

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

А. С. Потапов, Р. О. Малашин

СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ
Учебно-методическое пособие
по лабораторному практикуму



Санкт-Петербург

2012

Потапов А.С., Малашин Р.О. Системы компьютерного зрения: Учебно-методическое пособие по лабораторному практикуму. – СПб: НИУ ИТМО, 2012. – 41 с.

Пособие содержит краткое описание лабораторного практикума по методам обработки оптической информации, используемым в системах компьютерного зрения. Формулируются требования к проведению работ, даются рекомендации и примеры выполнения отдельных их этапов. Предназначено для студентов, обучающихся по направлению подготовки 200700 – «Фотоника и оптоинформатика».

Рекомендовано советом факультета Фотоники и оптоинформатики НИУ ИТМО для использования в качестве учебно-методического пособия для студентов высших учебных заведений, обучающихся по направлению подготовки 200700 – «Фотоника и оптоинформатика».

В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена Программа развития государственного образовательного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики» на 2009–2018 годы.



© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2012

© А. С. Потапов, 2012
© Р. О. Малашин, 2012

1. Моделирование биологических систем зрительного восприятия

Цель работы – исследовать возможность компьютерного моделирования поведения ганглиозных клеток сетчатки и нейронов стриарной коры либо в форме искусственных нейронных сетей с повторением структуры рецептивных полей, либо в форме последовательности фильтров, осуществляющих свертку изображения с различными ядрами.

Задание по работе:

1. Изучить теоретическую часть работы.
2. Реализовать компьютерную модель, воспроизводящую основные особенности работы ганглиозных клеток сетчатки, простых и сложных клеток зрительной коры.
3. Для различных видов клеток исследовать отклики на стимулы различных типов; экспериментально определить их рецептивные поля и установить степень соответствия с известными характеристиками их биологических аналогов.

Теоретическая часть

Ганглиозные клетки сетчатки

Сетчатка преобразует электромагнитное излучение в нервные сигналы благодаря слою светочувствительных рецепторов (палочек и колбочек). Этот слой, как ни странно, расположен дальше всего от поверхности глаза, но, с точки зрения обработки зрительной информации, является первым. Палочки обеспечивают способность видеть при слабом освещении, в то время как колбочки обеспечивают цветное зрение в условиях хорошего освещения. В каждом глазу имеется в сумме порядка 125 млн. палочек и колбочек, неравномерно распределённых по сетчатке (в центральной ямке, концентрация колбочек максимальна, что обеспечивает высокую остроту зрения).

Вместе со слоем рецепторов сетчатка содержит три крупных слоя нервных клеток, разделенных двумя слоями, образованными дендритами и аксонами этих клеток (толщина сетчатки составляет около четверти миллиметра).

Передний слой сетчатки состоит из *ганглиозных* клеток. Аксоны этих клеток проходят по поверхности сетчатки к слепому пятну, в котором они собираются в зрительный нерв, соединяющий глаз и мозг. Интересно отметить, что ганглиозных клеток, передающих зрительную информацию в мозг из сетчатки, всего около 1 миллиона, то есть примерно в 125 раз меньше, чем число рецепторов. Структура сетчатки приведена на рис. 1.

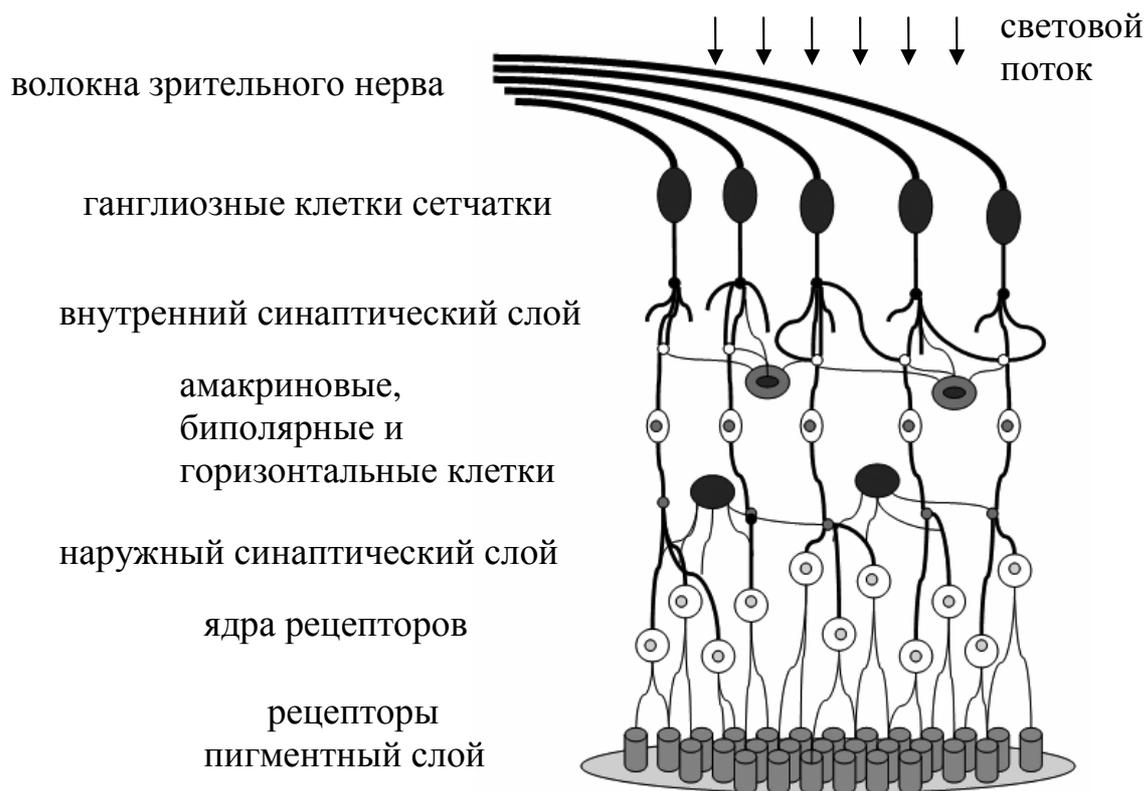


Рис. 1. Структура сетчатки

В среднем слое (между ганглиозными клетками и рецепторами) имеются три типа нейронов: биполярные, горизонтальные и амакриновые. Биполярные клетки получают входные сигналы от сравнительно небольшого числа рецепторов и в большинстве своем направляют выходные сигналы непосредственно ганглиозным клеткам. Биполярные клетки могут иметь входы не только от рецепторов, но и от горизонтальных клеток, которые соединены с большим числом рецепторов длинными связями, идущими вдоль слоя сетчатки. Как горизонтальные клетки связывают рецепторы и биполярные клетки, так и амакриновые клетки связывают биполярные клетки с ганглиозными (в дополнение к прямым связям между ними).

При проекции на сетчатку маленького пятнышка света можно найти на сетчатке области, освещение которых увеличивает или подавляет активность конкретной ганглиозной клетки. Такая область (то есть область рецепторов, от которых доходят сигналы до некоторого нейрона) называется *рецептивным полем* соответствующего нейрона.

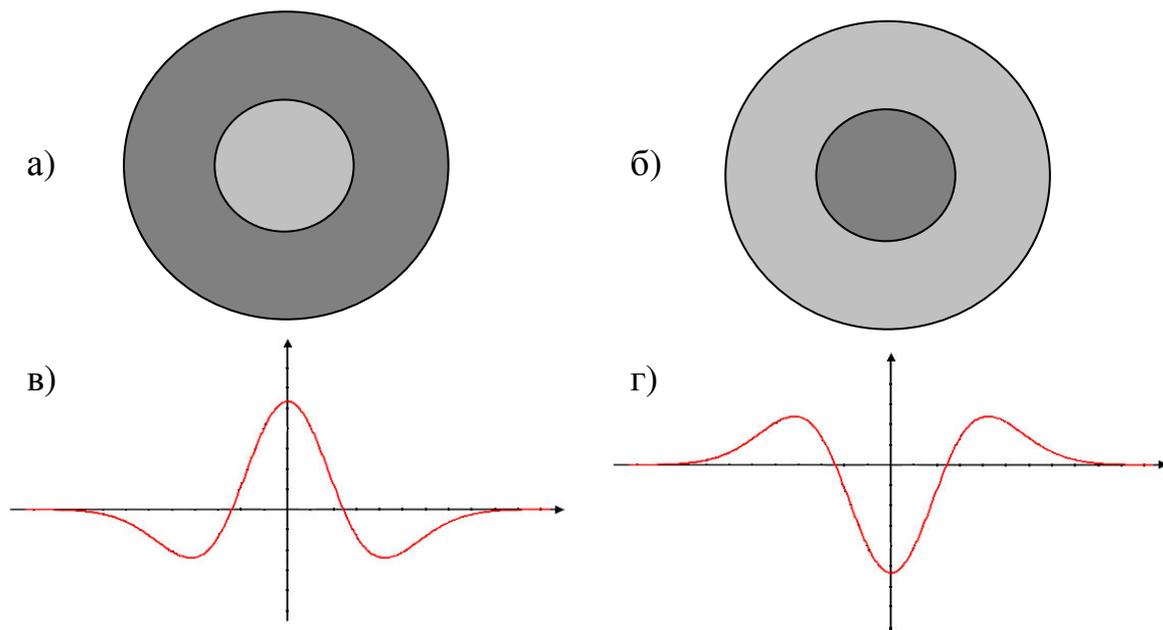


Рис. 2. Структура рецептивных полей ганглиозных клеток (тёмным показаны области торможения, светлым – возбуждения): (а) случай on-реакции; (б) случай off-реакции; (в) и (г) – сечения карт рецептивных полей в случае on- и off-реакций, аппроксимированных как разности гауссиан

Рецептивные поля ганглиозных клеток круглые и состоят из центра и периферии (см. рис. 2), причём эти клетки бывают двух типов: у клеток с так называемой on-реакцией засветка центра рецептивного поля приводит к повышению активности клетки, а засветка периферии – к подавлению её активности, в то время как у клеток с off-реакцией всё в точности наоборот.

Таким образом, если в центр рецептивного поля клетки с on-реакцией проецировать пятнышко света, то при увеличении размеров этого пятнышка активность клетки будет возрастать, пока пятно не заполнит весь центр; когда же пятно начнет заполнять периферию, то активность клетки будет уменьшаться (при равномерной засветке всего рецептивного поля ганглиозная клетка практически не будет проявлять активности).

Вполне возможно, что центр рецептивного поля ганглиозной клетки реализуется с помощью прямого пути распространения сигнала от рецепторов через биполярные клетки к ганглиозным клеткам, ведь биполяры получают входы от небольшого числа рецепторов, а ганглиозные клетки – от небольшого числа биполяров. Тогда периферия рецептивного поля ганглиозной клетки реализуется через не прямой путь, включающий горизонтальные и амакриновые клетки, имеющие более обширные связи.

