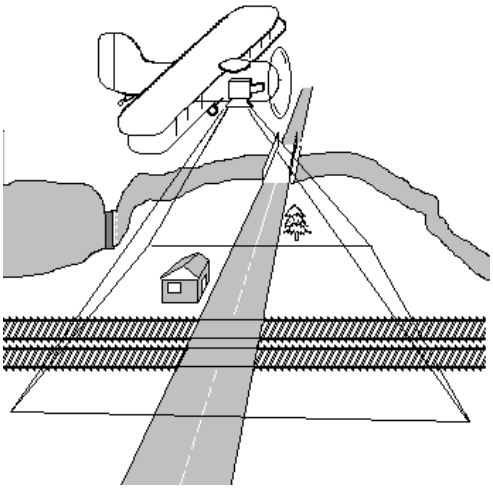
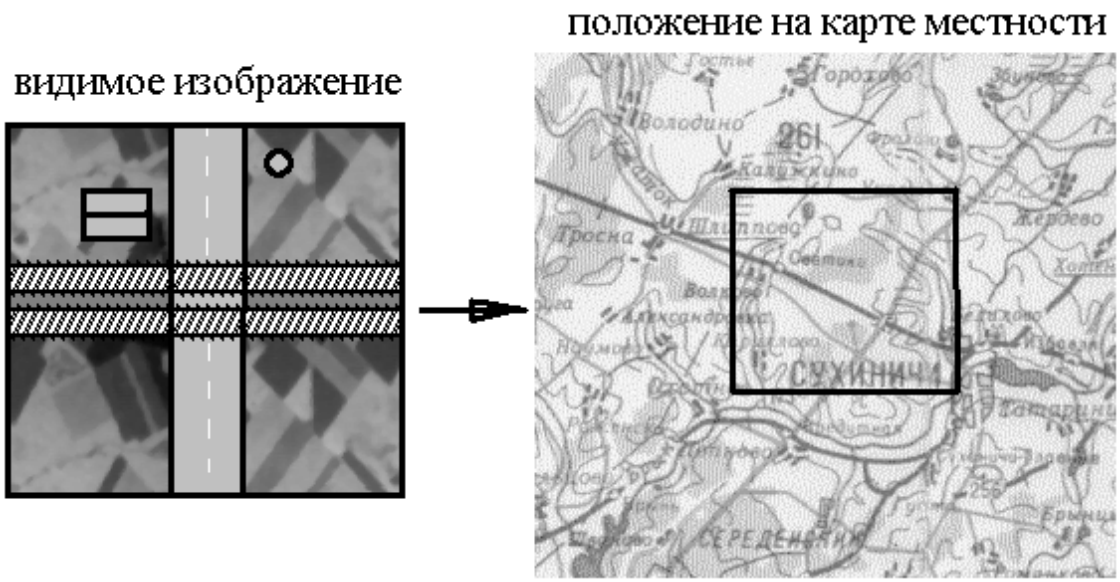
Лабораторная работа № 4

Поиск объекта методами спектральной корреляции.

Теория

Рассмотрим следующую задачу. Пусть имеется БПЛА, оснащенный видеокамерой. Сравнивая получаемое изображение с электронной бортовой картой, нужно определить положение кадра на карте.





Простейший способ решения –прикладывать кадр к карте в каждом из возможных положений и считать количество совпавших точек. Однако здесь мы сразу столкнемся с проблемой: кадр и карта получены в разные моменты времени, и условия освещенности тоже будут разными. Значит, совпадающих точек в значимых количествах у нас не будет.

Альтернативой является использование корреляционной функции:

$$R(x,y)=\Sigma\Sigma\ A(x+dx,y+dy)*B(dx,dy)$$

Положению объекта на изображении соответствует максимум двухмерной взаимно-корреляционной функции изображения и объекта. В этом случае точного совпадения яркостей точек не требуется, и данный подход можно использовать и при изменяющихся условиях освещённостей.

