рассматривать все возможные гипотезы, и для каждой из них будем увеличивать соответствующий счетчик.

По окончании перебора гипотез найдем локальные максимумы в массиве $H[x, y]$, положение которых будет соответствовать наиболее вероятным положениям объекта на изображении.

В качестве опорных точек для генерации гипотез, как правило, используются геометрические признаки (например, точки границ). Использование таких точек дает стратегии с ГНТ низкую чувствительность к изменениям условий видимости объекта, поскольку условия видимости практически не влияют ни на координаты опорных точек, ни на вероятность их обнаружения.

Стратегия с ГНТ позволяет учитывать дополнительные параметры точек (направление градиента, радиус кривизны контура и т.п.), что делает ее гораздо более надежной, нежели методы, основанные на расчете корреляции (при использовании одинаковых наборов признаков, например – точек границ).

Вместе с тем, стратегия весьма чувствительна к изменениям ракурса съёмки (к наличию неучтенных геометрических искажений). При изменении ракурса, верные гипотезы более не будут попадать в одну общую точку в ПП, а будут размазаны по некоторой окрестности достаточно нерегулярным и существенно негауссовым образом, который зависит как от формы самого объекта, так и от характера изменения ракурса.

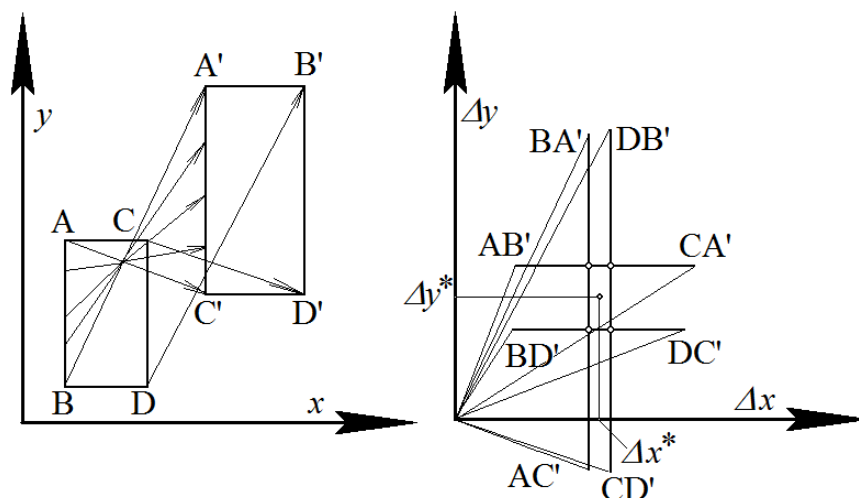
Размытие отклика приводит к существенному снижению его величины: например, если максимум размылся всего на четыре соседних точки, его величина может снизиться в четыре раза. Это может привести к тому, что отклик объекта будет потерян в откликах помех.

Вместе с тем существует возможность преодолеть либо существенно скомпенсировать этот недостаток путём дополнительной обработки ПП.

Стратегия с ГНТ также нечувствительна к отсутствию довольно существенного количества точек объекта: отклик в ПП всё равно будет равен количеству оставшихся точек, сколько бы мало их не было.

Специфической особенностью ГНТ является размытие отклика в ПП при наличии геометрических искажений объекта. Размытие, как отмечалось выше, имеет сложную форму, следствием чего является несовпадение нового локального максимума с положением центра пятна $(\Delta x^*, \Delta y^*)$. Поэтому, помимо уменьшения величины отклика, что само по себе снижает надежность

обнаружения объекта, происходит также и смещение его положения относительно реального положения.



Из рисунка видно, почему возникает смещение положения максимума: наибольший отклик будет соответствовать совпадению одного из углов прямоугольников, тогда как правильное положение $(\Delta x^*, \Delta y^*)$ вообще не даст ни одного отклика.

Таким образом, при наличии геометрических искажений, нескомпенсированных предобработкой и самим процессом поиска, требуется дополнительная обработка ПП для восстановления положения и величины отклика правильных гипотез. Это необходимо как для повышения надежности, так и для повышения точности.