Стоит отметить, что рецептивные поля "соседних" ганглиозных клеток сильно перекрываются, кроме того, размеры рецептивных полей сильно варьируются. Интерпретация действия ганглиозных клеток (совместно с

клетками среднего слоя сетчатки) может быть различной. С одной стороны, можно полагать, что происходит локальное вычитание яркости фона для получения относительной яркости объектов в целях достижения константности восприятия по отношению к условиям освещения. С другой стороны, считается, что форма рецептивных полей ганглиозных клеток неплохо описывается разностью двух гауссиан с разной дисперсией (рис. 2в и 2г), что является аппроксимацией оператора Лапласа. В таком случае, действие ганглиозных клеток можно трактовать как пространственное дифференцирование изображения с выделением на нём контуров.

Клетки зрительной коры

Аксоны ганглиозных клеток сетчатки (зрительный нерв) входят в наружное коленчатое тело (НКТ), а аксоны клеток НКТ идут далее к первичной зрительной коре, также называемой стриарной корой или полем 17.

При этом связи между сетчаткой, НКТ и стриарной корой имеют топографическую организацию, то есть сохраняют ту же упорядоченность в пространстве, что и у точек изображения на сетчатке. В частности, при разрушении какой-то области стриарной коры наступает такая же «локальная слепота», как и при разрушении некоторой области на сетчатке. Однако это не означает простого взаимнооднозначного соответствия, поскольку при этом производится сложная обработка информации.

Каждое из НКТ содержит примерно 1.5 миллиона нейронов, рецептивные поля которых весьма сходны с рецептивными полями ганглиозных клеток сетчатки. Однако, помимо входов от ганглиозных клеток, нейроны НКТ могут иметь входы от соседних клеток НКТ, обратные входы от тех корковых зон, в которые передаётся информация от НКТ, а также входы от ретикулярной формации. Такая схема связей может иметь отношение к тонким механизмам разрешения неоднозначности, возникающей в процессе зрительного восприятия. Тем не менее, в НКТ зрительная информация преобразуется не существенно. Даже входы от двух глаз разделены по слоям, и возможности стереозрения еще не реализуются. В связи с этим здесь более подробно функции НКТ не рассматриваются.

Напротив, нейроны стриарной коры обладают принципиально новыми свойствами. В то время как ганглиозные клетки и клетки НКТ обладают круговой симметрией, большая (более 70%) часть нейронов зрительной коры обладает свойством дирекционной избирательности (чувствительности к ориентации стимула).

Стриарная кора имеет несколько слоёв, и, естественно, с каждым следующим слоем реакции нейронов усложняются. Первые (в порядке обработки информации в зрительной системе) нейроны с дирекционной

избирательностью были названы *простыми клетками*. Эти нейроны бывают разных типов, чувствительных к трём видам стимула (рис. 3).



Рис. 3. Три типа стимулов (светлая полоса на темном фоне, тёмная полоса на светлом фоне и граница), к которым могут быть чувствительны клетки стриарной коры

Ключевой характеристикой является ориентация стимула. Как правило, простые клетки наиболее сильно реагируют на линию или край одной ориентации, а при отклонении угла ориентации на 10-20 градусов реакция клетки существенно уменьшается. Кроме того, наиболее сильную реакцию простой клетки вызывает стимул при определённом положении, при смещении от которого реакция сильно снижается.

Такая реакция простых клеток может быть реализована путем получения входов от нескольких ганглиозных клеток (или, точнее, клеток НКТ), причём от клеток, соответствующих белым областям полей, представленных на рис. 3, идут тормозящие связи, а от клеток, соответствующих черным областям – активирующие.

Следующему уровню анализа соответствуют *сложные клетки*, получающие входы от простых клеток. Эти клетки также чувствительны к стимулам трёх указанных видов и обладают дирекционной избирательностью. Однако в отличие от простых клеток они реагируют на соответствующий стимул независимо от того, в каком месте рецептивного поля он появляется. Это может быть реализовано путем интеграции откликов от двух или более простых клеток, реагирующих на полосы одинаковой ориентации в соседних точках изображения.

Экспериментальная часть

В данной работе осуществляют построение рецептивных полей клеток искусственной нейронной сети, моделирующей работу клеток зрительного тракта. ИНС может также быть заменена на последовательность фильтров, применяемых к изображению. Работа состоит из следующих шагов.

1. Реализовать программу, моделирующую работу ганглиозных клеток сетчатки и простых и сложных клеток зрительной коры.

2. Для модели ганглиозных клеток определить структуру рецептивного поля, используя стимулы разных размеров и формы. Установить отклик клетки на небольшое световое пятно, помещённое в различные области поля. Определить отклик клетки на светлую полосу на тёмном фоне в зависимости от её ориентации. Определить отклик клетки на светлое пятно на тёмном фоне (центр пятна должен совпадать с центром рецептивного поля) в зависимости от размеров пятна.

3. Для модели простых и сложных клеток коры выполнить такое же исследование, как и для модели ганглиозных клеток. При этом следует уделить внимание изменению отклика при изменении ориентации полосы и её положения на рецептивном поле.

4. Проанализировать полученные результаты. Установить основные характеристики рецептивных полей модельных клеток. Определить степень их согласованности с данными о свойствах соответствующих биологических нейронов (симметричностью рецептивных полей, дирекционной избирательностью, реакцией на различные стимулы). Сделать выводы.

Сокращенный пример выполнения работы

Реакцию ганглиозных клеток можно представить как результат применения фильтра, ядро которого определяется как разность гауссиан с разной дисперсией. С помощью функций библиотеки OpenCV такая фильтрация может быть вычислена путем следующей серии вызовов

```
cvSmooth( image, gauss_d1, CV_GAUSSIAN, 3, 0, 0, 0 );
```

```
cvSmooth( image, gauss_d2, CV_GAUSSIAN, 7, 0, 0, 0 );
```

```
cvSub( gauss_d1, gauss_d2, ganglionic, NULL );
```

Здесь `image` – структура с исходным изображением, а `ganglionic` будет содержать поле реакций модельных ганглиозных клеток с одинаковой формой рецептивного поля, но разным положением центра поля на изображении. Изменение параметров (3, 7) при вычислении свертки с гауссианой приведет к изменению рецептивного поля моделируемых клеток.

Для проведения требуемого в работе эксперимента требуется исследование реакции одной ганглиозной клетки на разные стимулы. Сначала установим реакцию клетки на световое пятно малого размера. Это может быть сделано с помощью следующей программы.

```
#include "highgui.h"
```

```
#include "cv.h"
```

```
int main( int argc, char **argv )
```

```
{
```

```
    IplImage *image = NULL, *gauss_d1 = NULL,
```

```
        *gauss_d2 = NULL, *ganglionic = NULL, *res = NULL;
```

```

image = cvCreateImage( cvSize(256, 256), IPL_DEPTH_16S, 1 );
gauss_d1 = cvCreateImage( cvSize(256, 256), IPL_DEPTH_16S, 1 );
gauss_d2 = cvCreateImage( cvSize(256, 256), IPL_DEPTH_16S, 1 );
ganglionic = cvCreateImage( cvSize(256, 256), IPL_DEPTH_16S, 1 );
res = cvCreateImage( cvSize(256, 256), IPL_DEPTH_8U, 1 );
cvZero( res );
for(int i = 0; i < 13; i++) {
  for(int j = 0; j < 13; j++) {
    cvZero( image );
    cvCircle( image, cvPoint(j+128-6, i+128-6), 1,
      CV_RGB(127,127,127), -1, 8, 0 );
    cvSmooth( image, gauss_d1, CV_GAUSSIAN, 3, 0, 0, 0 );
    cvSmooth( image, gauss_d2, CV_GAUSSIAN, 7, 0, 0, 0 );
    cvSub( gauss_d1, gauss_d2, ganglionic, NULL );
    unsigned char v = (int)(((short *)
      (ganglionic->imageData+(128)*ganglionic->widthStep))[128])*3+128;
    cvCircle( res, cvPoint(j*16+32, i*16+32), 2, CV_RGB(v,v,v), 3, 8, 0 );
  }
}
cvNamedWindow( "Test OpenCV", CV_WINDOW_AUTOSIZE );
cvShowImage( "Test OpenCV", res );
cvSaveImage( "field.png", res );
cvWaitKey(0);
return 0;
}

```

Здесь фиксируются отклики клетки на «точечные» стимулы, находящиеся в разных местах рецептивного поля. Центр рецептивного поля клетки совпадает с центром изображения. Результат работы программы представлен на рис. 4.

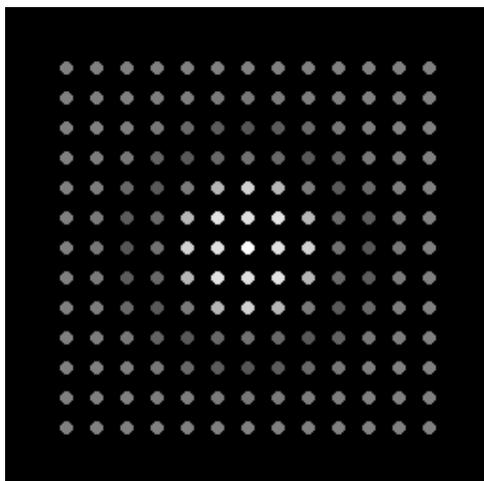


Рис. 4. Отклики модели ганглиозной клетки, полученные путем помещения светового пятна в разные точки рецептивного поля

Видно, что рецептивное поле является центрально-симметричным, причём максимальный отклик происходит на световое пятно, расположенное в центре поля.

Теперь используем стимул в форме светлого пятна на тёмном фоне и будем постепенно увеличивать размер этого пятна, регистрируя величину отклика. Для этого модифицируем программу следующим образом (показан только модифицированный цикл).

```
cvZero( res );
int v_prev = 0;
for(int i = 0; i < 30; i++) {
    cvZero( image );
    cvCircle( image, cvPoint(128, 128), i, CV_RGB(1000,1000,1000), -1, 8, 0 );
    cvSmooth( image, gauss_d1, CV_GAUSSIAN, 15, 0, 0, 0 );
    cvSmooth( image, gauss_d2, CV_GAUSSIAN, 31, 0, 0, 0 );
    cvSub( gauss_d1, gauss_d2, ganglionic, NULL );
    int v = (int)(((short*)(ganglionic->imageData+
        (128)*ganglionic->widthStep))[128]);
    printf("%d\n", v);
    cvLine( res, cvPoint(i * 8, 128-v_prev/5),
        cvPoint((i+1) * 8, 128-v/5), CV_RGB(255,255,255), 1, 8, 0 );
    v_prev = v;
}
```

Данная программа выводит следующие численные значения 18, 81, 184, 328, 420, 452, 422, 362, 280, 197, 130, 87, 57, 31, 16, 7, 3, 1, 0... Также ею построен график зависимости откликов от размера пятна, представленный на рис. 5.