Смысл корреляционной функции можно понять на следующем примере. Пусть и кадр, и карта будут монохромными, то есть будут содержать только черные (нули) и белые(единицы) точки. Тогда попарные произведения яркостей точек кадра и карты будут давать единицу только тогда, когда обе точки – белые. То есть смысл корреляционной функции для монохромного

изображения – количество совпадающих белых точек в данном положении. Разумеется, корреляционная функция может вычисляться и для полутоновых изображений.

Расчет корреляции по приведенной формуле имеет существенный недостаток: он требует просто огромного количества времени. Для карты в несколько десятков мегапикселей время поиска может составлять порядка суток. Понятно, что с такой скоростью расчета о работе в реальном времени придется забыть.

Как же ускорить расчет корреляции? Простые и очевидные способы, такие как уменьшение размера кадра и карты, не позволяют достичь заметного ускорения без катастрофического снижения надежности поиска. Мы пойдем другим путем.

Решение лежит в использовании *теоремы о спектре свертки*:

$$F(A \otimes B)=F(A)xF^*(B),$$

где F() –дискретное преобразование Фурье,

⊗ – операция кольцевой свёртки,

* – операция комплексного сопряжения.

Теорема о спектре свертки говорит нам очень важную вещь: вместо длинного расчета кольцевой свертки, в спектральной области можно использовать другую, более короткую и быструю операцию – почленное перемножение элементов спектра.

Корреляционную функцию R(A,B) можно представить в виде кольцевой свертки, дополнив меньшее изображение (кадр) до размера большего нулями. Очевидно, что значение корреляционной функции от этого не изменится: при умножении на добавленные нули мы будем получать и суммировать также нули. Тогда расчетная формула примет окончательный вид:

$$R(A,B)=F^{-1}[F(A)xF^*(B)],$$

где $F^{-1}[\ ]$– обратное преобразование Фурье.

Использование этой формулы дает феноменальный эффект: объем вычислений падает на четыре-пять порядков, и теперь для расчета корреляционной функции нам потребуются доли секунды.

Основной проблемой теперь будет чувствительность к условиям освещенности. При изменении освещенности даже на небольшую величину, эта величина добавляется к яркости всех точек, и значение двойной суммы изменится достаточно значительно.

Для снижения чувствительности к условиям освещенности можно использовать следующие подходы:

- Нормирование корреляционной функции
- Фазовая корреляция
- Градиентная корреляция
- Контурная корреляция.

Нормирование

Нормирование в простейшем случае предусматривает центрирование кадра и карты, то есть вычитание из яркости их точек средней яркости.

И тут нас ожидает пренеприятный сюрприз: распределение яркостей на изображении земли вовсе не будет гауссовым: оно будет мультимодальным. Это значит, что разные области изображения – леса, поля, застройка – будут разного цвета, и будут иметь разные средние яркости. Поэтому для нормирования каждой точки надо вычислять среднюю яркость по ее окрестности. Размер окрестности очевидным образом равен размеру искомого кадра. И снова мы упираемся в проблему чрезмерного объема вычислений: расчет средних яркостей окрестности каждой точки займет много часов.

К счастью, есть простое решение данной проблемы. Оно связано с тем, что расчет среднего по прямоугольной области можно разбить на два одномерных расчета среднего: сначала вычислим среднее по строчкам прямоугольной области, потом – по столбцам от предыдущего результата. Более того, среднее по строкам и по столбцам для двух соседних точек будет отличаться всего на

две величины – начала и конца строчки. Это позволяет построить быстрый алгоритм, требующий всего четырех проходов по изображению.

Нормирование по среднему позволяет повысить устойчивость корреляционных методов к изменению условий освещенности, но, к сожалению, его эффективность невелика. Рассмотрим другие методы.