При наличии геометрических искажений найти положение объекта, при котором точки его границ совпадут с границами изображения, будет невозможно. Соответственно, в пространстве поисков верные гипотезы уже не будут совпадать, а будут занимать некоторую небольшую область – кластер верных гипотез. Величина максимума H^* при этом существенно уменьшается и может быть потеряна в шуме. Более того, максимум может размываться кольцеобразно, так, что в центре кластера верных гипотез, соответствующему наиболее вероятному положению объекта на изображении, не окажется сколько-нибудь значимых величин H^* .

Для того чтобы устранить этот недостаток, нужно восстанавливать значение и положение исходного, неискаженного максимума. Этот этап будем называть коррекцией геометрических искажений, поскольку будем устранять последствия изменения ракурса съемки.

Для восстановления максимума можно использовать, например, суммирование откликов по некоторому окну. Размеры окна оцениваются исходя из предполагаемого размытия максимума, вносимого геометрическими искажениями. В этом случае положение максимума будет восстановлено, а его значение снова поднимется над уровнем шума, и сделает возможным надежное обнаружение объекта на изображении.

Алгоритм суммирования по окну требует достаточно небольшого количества операций (четыре сложения на точку), поскольку его можно реализовать за два прохода по изображению.

Ход работы

Часть 1 – Понятие о GHT. GHT – это просто!

В качестве карту загрузим уже известное нам изображение `'ConcordOrthoPhoto.png'`, а в качестве кадра – `'WestConcordOrthoPhoto.png'`.

Для поиска объекта создадим новую функцию, которая будет строить пространство поиска по заданным изображениям кадра и карты.

Найдем точки контура изображений (функция `edge()`), зададим пространство поиска в виде пустого массива (`zeros`) того же размера, что и карта.

Выпишем точки границ кадра и карты в список.

Будем перебирать все пары точек кадра и карты из списка, для каждой пары найдем гипотезу (`dx,dy`) о положении кадра на карте, и проголосуем за эту гипотезу в пространстве поиска.

Запустите функцию из скрипта. Выведите на экран результат ее работы. Замерьте время расчета (`tic,toc`). Найдите положение максимального отклика. Отобразите кадр в найденном положении.

Правильно ли найдено положение объекта?

Сколько времени занял поиск положения?

Можно ли использовать данный алгоритм в реальном времени?

Рассмотрите дуальное пространство. Какую корреляционную функцию оно напоминает?

Подумайте, как можно ускорить алгоритм.

Часть 2 – Преобразование Хафа с использованием направления градиента.

Легко убедиться, что полученное в первой части пространство поиска – это контурная корреляционная функция, но рассчитанная не напрямую и не через спектр, а через GHT. Однако этот способ требует больше времени, чем с использованием спектра.

Создайте новую функцию, скопируйте в нее текст предыдущей функции, сохраните под новым именем.

Преобразование можно ускорить, если перебирать не все точки, а лишь те, которые имеют примерно одинаковое направление контура. Тогда большая часть пар будет отброшена.

Как правильно отбросить эти пары? Простой, логичный и совершенно неправильный способ – это проверять градиенты в каждой паре: в этом случае операция инкремента в дуальном пространстве заменится операцией проверки с примерно равной длительностью, и выигрыша мы не получим. Правильный способ – заранее сгруппировать точки с одинаковым направлением градиента, так чтобы для каждой точки кадра с известным градиентом перебирались точки карты только внутри его группы. Остальные точки будут просто игнорироваться.

Давайте реализуем этот подход.

Для поиска направления градиента можно использовать функцию `imgradient()`:

```
[G,gamma]=imgradient(A,'Sobel');
```

Возвращаемые значения – это модуль градиента `G` и фаза градиента `gamma`.

Вы также можете воспользоваться функциями `imfilter` и `atan2d`, как в л/р№3, для получения аналогичного результата.