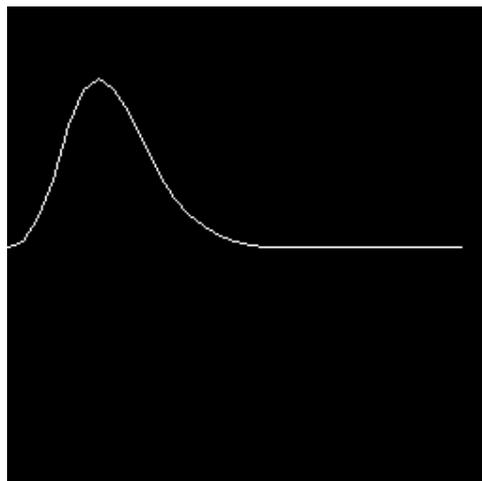


Рис. 5. Зависимость отклика модели ганглиозной клетки от размера стимула (светлого пятна на тёмном фоне)

Модифицируем программу так, чтобы вместо светлого пятна в качестве стимула использовалось кольцо. Для формирования такого стимула используем вызов

```
cvCircle( image, cvPoint(128, 128), i, CV_RGB(1000,1000,1000), 3, 8, 0 );
```

Модифицированная программа выдает следующие значения откликов
81, 269, 376, 435, 243, 100, -120, -252, -271, -278, -222, -168,
-126, -82, -54, -29, -17, -7, -2, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0.

Ею также строится график, представленный на рис. 6.

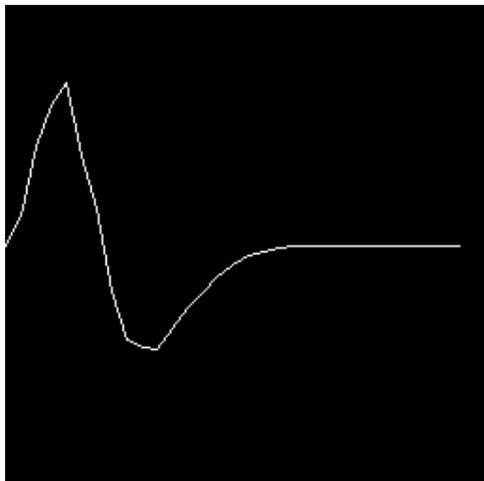


Рис. 6. Зависимость отклика модели ганглиозной клетки от размера стимула (светлого кольца на тёмном фоне)

Как видно из графика, при росте размера пятна сначала происходит увеличение активности модельной клетки, затем уменьшение, а когда пятно заполняет всё поле клетки, реакция исчезает. Это соответствует данным о работе ганглиозных клеток сетчатки с той поправкой, что биологические клетки имеют некоторый ненулевой уровень активности в спокойном состоянии, и их уровень активности не становится отрицательным (как в случае модельных клеток), а просто подавляется до нулевого уровня.

Используем в качестве стимула вместо окружности линию, проходящую через центр рецептивного поля:

```
cvLine( image, cvPoint(128+100*cos(i/57.2), 128+100*sin(i/57.2)),  
        cvPoint(128-100*cos(i/57.2), 128-100*sin(i/57.2)),  
        CV_RGB(1000,1000,1000), 1, 8, 0 );
```

Отклик оказывается небольшим и практически независимым от ориентации линии.

Сделаем выводы. Модель, эмулирующая работу ганглиозных клеток сетчатки, имеет центрально-симметричное рецептивное поле и не обладает дирекционной избирательностью. Максимальный отклик обнаруживается на светлое пятно некоторого размера на тёмном поле. При дальнейшем увеличении размера пятна отклик уменьшается. Подавление активности клетки наблюдается, когда в качестве стимула задается светлое кольцо на тёмном фоне. Данные результаты согласуются с основными свойствами ганглиозных клеток сетчатки.

Литература

1. Хьюбел, Д. Глаз. Мозг. Зрение. / Д. Хьюбел. – М.: Мир, 1990. – 240 с.
2. Марр Д. Зрение. Информационный подход к изучению представления и обработки зрительных образов / Д. Марр. – Пер. с англ. – М.: Радио и связь, 1987. – 400 с.

Вопросы для самопроверки:

1. Какие основные элементы включает зрительный тракт?
2. Какова информационная функция ганглиозных клеток сетчатки в процессе зрительного восприятия?
3. В чём различие между простыми и сложными клетками первичной зрительной коры?
4. Что такое рецептивное поле клетки? Какова форма рецептивных полей различных клеток зрительного тракта?
5. Какими способами можно промоделировать поведение клеток зрительного тракта?
6. Как реагируют различные клетки на стимулы с различной ориентацией и положением?

2. Исследование инвариантности методов сопоставления изображений в условиях изменчивости

Цель работы – ознакомиться с методами сопоставления изображений и исследовать их применимость к изображениям, подверженным различным типам изменчивости.

Задание по работе:

1. Изучить теоретическую часть работы.
2. Реализовать процедуры сопоставления изображений, основанные на амплитудно-фазовой и фазовой корреляции, а также преобразовании расстояния.
3. Провести тестирование процедур сопоставления с использованием изображений, полученных с разных ракурсов, в условиях изменения освещения, типа сенсора и собственной изменчивости объектов, присутствующих на сцене.

Теоретическая часть

Корреляционные методы сопоставления

В простейшем случае мера сходства двух изображений как функция взаимного смещения $(\Delta x, \Delta y)$ вычисляется как квадрат среднеквадратичного отклонения (СКО) яркостей соответствующих (с учётом величины смещения) пикселей

$$\Sigma_{f_1, f_2}(\Delta x, \Delta y) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} (f_1(x, y) - f_2(x - \Delta x, y - \Delta y))^2,$$

где f_1 и f_2 – изображения одного и того же объекта, отличающиеся только сдвигом; $N \times N$ – размер области, в которой вычисляется корреляция.

Минимизация критерия СКО эквивалентна максимизации значения корреляции:

$$C_{f_1, f_2}(\Delta x, \Delta y) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f_1(x, y) f_2(x - \Delta x, y - \Delta y).$$

Несложно убедиться, что положение корреляционного максимума при линейном глобальном преобразовании яркости $\alpha f_1(x, y) + \beta$ не меняется (только при $\alpha < 0$ максимум превращается в минимум). Таким образом, данный подход позволяет сопоставлять изображения при линейных глобальных искажениях яркости.

Существует возможность построения кросскорреляционного поля $C_{f_1, f_2}(\Delta x, \Delta y)$ за малое число операций, если воспользоваться быстрым преобразованием Фурье и тем фактом, что Фурье-образ

кросскорреляционного поля может быть получен как произведение спектров изображений.

По определению, прямое и обратное дискретные преобразования Фурье (ДПФ) вычисляются как

$$\tilde{f}(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-i \frac{2\pi}{N}(ux+vy)},$$

$$f(x, y) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{f}(u, v) e^{i \frac{2\pi}{N}(ux+vy)},$$

где u и v – пространственные частоты.

Кросскорреляционное поле тогда может быть получено как

$$C_{f_1, f_2}(\Delta x, \Delta y) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{f}_1(u, v) \tilde{f}_2^*(u, v) e^{i \frac{2\pi}{N}(u\Delta x + v\Delta y)} =$$

$$= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f_1(x, y) f_2^*(x - \Delta x, y - \Delta y).$$

Необходимо отметить, что в алгоритмах ДПФ изображение трактуется как периодический сигнал, то есть копии изображения как бы приставлены друг к другу, из-за чего появляется резкий переход от одного края изображения к другому. Это вызывает появление «креста» в центре спектра изображения (см. рис. 7) и ухудшение корреляционного отклика.

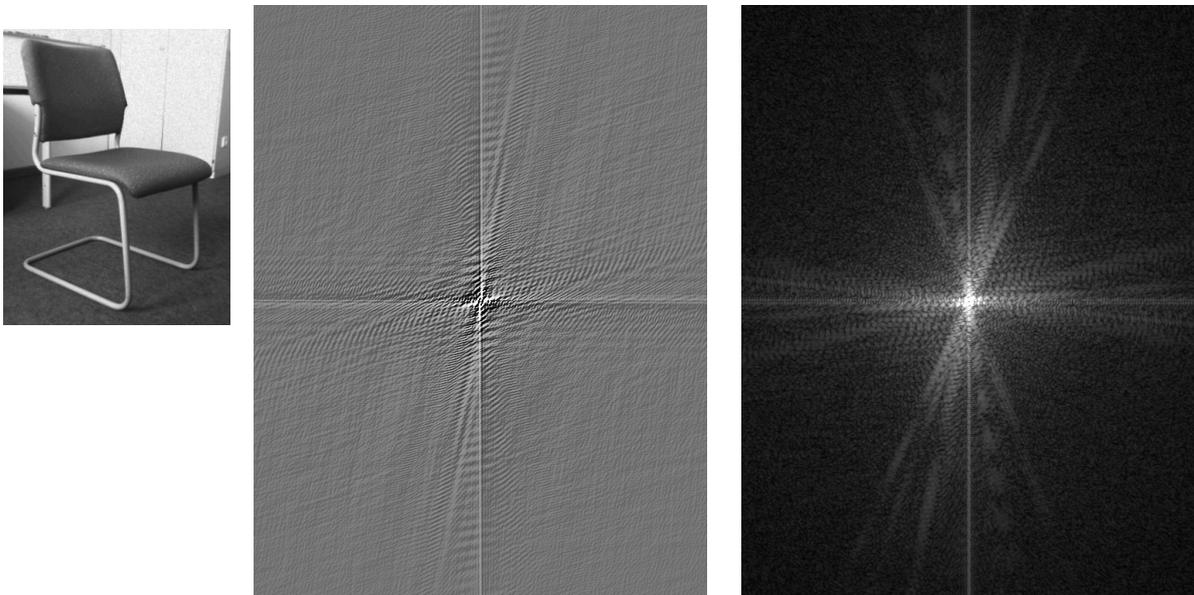


Рис. 7. Пример вычисления преобразования Фурье: (а) исходное изображение; (б) вещественная компонента спектра; (в) амплитудный (энергетический) спектр

Плавное приведение интенсивностей у краев к среднему значению устраняет эффект края. Как видно из рис. 8, центральный «крест», который отчетливо виден на рис. 7, пропадает. При этом также рекомендуется

расширять изображение, помещая его на поле однородной яркости (тогда коррелируемое с ним изображение не будет попадать на соседние периоды).