Фазовая корреляция

В основе фазовой корреляции лежит еще одна теорема –*теорема о спектре сдвига (теорема о запаздывании):*

$$F[X(t+dt)] = F[X(t)] \exp(*-2\pi j\, dt)$$

О чем нам говорит эта теорема? Она говорит важную вещь: при чистом сдвиге на dt амплитуда спектра не меняется. То есть если кадр просто вырезан из какого-то места карты, без поворотов и изменения размера, то вся информация о его положении будет находиться исключительно в фазе спектра. При этом амплитуда спектра несет в себе информацию о других воздействиях – шуме, условиях освещенности и т.д. Поэтому амплитуду спектра можно смело убрать из рассмотрения. Проще всего это сделать, разделив каждое значение кросс-спектра на его модуль (не забываем, что спектр – комплексный, у каждой гармоники есть амплитуда и фаза):

$$R(A,B)=F^{-1}[F(A)xF^*(B) / |F(A)xF^*(B)|]$$

Что даст такой подход? Во-первых, если кадр полностью совпадает с картой, то корреляционная функция станет дельта-функцией: единственный острый пик на нулевом фоне. И точность, и надежность поиска радикально возрастают.

Во-вторых, корреляционная функция стала полностью нечувствительной к условиям освещенности: они не влияют на фазу спектра, а амплитуду спектра мы предусмотрительно удалили.

В-третьих – шум также удалился вместе с амплитудой спектра. Метод стал очень устойчивым к шуму.

Слишком хорошо, чтобы быть правдой. Разумеется, у метода есть серьезный недостаток: теорема о сдвиге во времени справедлива только для чистого сдвига. Если же у нас присутствует также повороты, масштабирования, проективные преобразования, то условия теоремы уже не будут выполняться. Корреляционная функция уже не будет иметь острый максимум, он будет размываться, дробиться и тонуть в шуме.

Тем не менее, и здесь не все так страшно. Современные методы на основе фазовой корреляции могут использовать отдельно амплитуду спектра для определения масштаба и поворота (сдвиг не действует на амплитуду, не забыли?). А затем, учтя найденный масштаб и поворот, можно повторно использовать фазовую корреляцию уже для определения сдвига.

Градиентные методы

Градиентные методы используют корреляцию модулей градиента яркости вместо корреляции самих яркостей. Это позволяет благодаря дифференцированию удалить из рассмотрения изменения освещенности: градиент практически не изменяется при изменении освещенности. Это позволяет избавиться от условий освещенности, не прибегая к фазовой корреляции.

Градиентная корреляция может вычисляться как для модуля градиента яркости, так и для исходных компонент градиента dI/dx и dI/dy.:

$$R1(A,B)=F^{-1}[F(G_A)xF^*(G_B)],$$

$$R2(A,B)=F^{-1}[F(G_{Ax})xF^*(G_{Bx}) + (G_{Ay})xF^*(G_{By})],$$

где $G_{Ax}=dA/dx$, $G_{Ay}=dA/dy$, $G_{Bx}=dB/dy$ $G_{By}=dB/dy$, – производные карты A и кадра B по x и по y соответственно,

$G_A= \sqrt{(G_{Ax}^2+ G_{Ay}^2)}$ – модуль градиента карты A,

$G_B= \sqrt{(G_{Bx}^2+ G_{By}^2)}$ – модуль градиента кадра B.

Направление градиента можно использовать, чтобы дополнительно уточнить поворот кадра относительно карты.

Градиентная корреляция, подобно фазовой корреляции, устойчива к шуму и изменениям освещенности, но вместе с тем не слишком чувствительна к геометрическим искажениям. Максимум градиентной корреляционной функции отчетливый, но более размытый, что может снижать точность обнаружения объекта. Чувствительность к шуму у градиентного метода хуже: дифференцирование подчеркивает высокочастотные компоненты, и каждая шумовая точка получает в довесок восемь соседей с ненулевым градиентом.

В целом, градиентный метод не имеет критических слабых мест, и может использоваться в широком спектре приложений.

Контурная корреляция

Контурная корреляция решает проблему изменения освещенности радикальным образом: отказом от всех яркостных признаков в пользу признаков геометрических. В корреляции участвуют только точки контура, положение которых есть геометрический признак, никак не зависящий от освещенности.

Контура для корреляции могут быть найдены при помощи фильтра Кэнни, который мы рассматривали в предыдущей лабораторной работе.

Контурная корреляция дает начало целой группе методов, которые используют корреляцию не только для отдельных точек, но и для более крупных фигур – линий, отрезков, дуг, многоугольников. Это методы векторной корреляции.