Модуль и фаза градиента уже один раз вычислялись внутри функции `edge()`, но мы, к сожалению, не можем их оттуда достать, поэтому вычислим еще раз, это не займет много времени.

Обратите внимание – модуль градиента нам здесь не нужен, и для подавления предупреждений его можно заменить значком «тильда» (находится слева сверху, на клавише «Ё»:

```
[~,gamma]=imgradient(A,'Sobel');
```

Кадр и карта у нас уже представлены списком точек. Добавьте список направлений градиента в точки контура (для кадра и карты).

```
ga=zeros(size(x));  
for i=1:n, ga(i)=gammaA(y(i),x(i));end
```

Отсортируем точки границ кадра по возрастанию направления градиента. Обратите внимание – точек границ кадра очень много, поэтому для сортировки неприемлемы простые подходы типа сортировки пузырьком. Воспользуйтесь алгоритмом быстрой сортировки.

Обратите внимание – в практических задачах желательно сортировать точки карты, а не кадра, потому что это можно сделать заранее. Тогда в цикле мы должны перебирать границы

кадра, а не карты. В данной лабораторной работе мы поступим наоборот просто потому, что точек кадра у нас меньше и отсортируются они гораздо быстрее.

Функцию сортировки удобно реализовать в виде вложенной (nested) функции прямо внутри тела основной функции. В этом случае ей будут доступны общие переменные, в первую очередь – сортируемые массивы, и их не надо будет передавать как аргументы.

Проверьте правильность работы функции сортировки. Поставьте точку останова на команду её вызова, и убедитесь, что после сортировки направления градиента начинаются от -180 и заканчиваются на 179,99.

Теперь разобьём список на части, установив указатель на начало каждой части (каждый указатель содержит номер первого элемента своего подсписка). Массив указателей назовем *закладками*: они похожи на закладки в толстой книге для облегчения поиска. Отсортированные данные с закладками называются *партиципированными данными*.

Теперь, зная направление градиента в точке карты, легко получить подсписок точек с похожим направлением градиента: он будет начинаться со «своей» закладки и заканчиваться на следующей минус один: $ID=Q(n)..Q(n+1)-1$.

Для удобства следует также добавить начальную(нулевую) закладку на начало списка и конечную(терминальную), указывающую на конец массива.

Создайте массив закладок и заполните его. Первая закладка будет равна единице. Для расчета каждой последующей нужно рассмотреть элементы списка, начиная с предыдущей закладки, и увеличивать индекс до тех пор, пока значения угла соответствуют своей закладке.

Убедитесь в правильности заполнения закладок, установив точку останова на момент окончания их расчета.

Теперь осталось модифицировать сам алгоритм преобразования. Теперь для каждой точки границы карты с градиентом γ_A мы будем перебирать только подписки, соответствующие направлениям градиента от $\gamma_A - \epsilon$ до $\gamma_A + \epsilon$. Значение ϵ для фильтра Собела 3×3 примем равным пяти градусам.

Отладьте и запустите полученную функцию из скрипта. При правильно написанной функции величина максимума главного отклика должна составлять несколько тысяч. Выведите на экран результат работы функции. Замерьте время расчета (*tic,toc*). Найдите положение максимального отклика. Отобразите кадр в найденном положении.

Правильно ли найдено положение объекта?

Сколько времени занял поиск положения? Можно ли теперь использовать алгоритм в реальном времени?

Рассмотрите дуальное пространство. Оно может показаться абсолютно чёрным, однако это не так: слева чуть выше середины есть яркий острый максимум.

Часть 3 – Исследование обобщенного Преобразования Хафа

Определите граничные характеристики по углу поворота, масштабу и уровню шума.

Для этого нужно в цикле перебирать исследуемый фактор и рассчитывать смещение найденного положения относительно заведомо верного. Первое большое ошибочное значение принимается в качестве критического.

Постройте графики зависимости ошибки от исследуемых мешающих факторов.

Объясните наблюдаемые результаты.