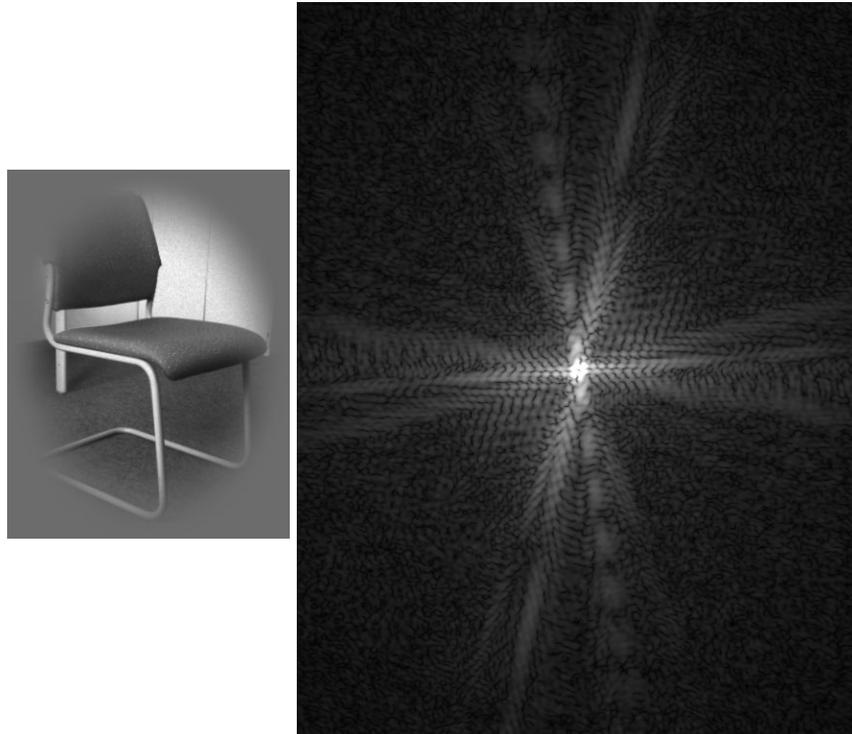


Рис. 8. Изображение, усредненное к краям, и его амплитудный спектр

Корреляционные методы, описанные выше, называются амплитудно-фазовыми, поскольку в них используется как амплитудная, так и фазовая спектральная информация. Помимо этого может применяться фазовая корреляция, при вычислении которой амплитудная информация не используется:

$$C_{f_1, f_2}^{\varphi}(\Delta x, \Delta y) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \frac{\tilde{f}_1(u, v)}{|\tilde{f}_1(u, v)|} \cdot \frac{\tilde{f}_2^*(u, v)}{|\tilde{f}_2^*(u, v)|} e^{i \frac{2\pi}{N}(u\Delta x + v\Delta y)}.$$

Фурье-образы $\tilde{f}_1(u, v)$ и $\tilde{f}_2(u, v)$ являются комплекснозначными, даже если исходные изображения представлены вещественнозначными функциями. Значения $|\tilde{f}_1(u, v)|$ и $|\tilde{f}_2(u, v)|$ являются модулями соответствующих комплексных величин.

Особенность фазовой корреляции заключается в том, что для неё автокорреляционное поле некоторой функции представляет собой дельта-функцию, в чём несложно убедиться

$$C_{f, f}^{\varphi}(\Delta x, \Delta y) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} e^{i \frac{2\pi}{N}(u\Delta x + v\Delta y)} = \begin{cases} 1, & (\Delta x, \Delta y) = (0, 0) \\ 0, & (\Delta x, \Delta y) \neq (0, 0) \end{cases}.$$

На рис. 9 приведен пример кросскорреляционного поля в случае фазовой корреляции. Видно, что здесь максимум может быть локализован с большей точностью (при этом видно, что в данном случае выделяется несколько максимумов, соответствующих сдвигу объекта на переднем плане и фона, имеющих разные смещения).

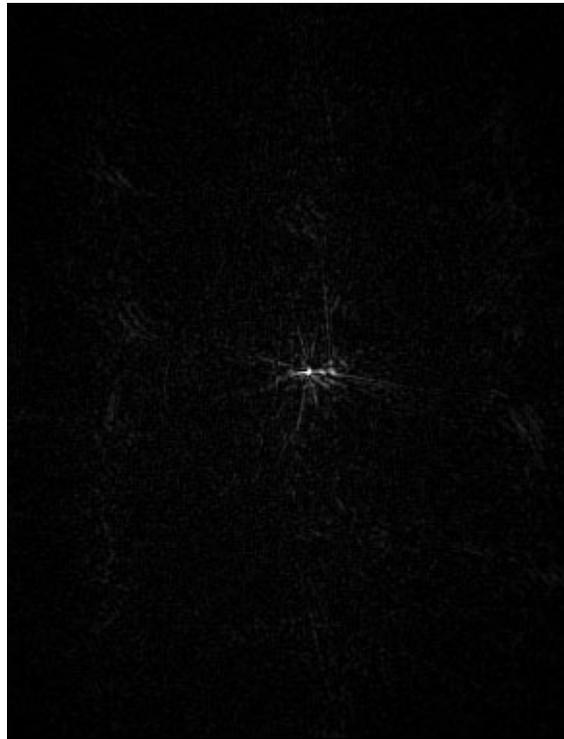


Рис. 9. Пример кросскорреляционного поля, получающийся в случае фазовой корреляции

Следует отметить, что в случае фазовой корреляции также полезным является сглаживание изображений к краю с дальнейшим расширением области изображений с заполнением области расширения средним значением.

Экспериментальная часть

В данной работе выполняют реализацию методов сопоставления и осуществляют их тестирование на парах изображений, подверженных преобразованиям различных типов, по отношению к которым необходимо определить степень инвариантности реализованных методов.

1. Реализовать методы амплитудно-фазовой и амплитудной корреляции. Первый метод может быть реализован как путем непосредственного вычисления коэффициента корреляции, так и через дискретное преобразование Фурье. Следует организовать как построение поля откликов используемой меры при разных смещениях, так и поиск

экстремума на этом поле с определением корректности найденного смещения.

2. Проверить корректность реализации программ, подавая на вход в качестве обоих изображений одно и то же изображение или его смещенный вариант.

3. Провести тестирование реализованных методов на изображениях, полученных в разные моменты времени, но не обладающих прочими отличиями.

4. Провести тестирование реализованных методов на изображениях, полученных с использованием различных типов сенсоров или обладающих существенной собственной изменчивостью объектов.

5. Сделать выводы о степени инвариантности разных методов по отношению к различным типам изменчивости.

Сокращенный пример выполнения работы

Реализуем методы амплитудно-фазовой и фазовой корреляции в форме следующей программы с использованием функций библиотеки OpenCV.

```
#include <math.h>
#include <stdio.h>
#include "highgui.h"
#include "cv.h"
#define PHASE_ONLY
void GetComplDFT( IplImage *image, IplImage *&image_f, int dftX, int dftY )
{
    int i, j;
    image_f = cvCreateImage( cvSize(dftX, dftY), IPL_DEPTH_32F, 2 );
    cvZero( image_f );
    for( i = 0; i < image->height; i++ ) {
        unsigned char *pc = (unsigned char *)
            ( image->imageData + i * image->widthStep );
        float *pf = (float *) ( image_f->imageData + i * image_f->widthStep );
        for( j = 0; j < image->width; j++ ) {
            pf[j * 2] = pc[j];
            pf[j * 2 + 1] = 0;
        }
    }
    cvDFT( image_f, image_f, CV_DXT_FORWARD, 0 );
#ifdef PHASE_ONLY
    for( i = 0; i < image_f->height; i++ ) {
        float *pf = (float *) ( image_f->imageData + i * image_f->widthStep );
```

```

for( j = 0; j < image_f->width; j++ ) {
    float v = sqrt( pf[j * 2] * pf[j * 2] + pf[j * 2 + 1] * pf[j * 2 + 1] );
    if( v > 1e-8 ) {
        pf[j * 2] /= v;
        pf[j * 2 + 1] /= v;
    }
}
}
}
#endif
}
int main( int argc, char **argv )
{
    int i, j, n, dftX, dftY;
    int shx, shy;
    double min_v = 1e+12, max_v = 0;
    IplImage *image1, *image2, *image_f1, *image_f2, *dst, *res;
    if( argc == 3 ) {
        image1 = cvLoadImage( argv[1], CV_LOAD_IMAGE_GRAYSCALE );
        image2 = cvLoadImage( argv[2], CV_LOAD_IMAGE_GRAYSCALE );
    } else return -1;
    if( image1 == NULL || image2 == NULL )
        return -2;
    dftX = cvGetOptimalDFTSize( image1->width );
    n = cvGetOptimalDFTSize( image2->width );
    if( n > dftX ) dftX = n;
    dftY = cvGetOptimalDFTSize( image1->height );
    n = cvGetOptimalDFTSize( image2->height );
    if( n > dftY ) dftY = n;
    GetComplDFT( image1, image_f1, dftX, dftY );
    GetComplDFT( image2, image_f2, dftX, dftY );
    res = cvCreateImage( cvSize(dftX, dftY), IPL_DEPTH_8U, 1 );
    dst = cvCreateImage( cvSize(dftX, dftY), IPL_DEPTH_32F, 2 );
    cvMulSpectrums( image_f1, image_f2, dst, CV_DXT_MUL_CONJ );
    cvDFT( dst, dst, CV_DXT_INV_SCALE, 0 );
    for( i = 0; i < res->height; i++ ) {
        float *pf = (float *) ( dst->imageData + i * dst->widthStep );
        for( j = 0; j < res->width; j++ ) {
            double v = sqrt( (pf[j*2])*(pf[j*2])+(pf[j*2+1])*(pf[j*2+1]) );
            if( v > max_v ) {
                shx = j;
                shy = i;
                max_v = v;
            }
        }
    }
}

```

```

    if( v < min_v ) min_v = v;
  }
}
if( shx > dftX/2 ) {
  shx -= dftX;
}
if( shy > dftY/2 ) {
  shy -= dftY;
}
for( i = 0; i < res->height; i++ ) {
  unsigned char *pc = (unsigned char *)
    ( res->imageData + i * res->widthStep );
  float *pf = (float *) ( dst->imageData + i * dst->widthStep );
  for( j = 0; j < res->width; j++ ) {
    double v = sqrt( (pf[j*2])*(pf[j*2])+(pf[j*2+1])*(pf[j*2+1]) );
    v = (v - min_v) * 255 / (max_v - min_v);
    pc[j] = ( v > 255 ? 255 : (unsigned char)v );
  }
}
cvNamedWindow( "Fourier", CV_WINDOW_AUTOSIZE );
cvShowImage( "Fourier", res );
cvSaveImage( "corr_field.png", res );
printf("Found shift: %d, %d\n", shx, shy);
cvWaitKey(0);
return 0;
}

```

Приведённая программа загружает два изображения, указанные в качестве её аргументов, вычисляет наиболее приемлемые размеры для вычисления дискретного преобразования Фурье. Далее с помощью реализованной в программе функции `GetCompIDFT` подготавливается изображение к вычислению преобразования Фурье через вызов `cvDFT`. Если включен флаг `PHASE_ONLY`, то осуществляется подавление амплитудной информации, что позволяет выполнять фазовую корреляцию. Затем с помощью функции `cvMulSpectrums` вычисляется произведение двух спектров, а вызов `cvDFT` с флагом `CV_DXT_INV_SCALE` позволяет получить кросскорреляционное поле. Далее производится поиск максимума на кросскорреляционном поле и нормировка этого поля для вывода в файл и на экран. Отрицательный сдвиг получается в том случае, когда та или иная координата положения максимума больше соответствующего половины размера кросскорреляционного поля.