Контурная корреляционная функция очень похожа на фазовую: это один яркий максимум на равнине из нулей. При этом контурная корреляция гораздо в меньшей степени чувствительна к геометрическим искажениям, нежели фазовая, и может представлять неплохую альтернативу.

К сожалению, у контурной корреляции есть характерный недостаток – чувствительность к шуму. При высоком уровне шума корректное детектирование границ и их дальнейшее использование становится невозможным.

Ход работы.

Часть 1. Прямой метод расчета корреляционной функции.

В качестве исходного изображения для карты возьмем аэрофотографию ConcordOrthoPhoto.png, а в качестве кадра – ее западную часть WestConcordOrthoPhoto.png

Загрузите эти изображения, отобразите их на экране.

Попытайтесь определить положение кадра на карте.

Обратите внимание на размеры изображений.

Рассчитайте корреляционную функцию с использованием прямой формулы:

$$R(x,y)=\Sigma\Sigma\ A(x+dx,y+dy)*B(dx,dy)$$

Удалось ли это сделать? Почему?

Внутри цикла по x желательно сделать диагностический вывод номера текущей строки.

Сколько времени займет полный расчет корреляционной функции?

Часть 2. Спектральный расчет корреляции.

Создайте новый скрипт, скопируйте в него загрузку кадра и карты.

Найдем корреляционную функцию с использованием теоремы о свёртке.

$$R(A,B)=F^{-1}[F(A)xF^*(B)],$$

Для этого нам потребуется:

1.Найти двухмерный дискретный спектр изображения Sa=F[A]. Для расчета двухмерного спектра имеется удобная функция fft2, которая его вычисляет:

```
Sa=fft2(A);
```

Функция fft2 в Матлабе взята из библиотеки fftw. Это бесплатная и хорошо оптимизированная библиотека с открытым кодом, вы можете использовать ее в своих приложениях.

2.Найти спектр кадра. Здесь сложность заключается в том, что спектр кадра должен иметь тот же размер, что и спектр карты, иначе мы не сможем их перемножить. Поэтому необходимо дополнить кадр нулями до размера карты. К счастью, функция fft2 позволяет легко это сделать: достаточно указать желаемые размеры вторым и третьим параметром:

```
[h,w]=size(A);
Sb=fft2(B,h,w);
```

3.Найти комплексно-сопряженный спектр Sb*, поменяв знак у комплексной части каждого элемента спектра. Здесь нам поможет функция conj (*conjunction* – сопряжение):

```
Sb1=conj(Sb);
```

4.Почленно перемножить элементы спектра, чтобы получить кросс-спектр. Операция почленного перемножения –«.*»:

```
Sr=Sa.*Sb1;
```

5. Выполнить обратное преобразование фурье, чтобы получить корреляционную функцию. Здесь нам поможет функция ifft2 (она тоже из библиотеки fftw):

```
R=ifft2(Sr);
```

Создайте функцию для расчета корреляции спектральным методом. Функция должна принимать на вход изображения кадра и карты, а на выход возвращать корреляционную функцию. Эта функция нам потребуется в дальнейшем. Вызовите функцию из главного скрипта, отобразите результат на экране.

С помощью функций tic и toc замерьте время расчета корреляционной функции.

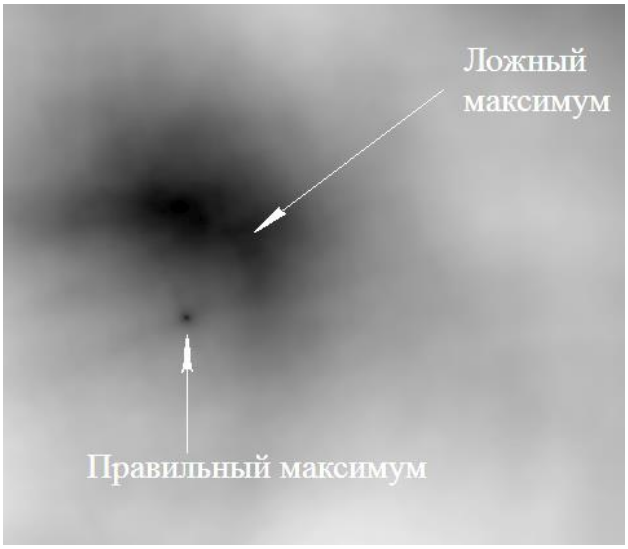
Во сколько раз ускорился алгоритм?

