

Распознавание образов Напишем программу, которая читает номерные знаки автомобилей

Содержание

[\[убрать\]](#)

1 Perl: Обнаружим объекты

- 1.1 Обнаружение, различение, опознавание
- 1.2 Значимость контуров
- 1.3 Согласованная фильтрация
- 1.4 Водяные знаки

Perl: Обнаружим объекты

Perl Марко Фиоретти

- [Часть 1: Основы языка](#)
- [Часть 2: Регулярные выражения](#)
- [Часть 3: Контроль за выполнением](#)
- [Часть 4: Функции, модули и прочие штучки](#)

Perl Михаил Смирнов

- [Perl: Создаем изображения](#)
- [Perl: Повысим качество фото](#)
- [Perl: Водяные знаки](#)
- **Perl: Обнаружим объекты**

Часть 4: Реализация объектов в *Perl* оставляет желать лучшего, но это не значит, что с его помощью нельзя находить объекты на фотографиях, считает [Михаил Смирнов](#).

Многие практические приложения обработки и анализа изображений включают процедуру обнаружения и распознавания объектов. В эту сферу попадают многие задачи современности: от медицинских исследований до навигации автоматических космических станций. В историческом плане наибольшую известность получили методы сравнения с эталоном, оценивающие степень корреляции объекта обнаружения и эталона. В последние десятилетия интенсивно развивались методы распознавания образов, основанные на формальном (структурном или семантическом) описании изображений. К успехам формальных описаний можно отнести приложения по распознаванию печатного текста, отпечатков пальцев, классификации хромосом в биомедицинских приложениях и так далее. В распознавании образов получили дальнейшее развитие и искусственные нейронные сети, основное преимущество которых заключается в наличии универсального механизма обучения. Вместе с тем, в ряде экстремальных задач обнаружения метод сравнения с эталоном остается доминирующим.



Рис. 1. Фотографическая карта местности: а)

полутонное входное изображение; б) контурное представление.

Ярким примером служат системы навигации и наведения крылатых ракет по рельефу местности. В качестве эталонов используются карты местности, изображения которых предварительно заносятся в память бортового компьютера. В ходе полета видеосистема ракеты сканирует местность, а компьютер оценивает корреляцию текущего изображения местности с набором эталонов и корректирует положение ракеты в пространстве относительно расчетной траектории полета. В систему наведения крылатой ракеты входит электронно-оптическая корреляционная подсистема, которая выдает результаты в реальном времени. Цифровые картины-эталонны предварительно отснятых районов местности последовательно вводятся в электронно-оптический коррелятор по мере перемещения ракеты вдоль траектории полета. Бортовой компьютер оценивает величины корреляции и прокладывает курс ракеты по рельефу местности. Нужно отметить, что в соответствии со свойствами зрительного восприятия для улучшения результатов сравнения применяется корреляция по контурам рельефа местности. На рис. 1 представлен пример фотографической карты местности и её контурное (градиентное) представление. Контурное представление может быть получено с помощью, например, дифференциального оператора Собеля или с помощью высокочастотной фильтрации видеоданных.

Разумеется, на данном уроке мы поставим перед собой более скромную и куда более мирную задачу, а именно: реализуем обнаружитель-коррелятор для простых сцен. Для нас будет важен принцип построения такого обнаружителя на практическом уровне с помощью программных средств языка *Perl*.

Обнаружение, различение, опознавание

Джонсон [J. Johnson], исследуя влияние видеосистем на восприятие военных объектов, установил три основных уровня видения, характеризующих процесс отождествления изображений данных объектов: обнаружение (появление объекта в поле зрения), различение (разбиение на классы: танки, автомобили) и опознавание типа объекта (например, наблюдается танк t-72). На рис. 2 иллюстрируется критерий Джонсона, где слева показаны периоды строчной развертки телевизионной системы наблюдения, а справа – контурное изображение автомобиля. Таким образом, Джонсон установил, что число полупериодов развертки, приходящихся на минимальный размер объекта, должно быть равно $2 (+1/-0.5)$ для обнаружения (рис. 2а), $8 (+1.6/-0.4)$ для классификации (рис. 2б) и $12.8 (+3.2/-2.8)$ для опознавания (рис. 2в).