Результаты исследования возможностей методов на разных изображениях показывают следующее.

При вычислении автокорреляционного поля каждого изображения программой всегда находится нулевой вектор смещения. При этом поле в случае амплитудно-фазовой корреляции может иметь сложный вид, в то время как в фазовой корреляции получается поле, содержащее только одну ненулевую точку (рис. 10). Это показывает, что программа работает корректно.



Рис. 10. Пример вычисления автокорреляционного поля изображения в случае амплитудно-фазовой и фазовой корреляции

Построение корреляционного поля для изображений, подверженных сдвигу и не содержащих существенных яркостных различий, приводит к результатам, аналогичным случаю автокорреляции, однако положение максимума оказывается сдвинутым. Например, для пары изображений, представленных на рис. 11, положение максимума для обоих методов оказывается (16, -12). При этом замечено, что при использовании изображений, размеры которых не соответствуют размерам, находящимся с помощью функции `cvGetOptimalDFTSize`, результаты оказываются значительно хуже, что связано с расширением изображений до соответствующих размеров и заполнением области расширения нулями.

Иными словами, в приведенной выше реализации не принята во внимание необходимость устранения эффекта края.

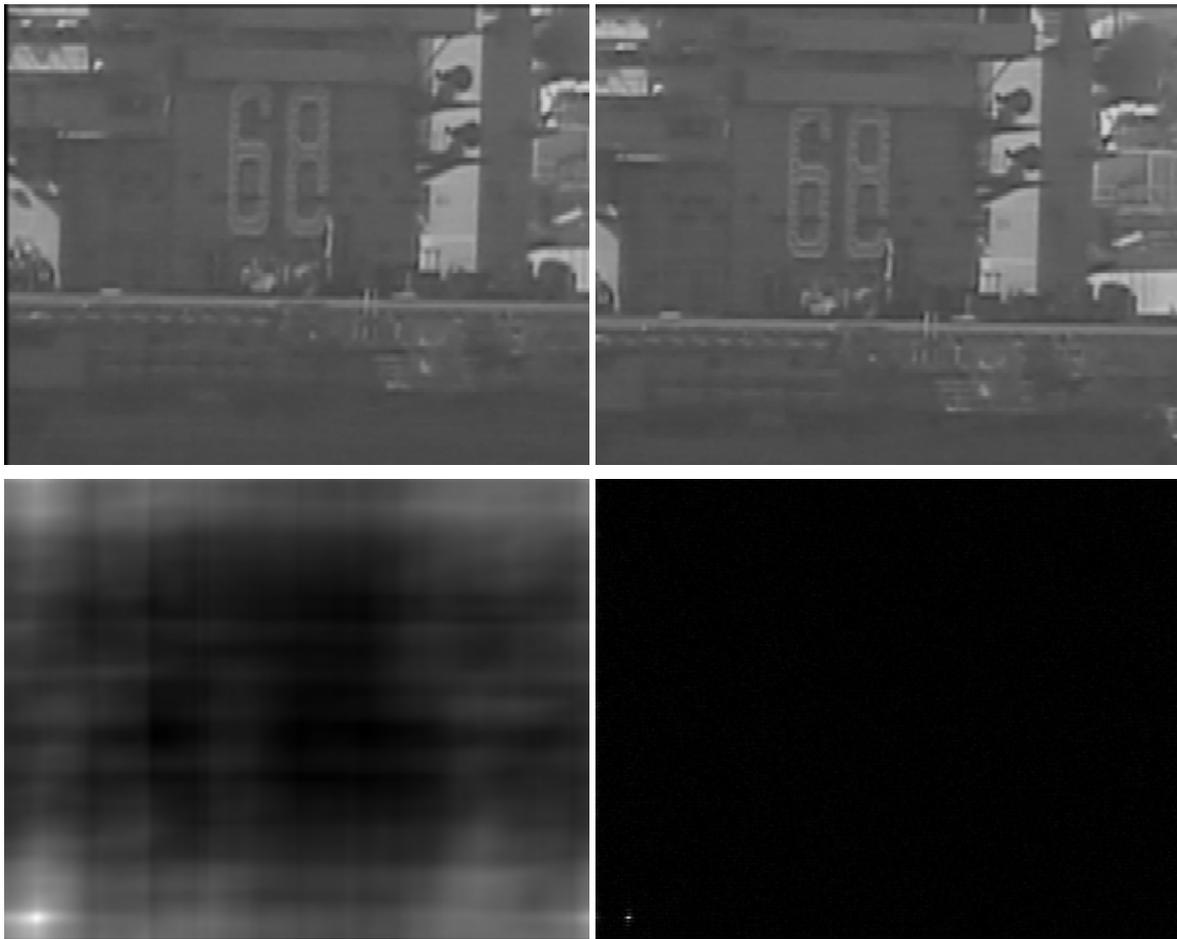


Рис. 11. Пример вычисления кросскорреляционного поля для двух изображений, не содержащих существенных яркостных искажений

Реализованные корреляционные методы оказываются способными находить правильное взаимное смещение изображений, достаточно сильно отличающихся друг от друга по яркостным характеристикам, если на этих изображениях присутствуют стабильные объекты, дающие заметный вклад в энергию спектра. В частности, на рис. 12 представлен пример подобной пары изображений с успешным выполнением корреляции (в обоих вариантах корреляции находится смещение (12, б), соответствующее истинному сдвигу). В целом, однако, яркостные искажения заметно снижают вероятность обнаружения правильного смещения.

Исследование программы показывает, что способность нахождения истинного смещения корреляционными методами быстро нарушается даже при небольших вращениях и масштабных искажениях изображений. Также на надежность заметно влияет, какая доля площади изображений перекрывается.

Можно заключить, что корреляционные методы применимы в случаях, когда нет взаимного вращения и масштабирования изображений и не слишком значительны яркостные искажения.

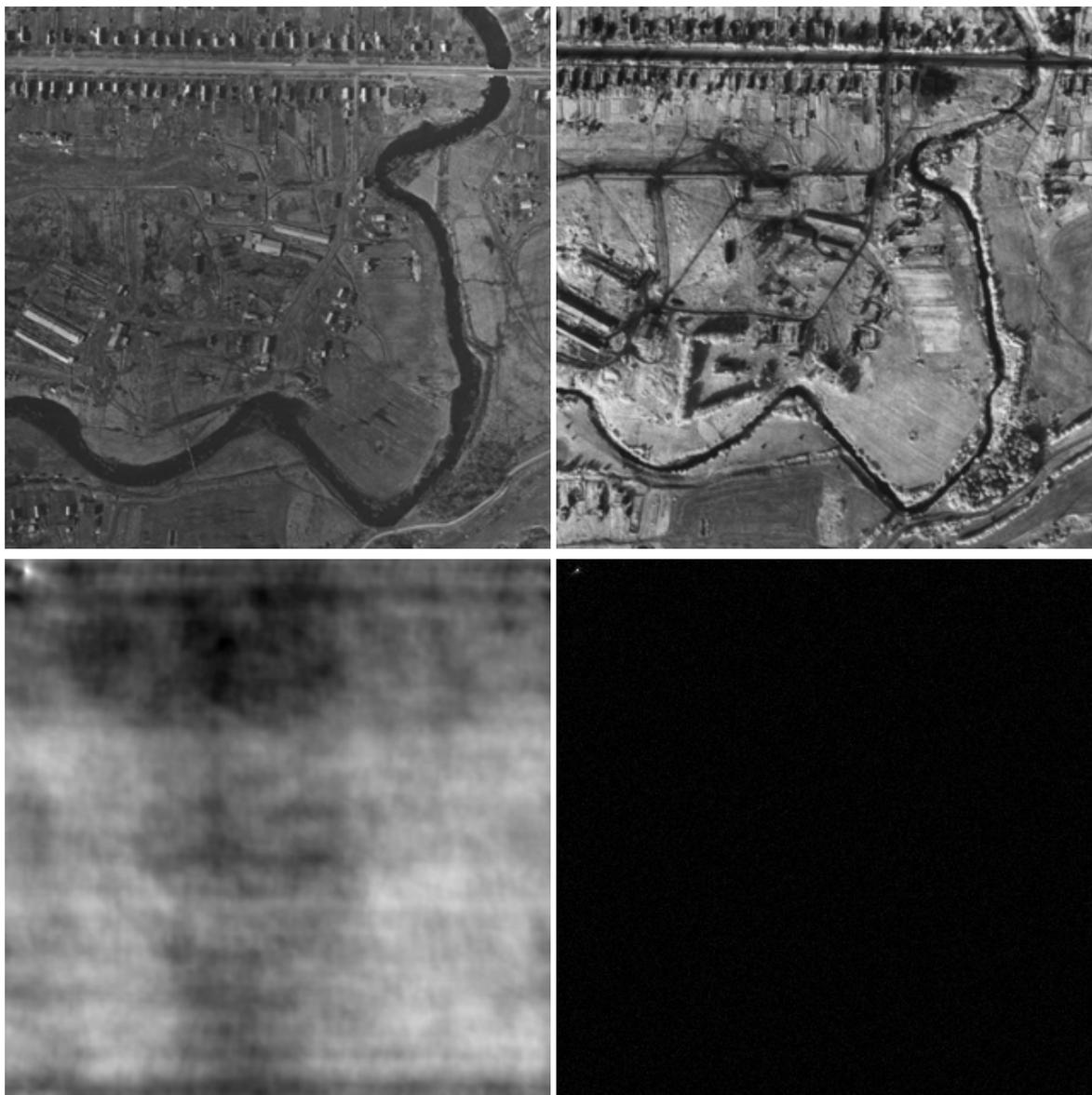


Рис. 12. Пример вычисления кросскорреляционного поля для двух изображений, содержащих яркостные искажения

Литература

1. Прэтт, У. **Цифровая обработка изображений. Кн. 2** / У. Прэтт. – М.: Мир, 1982. – 480 с.

Вопросы для самопроверки:

1. В чём преимущество использования преобразования Фурье в корреляционных методах сопоставления? Какие при этом возникают дополнительные трудности?
2. В каких случаях предпочтительнее использовать непосредственное ("по определению") вычисление корреляционного поля?
3. Чем процедура фазовой корреляции отличается от процедуры амплитудно-фазовой корреляции?
4. Какова автокорреляционная функция в случае фазовой корреляции?
5. Какова вычислительная сложность корреляционных алгоритмов?
6. Какими свойствами обладает амплитудный пространственный спектр?

3. Построение карт стереодиспаратности

Цель работы – изучить корреляционные методы стереозрения для получения плотных карт глубины, установить особенности и ограничения их использования; исследовать влияние используемых размеров окна корреляции и диапазона допустимых диспаратностей.

Задание по работе:

1. Изучить теоретическую часть работы.
2. Реализовать корреляционный алгоритм стереозрения с использованием корреляции Пирсона.
3. Провести исследование разработанного алгоритма на нескольких стереопарах с различными характерными значениями диспаратности, используя разные размеры корреляционного окна и значения диапазона допустимых диспаратностей.