Можно ли теперь его использовать для приложений реального времени?

Давайте определим, совпадает ли максимум корреляционной функции с положением кадра на карте. Найдите положение максимума R и отрисуйте в этом положении кадр поверх карты:

```
Rmax=max(R(:));
[y,x]=find(R==Rmax);
A(y:y+h1-1,x:x+w1-1)=B+50;
figure('Name','Найденное положение'),imshow(A);
```

Удалось ли найти правильное положение? Почему?



Потому что помимо правильного острого максимума у нас откуда-то появился обширный лажный максимум правее и выше. Он появился из-за того, что условия видимости карты и кадра отличаются: кадр светлее.

Давайте скорректируем яркость кадра: вычтем из яркостей всех точек 20 единиц:

B=B-20;

Теперь кадр находится правильно.

Часть 3. Фазовая корреляция

Мы справились с проблемой, заранее зная, что кадр на 20 единиц светлее. Что же делать, если изменение яркости заранее невозможно? Если вы читали теоретическую часть, то должны знать: один из возможных вариантов – использование фазовой корреляции.

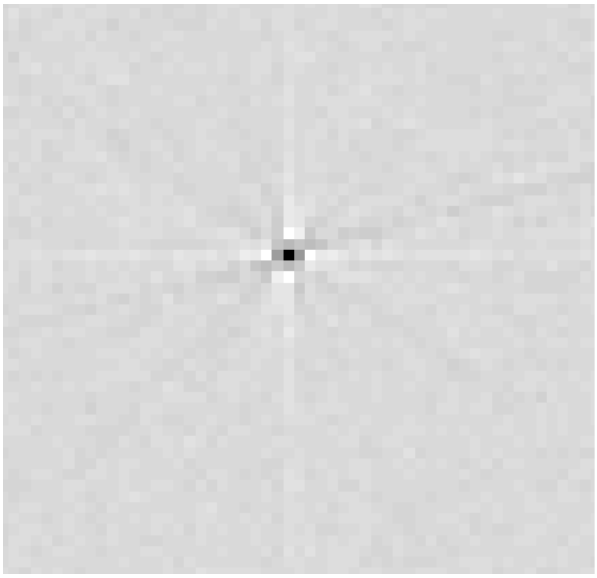
Создайте новую функцию и перенесите в нее код из предыдущей.

В функции удалите амплитудную часть спектра, разделив каждый элемент спектра на его модуль (abs).

Вызовите полученную функцию из главного скрипта вместо предыдущей.

Удалось ли найти точное положение кадра?

Внимательно рассмотрите вид корреляционной функции. Она состоит из одной яркой точки на чистом нулевом фоне (точку следует искать слева чуть выше середины).



Обратите внимание, что корреляционная функция сохранит свою форму практически при любой освещенности. Измените яркость кадра B на ±20, ±50, ±100 единиц и убедитесь в этом.

Добавьте на изображение шум (imnoise).

Удается ли найти положение кадра в этом случае? Почему?

Как ведет себя корреляционная функция при воздействии шума?

Закомментируйте удаление шума и добавьте поворот кадра (imrotate) на угол от 1 до 5 градусов. Удается ли найти положение кадра в этом случае? Почему? Как ведет себя корреляционная функция при воздействии поворота?

Закомментируйте поворот и добавьте масштабирование (imresize) на величину от 0,95 до 0,9. Удается ли найти положение кадра в этом случае? Почему? Как ведет себя корреляционная функция при изменении масштаба?

Часть 4. Градиентная корреляция

Создайте новую функцию и перенесите в нее код из нашей первой функции с расчетом корреляции. В начале функции добавьте расчет модуля градиента (imgradient):

Ga=imgradient(A,’Sobel’);

Gb=imgradient(B,’Sobel’);

Замените расчет спектра яркости на расчет спектра градиента.

Удаление амплитуды кросс-спектра в данном методе не требуется.