Сравните характеристики Обобщенного преобразования Хафа с характеристиками спектральных методов. Найдите сильные и слабые стороны данного метода.

Наиболее существенный фактор для ГНТ – это поворот. При повороте кадра относительно карты количество совпадающих точек очень быстро уменьшается, а максимум размывается сложным образом.

Скорректируйте воздействие шума за счет усреднения пространства поиска по окну (функция $R1=imfilter(R,ones(7))$). Как изменится чувствительность к масштабу и повороту?

Приложения

Приложение 1 – преобразование Хафа без использования направления градиента

```
function [ H ] = FindGHT( A,B )
%FindGHT вычисляет GHT кадра и карты
%A-карта
%B-кадр

[h,w]=size(A);
[h1,w1]=size(B);

H=zeros(h,w);

Ca=edge(A, 'Sobel');
Cb=edge(B, 'Sobel');

[y,x]=find(Ca==1);
[y1,x1]=find(Cb==1);
n=length(x);
n1=length(x1);

for i=1:n
    for j=1:n1
        dx=x(i)-x1(j);
        dy=y(i)-y1(j);
        if (dx>=1) && (dy>=1) && (dx<=w) && (dy<=h)
            H(dy,dx)=H(dy,dx)+1;
        end
    end
end

end

end
```

Приложение 2 – преобразование Хафа с использованием направления градиента

```
function [ H ] = FindGHT2( A,B )
%FindGHT вычисляет GHT кадра и карты с использованием направления градиента
%A-карта
%B-кадр
%версия с N=360 градусов и сортировкой по кадру

[h,w]=size(A);
H=zeros(h,w);

%находим границы
Ca=edge(A, 'Sobel');
Cb=edge(B, 'Sobel');

%находим направление градиента
[~,gammaA]=imgradient(A, 'Sobel');
[~,gammaB]=imgradient(B, 'Sobel');

%строим список координат
[y,x]=find(Ca==1);
[y1,x1]=find(Cb==1);
n=length(x);
n1=length(x1);

%строим список направлений градиента
```

```

ga=zeros(size(x));
for i=1:n, ga(i)=gammaA(y(i),x(i));end
gb=zeros(size(x1));
for i=1:n1, gb(i)=gammaB(y1(i),x1(i));end

%сортируем по возрастанию фазы градиента
function QSort(a,b)
    L=a;R=b;
    m=gb(round((a+b)/2));
    while(L<=R)
        while(gb(L)<m), L=L+1;end
        while(gb(R)>m), R=R-1;end
        if(L<=R)
            g=gb(L);gb(L)=gb(R);gb(R)=g;
            t=x1(L);x1(L)=x1(R);x1(R)=t;
            t=y1(L);y1(L)=y1(R);y1(R)=t;
            L=L+1;
            R=R-1;
        end
    end
    if(a<R),QSort(a,R);end
    if(L<b),QSort(L,b);end
end

if(n1>1),QSort(1,n1);end

%формируем закладки через 1 градус
N=360;
Q=zeros(N+1,1);
Q(1)=1;

for i=2:N+1
    Q(i)=Q(i-1);
    q=(i-1)-180;
    for j=Q(i-1)+1:n1
        if gb(j)>q
            Q(i)=j;
            break;
        end
    end
end
Q(N+1)=n1+1;

%Преобр. Хафа
for i=1:n %для каждой точки границ карты
    f=ga(i);
    q=round(f+181);%пересчитаем фазу градиента в № закладки
    q1=q-5; if q1<1, q1=1; end %добавим плюс-минус 5 градусов
    q2=q+5; if q2>360, q2=360; end
    for j=Q(q1):Q(q2)-1 %теперь считаем от одной закладки до другой
        dx=x(i)-x1(j); %строим гипотезу
        dy=y(i)-y1(j);
        if (dx>=1)&&(dy>=1)&&(dx<=w)&&(dy<=h)
            H(dy,dx)=H(dy,dx)+1; %размечаем ее в ПП
        end
    end
end
end

end

```