Теоретическая часть

Задача сопоставления изображений в случае, когда съемка производится с двух известных ракурсов, отличающихся только сдвигом по направлению, перпендикулярному оптической оси (см. рис. 13), называется задачей *стереозрения*, а расстояние между камерами – *стереобазой*. При этом оптические оси камер полагаются параллельными. В общей постановке задача стереозрения может рассматриваться для двух произвольных (но известных) ракурсов, однако, как правило, этот случай принципиальных отличий от указанной выше упрощенной постановки не имеет.



Рис. 13. Пример стереопары

Стереозрение является эффективным средством восстановления трёхмерной формы объектов и расстояний до них, что широко используется в биологических системах зрительного восприятия. В

частности, осуществляющееся в зрительной коре объединение информации, поступающей от двух глаз, помогает человеку воспринимать мир трёхмерным (хотя это и не единственный работающий механизм).

Использование стереозрения оправдано и с технической точки зрения, например, для обнаружения препятствий на дорогах. Смещение объекта в плоскости изображения на стереопаре происходит вдоль той же оси, вдоль которой смещена камера. Это смещение называется *диспаратностью*. Значение диспаратности зависит только от дальности (глубины) до объекта, но не направления на него. При этом диспаратность не должна быть отрицательной (при сонаправленных оптических осях камер), и она равна нулю при бесконечной (достаточно большой) удалённости объекта.

Задача стереозрения сводится к отождествлению точек левого изображения (полученного с левой камеры) с точками правого изображения (полученного с правой камеры) при ограничении, заключающемся в том, что сопряженные точки лежат на одной и той же горизонтальной линии. Как правило, также известна информация о диапазоне возможных диспаратностей. Поясняющая схема изображена на рис. 14.

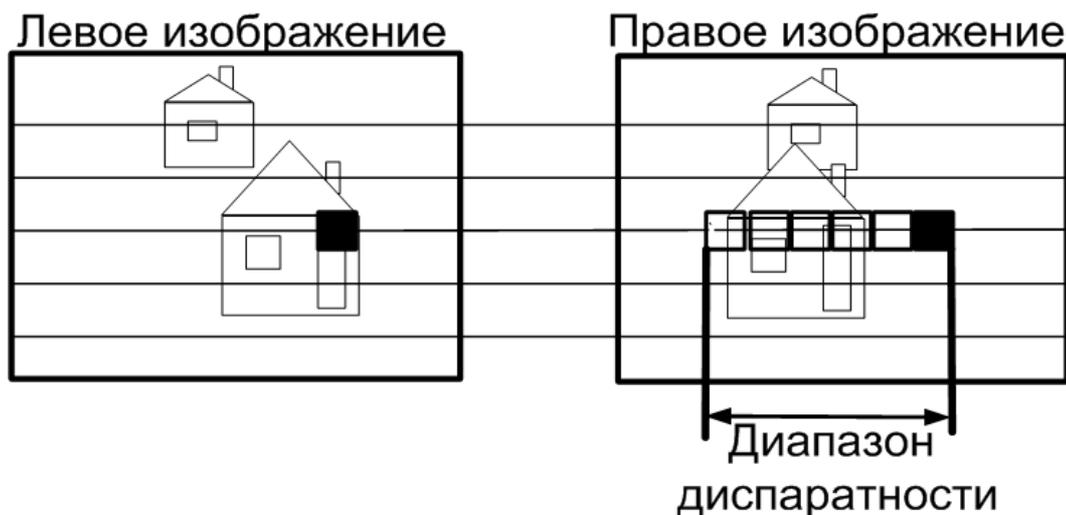


Рис. 14. На правом изображении отмечены регионы-кандидаты, среди которых надо искать область, соответствующую квадрату, выделенному чёрным цветом на левом изображении

Поскольку в этом случае диспаратность объекта обратно пропорциональна расстоянию до него, в результате такого отождествления при известной стереобазе и прочих параметрах камер может быть восстановлено расстояние до объектов сцены.

Отождествление точек на паре изображений может производиться в рамках различных представлений изображений: низкоуровневых (пиксельных), контурных или структурных.

Карта глубины (дальности) – это изображение, в котором яркость пикселей пропорциональна расстоянию от матрицы фотоприемника до

объектов реальной трёхмерной сцены, снимаемой с помощью стереокамеры. Карты глубины, в которых расстояние удаётся получить до каждого пикселя исходного изображения, называют плотными. Поскольку в стереозрении глубина обратно пропорциональна диспаратности, то можно также говорить о *карте диспаратности*. На рис. 15 приведен пример карты диспаратности.

Как правило, для построения плотных карт глубины применяются корреляционные методы. В общем случае такие методы используют два предположения:

1. Области вокруг пикселей, соответствующих друг другу на двух изображениях стереопары, имеют высокое значение кросскорреляции.
2. Карты глубины реальных трёхмерных сцен имеют малое количество резких перепадов уровня глубины. Такие перепады существуют только на границах объектов.

Далее мы не будем рассматривать способы учёта гладкости карты глубины, а остановимся подробнее на расчете корреляции.



Рис. 15. Карта диспаратности для стереопары, изображенной на рис. 13

Сложность отождествления двух точек, являющихся проекциями одной точки физического объекта на двух матрицах фотоприемников, обычно связана с тем, что одиночные пиксели не уникальны и, как правило, два пикселя, соответствующие друг другу на двух изображениях, имеют разные значения яркости. В связи с этим принято сопоставлять не индивидуальные пиксели, а области вокруг каждого пикселя. Значения яркостей пикселей в этих областях можно рассматривать как значения случайной величины и в таком случае использовать коэффициент корреляции Пирсона для двух случайных величин:

$$r_{XY} = \frac{\text{cov}_{XY}}{\sigma_X \sigma_Y} = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2 \sum_i (Y_i - \bar{Y})^2}},$$

где r_{XY} – это коэффициент корреляции, X_i – значение яркостей пикселей окна на левом изображении, а Y_i – на правом. Значение r_{XY} лежит в интервале от -1 до 1 .

Если значение коэффициента корреляции положительно, то случайные величины X и Y имеют прямую корреляционную связь. Как правило, при незначительных изменениях ракурса съемки одним и тем же объектам на двух изображениях стереопары соответствуют области с высоким коэффициентом корреляции. Поэтому задачу отождествления точек можно рассматривать как задачу поиска области (окна) на правом изображении, имеющего максимальный коэффициент корреляции с областью на левом изображении.

Алгоритм стереозрения в этом случае можно представить как последовательность из двух шагов. Сначала вычисляется корреляция области вокруг каждого пикселя левого изображения со всеми возможными областями правого изображения с учётом ограничений стереозрения. В результате получается так называемая матрица корреляционной энергии, каждый элемент которой показывает насколько сильно коррелируют окна двух изображений с определённым сдвигом по оси абсцисс. Затем по этой матрице каждой точке левого изображения ставится в соответствие точка правого с наибольшим коэффициентом корреляции областей вокруг них. Таким образом, определяется диспаратность всех пикселей левого изображения. Вместо простой максимизации коэффициента корреляции может использоваться и более сложная оптимизирующая функция, учитывающая, например, степень гладкости карты глубины. Выбор же размеров корреляционного окна зависит от требуемой точности на границах объектов, скорости работы алгоритма и необходимой устойчивости к шумам.

Экспериментальная часть

В данной работе осуществляются разработка и исследование метода стереозрения, основанного на определении диспаратности точек путем поиска максимального коэффициента корреляции областей вокруг пикселей. Работа состоит из следующих шагов.

1. Реализовать программу, вычисляющую диспаратность для каждого пикселя левого изображения стереопары.
2. Проверить работоспособность программы на нескольких стереопарах. Следует убедиться, что карта глубины (диспаратности) адекватна изображенной на стереопаре трёхмерной сцене.

3. Исследовать реализованную программу. Следует исследовать влияние изменения размера корреляционного окна и диапазона допустимых диспаратностей на точность получаемой карты диспаратности.

4. Проанализировать полученные результаты. Сделать выводы о работоспособности корреляционного алгоритма стереозрения, его ограничениях и недостатках. Следует выявить ошибки алгоритма в определении диспаратности для конкретных случаев и объяснить причину их появления.

Сокращенный пример выполнения работы

Для решения задачи стереозрения с помощью корреляционного метода реализована следующая программа, использующая функции библиотеки OpenCV.

```
#include <highgui.h>
#include <cv.h>
double correlation(IplImage* imgL,IplImage* imgR,int x,int y, int dx,
                 int w_size)
{
    unsigned char* a = (unsigned char*)imgL->imageData;
    unsigned char* b = (unsigned char*)imgR->imageData;
    double mean1=0, mean2=0, covxy=0, std1=0,std2=0;
    double sq1, sq2, corr, ela, elb;
    int tx, ty, tdx, n=0;
    int s = w_size/2;
    int w = imgL->width;
    int h = imgL->height;
    for(int j = -s; j <= s; j++){
        for(int i = -s; i <= s; i++){
            tx = x+i;
            tdx = tx+dx;
            ty = y+j;
            if(tx < 0 || tx >= w || tdx < 0 || tdx >= w || ty < 0 || ty >= h) continue;
            mean1+= (double)(a[ty*imgL->widthStep + tx]);
            mean2+= (double)(b[ty*imgL->widthStep + tdx]);
            n++;
        }
    }
    if( n<w_size*w_size/2){
        return -1;
    }
}
```

```

mean1/=n;
mean2/=n;
for(int j = -s; j <= s; j++){
    for(int i = -s; i <= s; i++){
        tx = x+i;
        tdx = tx+dx;
        ty = y+j;
        if(tx < 0 || tx >= w || tdx < 0 || tdx >= w || ty < 0 || ty >= h) continue;
        ela = (double)a[ty*imgL->widthStep + tx];
        elb = (double)b[ty*imgL->widthStep + tdx];
        sq1 = (ela-mean1);
        sq2 = (elb-mean2);
        covxy+=sq1*sq2;
        std1+=sq1*sq1;
        std2+=sq2*sq2;
    }
}
corr = covxy/(sqrt(std1*std2));
return corr;
}
int compute_cmatrix(IplImage* imgL, IplImage* imgR, double** cor_matr,
                    int dMin, int dMax)
{
    double* cor_matrix = *cor_matr;
    int D = dMax-dMin+1;
    int W = imgL->width;
    for(int i = 0; i<imgL->height; i++){
        for(int j = 0; j<imgL->width; j++){
            for(int d=0;d<D;d++){
                cor_matrix[(i*W+j)*D+d] = correlation(imgL, imgR, j, i, -(d + dMin), 7);
            }
        }
    }
    return 1;
}
int findMax(double* srcAry, int aryLth)
{
    int maxIdx=0;
    double maxVal=-1000;
    for (int i=0; i<aryLth; i++){
        if(srcAry[i] > maxVal){
            maxVal = srcAry[i];
            maxIdx = i;
        }
    }
}