Вызовите функцию из главного скрипта.

Удалось ли найти положение объекта?

Внимательно рассмотрите полученную корреляционную функцию. В чем ее отличие от фазовой корреляции?

Исследуйте устойчивость градиентной корреляции к изменениям яркости, шуму, повороту и масштабированию, как в предыдущем случае. Проследите за характером изменения максимума

корреляционной функции. Определите, к каким мешающим факторам градиентная корреляция более устойчива, а к каким – менее, по сравнению с фазовой корреляцией.

Создайте еще одну функцию для покомпонентной градиентной корреляции.

Компоненты вектора градиента dI/dx и dI/dy вычислите с помощью функции imgradientxy:

[Gya,Gxa]=imgradientxy(A,'Sobel');

[Gyb,Gxb]=imgradientxy(B,'Sobel');

Кросс-спектр рассчитайте по формуле:

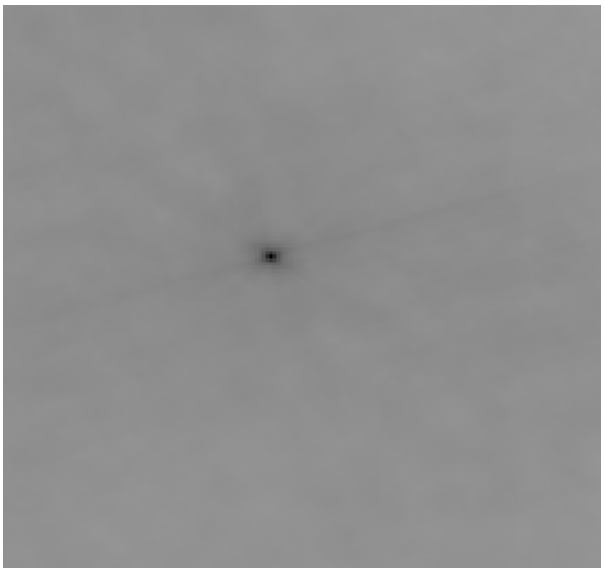
$$Sr= F(G_{Ax})\ F^*(G_{Bx}) + (G_{Ay})\ F^*(G_{By})$$

Сравните результат с исходной градиентной корреляцией.

Какой метод будет работать надежнее? Почему?

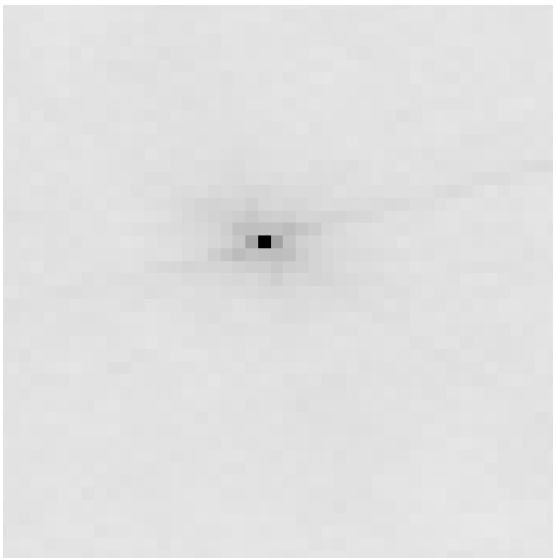
Исследуйте устойчивость градиентной корреляции к изменениям яркости, шуму, повороту и масштабированию, как в предыдущем случае.

Какая из функций будет работать быстрее? Насколько? Почему?



Часть 5. Контурная корреляция

В контурной корреляции участвуют только точки контура. Положение точек контура не зависит от условий видимости, поэтому следует ожидать, что контурная корреляция будет нечувствительна к условиям освещенности.



Создайте новую функцию, скопируйте туда код для градиентной корреляции.

Замените расчет спектра модуля градиента на расчет спектра контура. Для расчета точек контура используйте функцию edge.

Удалось ли найти точное положение кадра?

Внимательно рассмотрите вид корреляционной функции. Она снова состоит из одной яркой точки на чистом нулевом фоне, как и в случае фазовой корреляции. Таким образом, этот метод будет давать очень высокую точность поиска: при смещении даже на один пиксель в сторону количество совпадающих точек контуров резко падает, и спутать правильное положение с чем-либо не удасться.