```

```

    }
}
return maxIdx;
}
int main(int argc, char *argv[])
{
    IplImage *imgL = 0;
    IplImage *imgR = 0;
    int dMin=0, dMax=80;
    int D = dMax-dMin+1;
    double scale=0;
    double* cor_matrix;
    imgL = cvLoadImage(argv[1], 0);
    imgR = cvLoadImage(argv[2], 0);
    cor_matrix = (double*)malloc(imgL->width*imgL->height*D
                                *sizeof(double));
    compute_cmatrix(imgL, imgR, &cor_matrix, dMin, dMax);
    IplImage* dispimgL = cvCreateImage(cvSize(imgL->width, imgL->height),
                                      8, 1);

    scale = 255.0/D;
    for (int i=0; i<imgL->height; i++) {
        for (int j=0; j<imgL->width; j++) {
            int max= findMax(cor_matrix+(i*imgL->width+j)*D, D);
            dispimgL->imageData[i*imgL->widthStep+j] = (max)*scale;
        }
    }
    cvSmooth(dispimgL, dispimgL, CV_MEDIAN, 3, 3, 0, 0);
    cvSaveImage("disp.png", dispimgL);
    cvReleaseImage(&imgL);
    cvReleaseImage(&imgR);
    free(cor_matrix);
    return 0;
}

```

В данной программе сначала производится загрузка двух изображений из файлов, имена которых переданы в командной строке в качестве аргументов. Затем с помощью функции `compute_cmatrix` вычисляется матрица корреляционной энергии. Для каждой возможной позиции корреляционного окна ищется коэффициент корреляции с окном, сдвинутым вдоль горизонтальной линии, на правом изображении. Величины `dMin` и `dMax` задают в пикселях диапазон допустимых диспаратностей, который лежит между этими двумя величинами. После этого осуществляется поиск окон с максимальным коэффициентом корреляции. Для этого используется функция `findMax`, которая ищет

смещение окна, которому соответствует максимальное значение в матрице корреляционной энергии. Параметр `scale` используется для того, чтобы привести карту диспаратности `dispimgL` к изображению с интенсивностями пикселей от 0 до 255.

Функция `correlation` отвечает за вычисление корреляции Пирсона. Среди прочих она имеет параметр `w_size`, который задает размер корреляционного окна в пикселях. Поскольку необходимо ориентировать центр окна относительно каждого индивидуального пикселя, то параметр `w_size` должен иметь нечётное значение. На последнем шаге алгоритма выполняется сглаживание полученной карты глубины с помощью медианного фильтра с окном 3 (`cvSmoothImage`).

Было проведено исследование разработанной программы на стереопарах при разных параметрах алгоритма. Тестовая стереопара приведена на рис. 13.

На рис. 16 приведен результат вычисления значений диспаратности при `dMin = 0`, `dMax = 80`, `w_size = 7`. Видно, что в целом карта диспаратности адекватна изображенной на стереопаре трёхмерной сцене – предметы, находящиеся ближе к камере, имеют более яркие цвета. Тем не менее, присутствует большое количество шумов.

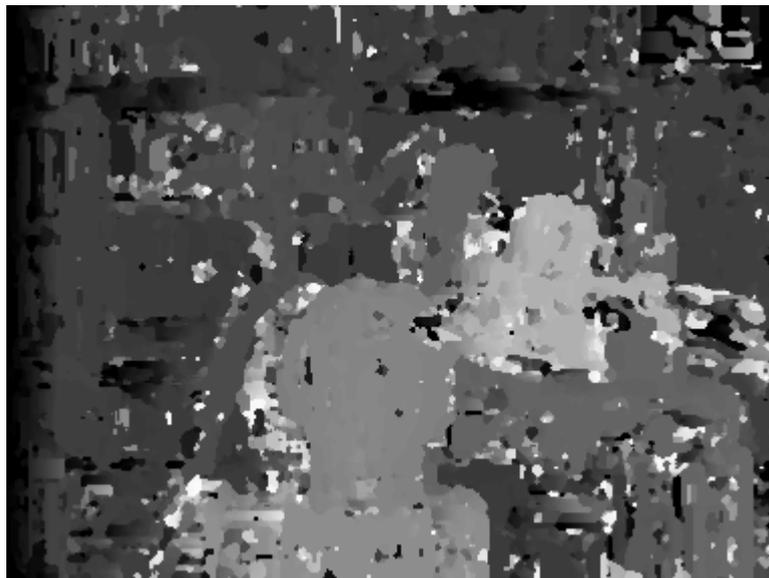


Рис. 16. Карта диспаратности при размере корреляционного окна = 7 и диапазоне допустимых диспаратностей [0,80]

Для того чтобы подавить их, изменим размер корреляционного окна на больший и диапазон допустимых диспаратностей на более точный.

Для этого изменим следующие строчки программы:

```
cor_matrix[(i*W+j)*D+d] = correlation(imgL,imgR, j, i,-(d + dMin), 13);
```

и

```
int dMin=10, dMax=60;
```

На рис. 17 представлен результат вычисления значений диспаратности при размере корреляционного окна $w_size = 13$ и диапазоне диспаратности $[10,60]$.

В целом карта диспаратностей улучшается за счет уменьшения количества шумов – корреляция в большом окне дает более помехоустойчивую оценку. В исследованных примерах выбор более точного диапазона диспаратностей для каждой конкретной стереопары также уменьшает количество ошибок, т.к. уменьшается количество возможных кандидатов для сопоставления.



Рис. 17. Результат построения карты диспаратности при размере корреляционного окна $w_size= 13$, и диапазоне допустимых диспаратностей $[10,60]$

Однако мы можем видеть ошибки – «сложными» оказываются зоны окклюзии. Таким зонам нельзя найти соответствия на стереопарах. Увеличение размеров окна корреляции приводит также к более «грубой» работе алгоритма на границах объектов. Также видны ошибки, например, на тонкой ножке лампы. Это можно объяснить, например, тем, что корреляционное окно оказывается слишком большим и в него попадают предметы с разным уровнем глубины. Можно говорить о том, что для преодоления этих ошибок необходима более сложная обработка. Например, при отождествлении пикселей можно учитывать степень гладкости карты диспаратности, а также дополнительно анализировать изображение на иерархически более высоком уровне с привлечением контурного и структурного представления изображений.

Литература

1. Хорн Б.К.П. Зрение роботов / Б.К.П. Хорн. – Пер. с англ. – М.: Мир, 1989. – 489 с.
2. Шапиро Л. Компьютерное зрение: Учеб. пособие для вузов / Л. Шапиро, Дж. Стокман. – Пер. с англ. – БИНОМ. Лаборатория знаний, 2006. – 752 стр.

Вопросы для самопроверки:

1. В чём заключается задача стереозрения?
2. Какая величина называется стереодиспаратностью? Как она связана с расстоянием до объектов трёхмерной сцены?
3. В каком случае при известной диспаратности может быть восстановлено абсолютное расстояние до объектов трёхмерной сцены?
4. Какова сущность корреляционных методов стереозрения для построения плотных карт глубины?
5. Какая мера может быть использована при отождествлении точек в задаче стереозрения? Какие ограничения накладываются на положение отождествляемых точек?
6. Каким ошибкам подвержены корреляционные алгоритмы стереозрения? Как можно бороться с ними?

4. Анализ структуры оптического потока

Цель работы – ознакомиться с методами восстановления оптического потока и исследовать их применимость к естественным изображениям в случае движения объектов сцены или камеры.

Задание по работе:

1. Изучить теоретическую часть работы.
2. Реализовать низкоуровневые, контурные или признаковые методы восстановления оптического потока с использованием предположения о гладкости оптического потока.
3. Исследовать возможности корректного определения скоростей при различных условиях движения и выбранных параметров метода восстановления.

Теоретическая часть

Оценивание оптического потока

Пусть изображение задано как изменяющееся во времени поле яркости $f(x, y, t)$. В пространственно-временной окрестности точки (x, y, t) это поле можно разложить в ряд Тейлора:

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial f}{\partial t} \Delta t + \varepsilon,$$

где ε – сумма слагаемых, содержащих частные производные вторых и более высоких степеней.

Надо найти смещение окрестности (x, y) изображения в момент времени t в равную по интенсивности окрестность $(x + \Delta x, y + \Delta y)$ в момент времени $t + \Delta t$. Следовательно

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t).$$

При малых изменениях $\Delta x, \Delta y, \Delta t$ можно полагать $\varepsilon \approx 0$. Это допустимо при условии, что точки изображения на соседних кадрах смещаются не более чем на 1 пиксель. В итоге получаем условие

$$\frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial f}{\partial t} \Delta t = 0,$$

которое можно преобразовать к виду

$$\frac{\partial f(x, y, t)}{\partial x} v_x(x, y, t) + \frac{\partial f(x, y, t)}{\partial y} v_y(x, y, t) + \frac{\partial f(x, y, t)}{\partial t} = 0,$$

где $v_x = \Delta x / \Delta t$, $v_y = \Delta y / \Delta t$ – оценки скоростей перемещения точек на изображении. Представленное выше уравнение называется *уравнением оптического потока*.

В этом уравнении мы можем для каждого момента времени и каждой точки изображения оценить все частные производные $\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial t}\right)$. Для оценки частной производной по времени необходимо иметь, по крайней мере, два кадра видеоряда.

Таким образом, видно, что для каждой точки изображения одно уравнение оптического потока связывает две неизвестные v_x и v_y , которые однозначно восстановить нельзя. Восстанавливается лишь проекция вектора скорости на направление пространственного градиента яркости $\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$. Действительно, несложно убедиться, что если пара значений $v_x^{(0)}$ и $v_y^{(0)}$ являются решением уравнения оптического потока, то пара $v_x^{(0)} - \alpha \frac{\partial f}{\partial y}$ и $v_y^{(0)} + \alpha \frac{\partial f}{\partial x}$ также будет решением при любом значении α .

Кроме того, если точка изображения располагается внутри области постоянной яркости, то все три частные производные (по x , y и t) будут равны нулю. Следовательно, для таких точек вообще никакая информация о направлении движения не будет доступна.

Таким образом, определение оптического потока с использованием стандартного уравнения оптического потока сопряжено с существенной неоднозначностью. Наличие шумов на изображении приводит к заметному искажению оцененных значений скоростей.

Доопределение оптического потока может быть сделано путем интерполяции измеренных значений. Измерение значений скорости окажется наиболее надежным в точках с максимальным модулем градиента яркости, то есть в контурных точках. В точках контура будет определяться только компонента скорости в направлении, перпендикулярном контуру. Однако контур, как правило, имеет изломы. Углы, располагающиеся на контуре, имеют в своей окрестности точки с разным направлением градиента яркости, а значит, для них несложно восстановить полный вектор скорости, пользуясь предположением непрерывности поля скоростей.

Отождествление сопряженных точек

Уравнение оптического потока непосредственно применимо только при малых перемещениях точек на соседних кадрах, поскольку даже при извлечении контуров и обнаружении на них углов (что соответствует уже не низкоуровневым, а контурным и структурным представлениям) оценка скорости получается из уравнения оптического потока, опирающегося на яркостные характеристики.

При значительных перемещениях в рамках низкоуровневого подхода можно было бы строить изолинии (линии постоянной яркости) и определять, куда произошло их перемещения. Однако несложно убедиться, что при вращении объекта яркость его точек будет меняться (за исключением случая рассеянного освещения, который на практике бывает не так часто). Таким образом, на разных кадрах видеоряда изолинии с одинаковой яркостью не будут состоять из сопряженных точек.

Поскольку яркость точек не является хорошей характеристикой для их отождествления, необходимо привлекать представления более высоких уровней. Вместо изолиний можно использовать контуры.

При отождествлении точек контуров на изображениях видеоряда используются те же предположения, что и в низкоуровневых методах оценивания оптического потока. Во-первых, перемещения полагаются не слишком большими. Во-вторых, считается, что оптический поток имеет мало разрывов. Это говорит о том, что поиск сопряженной точки должен осуществляться в окрестности текущей точки, причём для соседних контурных точек сдвиги должны быть близкими. Иными словами, можно выполнять чамферное сопоставление каждого фрагмента контура первого изображения с картой расстояний, построенной на основе контурных точек второго изображения.

Очевидно, если фрагмент соответствует прямолинейному участку контура, то мера Хаусдорфа не будет меняться при перемещениях вдоль контура. Таким образом, и здесь полный вектор скорости может быть определён только для углов. Отличие от рассмотренного ранее случая заключается в способе оценивания скорости. Здесь вместо уравнения оптического потока используется непосредственная оценка перемещения фрагмента контура.

При простейшей реализации данного подхода к оцениванию оптического потока выделяются углы на последовательных изображениях видеоряда, и эти углы ставятся друг другу в соответствие на основе близости их положений. При малых перемещениях такой подход может давать приемлемые результаты, однако при увеличении перемещений отождествление угловых точек только на основе их близости становится невозможным. Дополнительная трудность связана с тем, что некоторые угловые точки могут пропадать или, напротив, появляться, так что для них нет сопряженных точек среди углов, выделенных на другом изображении.

Для увеличения надежности сопоставления могут применяться различные приемы. В рамках признакового подхода каждая ключевая точка описывается вектором инвариантных признаков, на основе которых и производится её отождествление. Здесь могут применяться те же признаки, что и при сопоставлении изображений трёхмерных сцен.

Экспериментальная часть

В данной работе осуществляют разработку и исследование метода восстановления оптического потока, основанного на отождествлении точек интереса или углов на контурах с использованием предположения о гладкости оптического потока.

1. Реализовать программу, определяющую движения точек интереса на изображении.

2. Проверить работоспособность программы на изображениях сцены в случае отсутствия движения. Следует убедиться, что в результате оптический поток оказывается нулевым.

3. Исследовать реализованную программу. При этом следует исследовать влияние различных компонентов скорости (движения, вращения) и их абсолютной величины на возможность определения корректных значений скоростей. Исследовать влияние параметров алгоритма (в частности, предположение о гладкости оптического потока) на результаты.

4. Проанализировать полученные результаты. Сделать выводы о работоспособности реализованного метода восстановления оптического потока и возможности его улучшения.

Сокращенный пример выполнения работы

Для определения векторов перемещения точек на паре последовательных кадров видеоряда реализована следующая программа.

```
#include "highgui.h"
#include "cv.h"
int main( int argc, char **argv )
{
    int i = 0, j;
    IplImage *image1, *image2, *res;
    IplImage *eig1, *temp1, *eig2, *temp2;
    const int MAX_COUNT = 500;
    if( argc == 3 ) {
        image1 = cvLoadImage( argv[1], CV_LOAD_IMAGE_GRAYSCALE );
        image2 = cvLoadImage( argv[2], CV_LOAD_IMAGE_GRAYSCALE );
    } else return -1;
    if( image1 == NULL || image2 == NULL )
        return -2;
    eig1 = cvCreateImage( cvGetSize(image1), 32, 1 );
    temp1 = cvCreateImage( cvGetSize(image1), 32, 1 );
    int count1 = MAX_COUNT;
```

```

CvPoint2D32f *points1 =
    (CvPoint2D32f*)cvAlloc( MAX_COUNT*sizeof(*points1) );
cvGoodFeaturesToTrack( image1, eig1, temp1, points1, &count1,
    0.1, 10, 0, 3, 0, 0.04 );
eig2 = cvCreateImage( cvGetSize(image2), 32, 1 );
temp2 = cvCreateImage( cvGetSize(image2), 32, 1 );
int count2 = MAX_COUNT;
CvPoint2D32f *points2 =
    (CvPoint2D32f*)cvAlloc( MAX_COUNT*sizeof(*points2) );
cvGoodFeaturesToTrack( image2, eig2, temp2, points2, &count2,
    0.1, 10, 0, 3, 0, 0.04 );
res = cvCreateImage( cvGetSize(image2), 8, 3 );
cvCvtColor( image2, res, CV_GRAY2RGB );
for( i = 0; i < count1; i++ ) {
    double min_dist = 30*30;
    int best_j = -1;
    for( j = 0; j < count2; j++ ) {
        double dist = (points1[i].x-points2[j].x)*(points1[i].x-points2[j].x)+
            (points1[i].y-points2[j].y)*(points1[i].y-points2[j].y);
        if( dist < min_dist ) {
            min_dist = dist;
            best_j = j;
        }
    }
    if( best_j != -1 ) {
        cvCircle( res, cvPoint(points2[best_j].x, points2[best_j].y), 3,
            CV_RGB(0,0,255), -1 );
        cvLine( res, cvPoint(points1[i].x,points1[i].y),
            cvPoint(points2[best_j].x, points2[best_j].y), CV_RGB(255,0,0) );
    }
}
cvNamedWindow( "points", 1 );
cvShowImage( "points", res );
cvSaveImage( "points.png", res );
cvWaitKey(0);
return 0;
}

```

В данной программе после загрузки пары изображений производится выделение на них ключевых точек с помощью процедуры cvGoodFeaturesToTrack. Далее для каждой точки первого изображения находится ближайшая точка второго изображения. Если эти две точки располагается достаточно близко, то они считаются отождествленными.

При этом предполагается, что в видеоряде отсутствуют быстрые перемещения.

На рис. 18 представлен пример результата выполнения программы для пары изображений. Как видно из рисунка, менее чем для половины точек найдены смещения, близкие к реальным.

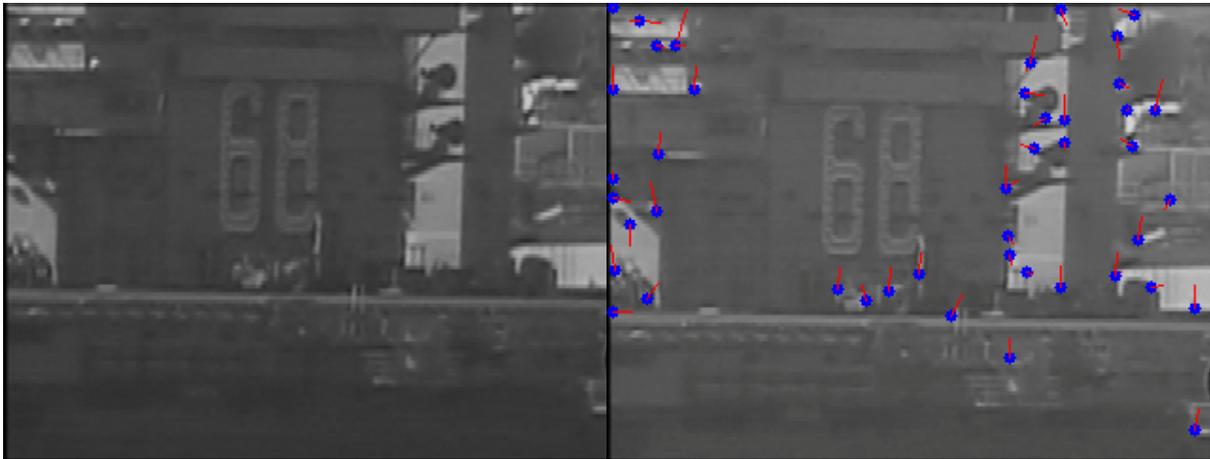


Рис. 18. Результат определения смещений для ключевых точек

При этом ряд ошибок связан с тем, что некоторые ключевые точки располагаются друг от друга на расстоянии, меньшем, чем длина их вектора перемещения, поэтому критерий близости их положения на изображении не позволяет произвести правильное отождествление. В ряде случаев, напротив, смещения находятся слишком большими, что говорит об отсутствии соответствующей сопряженной точки (процедура выделения точек не достаточно устойчива).

Можно сделать вывод, что для снятия неоднозначности при построении оптического потока по ключевым точкам необходимо производить дополнительное описание этих точек системой признаков, по которым осуществлять их отождествление. Вместо системы признаков можно также использовать и коэффициент корреляции областей, взятых вокруг сопоставляемых ключевых точек.

Литература

1. Хорн Б.К.П. Зрение роботов / Б.К.П. Хорн. – Пер. с англ. – М.: Мир, 1989. – 489 с.
2. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – Пер. с англ.– М.: Издательский дом "Вильямс", 2004 . – 926 с.

Вопросы для самопроверки:

1. В чём заключается задача восстановления оптического потока?
2. В чём заключается проблема неоднозначности решения задачи определения оптического потока? Какими способами она может решаться?
3. Охарактеризуйте отличительные особенности алгоритмов оценивания оптического потока на основе локальной корреляции и ключевых точек.
4. Какие признаки могут использоваться при отслеживании ключевых точек?
5. Чем задача восстановления оптического потока при движении наблюдателя отличается от задачи стереозрения?
6. Могут ли быть наложены какие-либо ограничения на взаимные перемещения точек изображения и при каких условиях?

ОГЛАВЛЕНИЕ

1	МОДЕЛИРОВАНИЕ БИОЛОГИЧЕСКИХ СИСТЕМ ЗРИТЕЛЬНОГО ВОСПРИЯТИЯ	3
2	ИССЛЕДОВАНИЕ ИНВАРИАНТНОСТИ МЕТОДОВ СОПОСТАВЛЕНИЯ ИЗОБРАЖЕНИЙ В УСЛОВИЯХ ИЗМЕНЧИВОСТИ	13
3	ПОСТРОЕНИЕ КАРТ СТЕРЕОДИСПАРАТНОСТИ	24
4	АНАЛИЗ СТРУКТУРЫ ОПТИЧЕСКОГО ПОТОКА	34
	ОГЛАВЛЕНИЕ.....	41
	КАФЕДРА КОМПЬЮТЕРНОЙ ФОТОНИКИ И ВИДЕОИНФОРМАТИКИ	<i>i</i>