Исследуйте устойчивость контурной корреляции к изменениям яркости, шуму, повороту и масштабированию, как мы это делали ранее.

Часть 6. Сравнение методов.

Мы рассмотрели несколько методов поиска объектов. Какой же из них следует выбрать на практике? Для этого нужно уметь сравнивать характеристики методов. В качестве характеристик выступает скорость работы, точность определения положения и устойчивость к воздействию мешающих факторов (шум, изменение освещенности и ракурс).

Исследуем воздействие шума на алгоритмы. Для этого будем изменять уровень шума в кадре от 0 до 100%, и для каждого зашумленного кадра будем определять его положение. Для каждого найденного положения будем вычислять ошибку – отклонение найденных координат от правильных. За правильные координаты можно принять положение кадра без шума, оно находится точно. Введем допустимый порог ошибки – к примеру, 20..25 точек. Уровень шума, при котором ошибка превысит порог, будет являться критическим для данного алгоритма.

Постройте графики ошибки для каждого из алгоритмов, определите критический уровень шума, занесите его в таблицу. Шаг по уровню шума можно взять 5..10%.

Аналогично исследуйте устойчивость алгоритмов к повороту и к масштабированию. Шаг по углу возьмите равным одному градусу, а по масштабу – одному проценту.

Поскольку эксперименты могут занять существенное время, то целесообразно будет сначала сделать замеры каждого параметра для каждого метода и сохранить их, а потом – загрузить и отобразить на общем графике.

Вопросы и задания.

Выполните все эксперименты, описанные в части 6. Результаты отобразите в виде таблиц (критические значения) и графиков.

Определите, какой метод наиболее устойчив к шуму, а какой –наименее.

Определите, какой метод наиболее устойчив к повороту, а какой –наименее.

Определите, какой метод наиболее устойчив к масштабированию, а какой –наименее.

Объясните полученные результаты.

Определите, какой метод наиболее точен, а какой – наименее.

Для каждого метода определите время его работы.

Приложения

Приложение 1. Прямой расчет корреляции

```
%-----------------------------------
% Л/р №4
% Поиск объекта
% Часть 1 - прямой расчет корреляции
%-----------------------------------

clear;
close all;
A=imread('ConcordOrthoPhoto.png');
figure,imshow(A),title('Карта');
B=imread('WestConcordOrthoPhoto.png');
figure,imshow(B),title('Кадр');

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

for x=1:w-w1-1
    fprintf('%d\n',x);
    for y=1:h-h1-1
        s=0;
        for dx=1:w1
            for dy=1:h1
                s=s+A(y+dy,x+dx)*B(dy,dx);
            end
        end
        R(y,x)=s;
    end
end
```

Приложение 2. Скрипт для запуска методов

```
%-----------------------------------
% Л/р №4
% Поиск объекта
% Часть 2 - скрипт запуска алгоритмов корреляции
%-----------------------------------

clear;
close all;
A=imread('ConcordOrthoPhoto.png');
% figure,imshow(A),title('Карта');
B=imread('WestConcordOrthoPhoto.png');
% figure,imshow(B),title('Кадр');
% B=B-100;
% B=imnoise(B,'salt & pepper',0.25);

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

tic
R=FindCorr(A,B);
% R=PhaseCorr(A,B);
% R=GradCorr(A,B);
% R=GradCorrXY(A,B);
% R=ContCorr(A,B);
toc
```

```
F=figure('Name','корреляционная функция');imshow(-R,[]);
Rmax=max(R(:));
[y,x]=find(R==Rmax);
x=x(1);y=y(1);

A(y:y+h1-1,x:x+w1-1)=B*1.2+50;
figure('Name','Найденное положение');imshow(A);
```

Приложение 3. Спектральная корреляция по яркости

```
function [ R ] = FindCorr( A,B )
%FindCorr Считает корреляционную функцию двух изображений
%спектральным методом

[h,w]=size(A);
Sa=fft2(A);
Sb=fft2(B,h,w);
Sr=Sa.*conj(Sb);
R=ifft2(Sr);

end
```

Приложение 4. Фазовая корреляция

```
function [ R ] = PhaseCorr( A,B )
%PhaseCorr Считает фазовую корреляционную функцию двух изображений
%спектральным методом

[h,w]=size(A);
Sa=fft2(A);
Sb=fft2(B,h,w);
Sr=Sa.*conj(Sb);
Sr=Sr./abs(Sr); %Оставляем только фазу спектра
R=ifft2(Sr);

end
```

Приложение 5. Градиентная корреляция

```
function [ R ] = GradCorr( A,B )
%GradCorr Считает градиентную корреляционную функцию двух изображений
%спектральным методом

[h,w]=size(A);
Ga=imgradient(A,'Sobel');
Gb=imgradient(B,'Sobel');
Sa=fft2(Ga);
Sb=fft2(Gb,h,w);
Sr=Sa.*conj(Sb);
R=ifft2(Sr);

end
```

Приложение 6. Корреляция по компонентам вектора градиента

```
function [ R ] = GradCorrXY( A,B )
%GradCorrXY Считает корреляционную функцию двух изображений
%спектральным методом по компонентам вектора градиента
```

```
[h,w]=size(A);
[Gya,Gxa]=imgradientxy(A,'Sobel');
[Gyb,Gxb]=imgradientxy(B,'Sobel');
Sa=fft2(Gxa);
Sb=fft2(Gxb,h,w);
Sr=Sa.*conj(Sb);

Sa=fft2(Gya);
Sb=fft2(Gyb,h,w);
Sr=Sr+Sa.*conj(Sb);

R=ifft2(Sr);

end
```

Приложение 7. Контурная корреляция

```
function [ R ] = ContCorr( A,B )
%ContCorr Считает контурную корреляционную функцию двух изображений

[h,w]=size(A);
Ea=edge(A,'Sobel');
Eb=edge(B,'Sobel');
Sa=fft2(Ea);
Sb=fft2(Eb,h,w);
Sr=Sa.*conj(Sb);
R=ifft2(Sr);

end
```

Приложение 8. Сравнение методов

```
%-----------------------------------
% Л/р №4
% Поиск объекта
% Часть 6 - сравнение алгоритмов корреляции
%-----------------------------------

clear;
close all;
A=imread('ConcordOrthoPhoto.png');
% figure,imshow(A),title('Карта');
B=imread('WestConcordOrthoPhoto.png');
% figure,imshow(B),title('Кадр');

N=0:5:100;  %Уровень шума
Ang=0:1:20;    %Угол поворота
M=100:-1:80; %Масштаб

Exp=2;   %* 1-шум 2-поворот 3-масштаб *

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

%Ищем правильное положение
R=PhaseCorr(A,B);
Rmax=max(R(:));
[y,x]=find(R==Rmax);
xtrue=x(1);ytrue=y(1);
```

```
E=zeros(20,1);  %график ошибки
T=zeros(20,1);  %подпись для оси x

for pass=1:20

    if     Exp==1, B1=imnoise(B,'salt & pepper',N(pass)/100);
           T(pass)=N(pass); tt='Чувствительность к шуму';
    elseif Exp==2, B1=imrotate(B,Ang(pass));
           T(pass)=Ang(pass); tt='Чувствительность к повороту';
    elseif Exp==3, B1=imresize(B,M(pass)/100);
           T(pass)=M(pass); tt='Чувствительность к масштабу';
    else B1=B; T(pass)=pass; tt='Не задано.';
    end

% *Выбираем один из методов:*
R=PhaseCorr(A,B1);
% R=GradCorr(A,B1);
% R=GradCorrXY(A,B1);
% R=ContCorr(A,B1);

Rmax=max(R(:));
[y,x]=find(R==Rmax);
x=x(1);y=y(1);

err=round(sqrt((x-xtrue)^2+(y-ytrue)^2));
E(pass)=err;

fprintf('Pass=%d Err=%d\n ',pass,err);
end;

Ecrit=20*ones(size(E));
figure('Name',tt);plot(T,E,T,Ecrit);
```