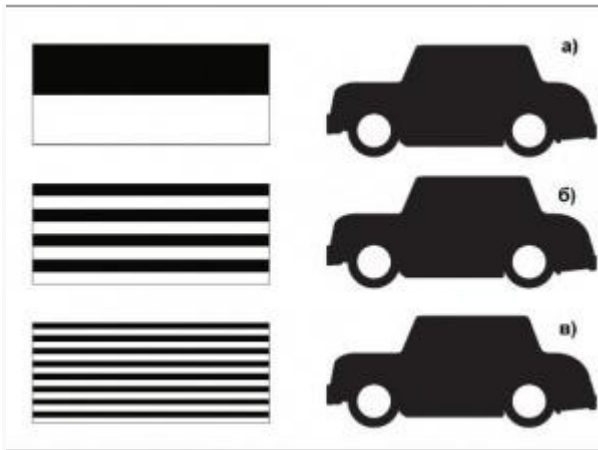


Рис. 2. Иллюстрации к критерию Джонсона: а)

уровень обнаружения объектов; б) уровень классификации объектов; в) уровень опознавания типа объекта.

Следовательно, если мы установим, что в изображении имеются танки и автомашины, то мы решим задачу обнаружения. Если нам удастся отделить изображения танков от изображений автомобилей, то мы решим задачу классификации. И, наконец, если мы сможем сказать, что танк на изображении является российским т-72, а не американским М1А2, мы решим задачу опознавания типа.

Значимость контуров

Из исследований зрительного анализатора известно, что основную информацию о зрительных образах несут контуры объектов и резкие перепады яркости на границах последних. Простейшей процедурой для выделения контуров объектов является вычисление разностей значений яркости между соседними элементами изображения, что представляет собой процедуру дифференцирования. Чем больше будут эти разности, тем сильнее проявятся контуры объектов в изображении. Наибольшим значениям разностей будут соответствовать максимумы градиента яркости. Эквивалентом пространственному дифференцированию в плоскости изображения будет применение градиентного фильтра в частотной области. Примером такого фильтра может служить формула:

$$H(u, v) = u^2 + v^2,$$

где u, v – пространственные частоты. Действие данного фильтра как раз и представлено на рис. 1а. Программную реализацию градиентного фильтра мы рассмотрим чуть ниже, совместно с выполнением процедуры сравнения с эталоном, которая получила название согласованной фильтрации.

Согласованная фильтрация

Согласованный фильтр был впервые использован для обнаружения сигналов в радиолокации. В одной из предыдущих статей цикла ([LXF135](#)) мы рассмотрели процедуру фильтрации для повышения качества фотоизображения. Для этого мы сформировали фильтр $H(u, v)$ и умножили его на спектр фотоизображения $S(u, v)$. На формальном уровне, с позиции теории линейной

фильтрации, как фильтр для повышения качества, так и фильтр для согласованной фильтрации равноценны. Основное отличие согласованного фильтра состоит в том, что его импульсная характеристика $h(x,y)$ должна быть согласована и сопряжена с функцией объекта $s(x,y)$:

$$h(x, y) = s(x_0 - x, y_0 - y) \quad (1)$$

Будем перемещать отклик $h(x,y)$ относительно объекта и вычислять корреляцию между двумя этими функциями. Очевидно, что максимум корреляции будет достигнут в момент их совмещения. Такая процедура получила название свертки двух функций. Функция согласованного фильтра рассчитывается как преобразование Фурье функции отклика (1):

$$H(u, v) = S^*(u, v) \exp\{-j(u \cdot x_0 + v \cdot y_0)\} \quad (2)$$

где $S(u,v)$ – спектр эталона (1), j – мнимая единица, а знак * обозначает комплексное сопряжение. Как мы видим, функция согласованного фильтра является комплексной, и поэтому в программной реализации мы должны предусмотреть перемножение комплексных чисел. Ниже представлен программный код, реализующий согласованную фильтрацию с помощью градиентного согласованного фильтра. Входное изображение предварительно заносится в двумерный массив `@re`. Размерность входного изображения равна $\mathbf{N} \times \mathbf{N}$ точек и определяется в программе параметром `$N`, а размерность эталона $s(x,y)$ задается параметрами `$sx` по ширине и `$sy` по высоте.

```
for($j=0;$j<$N; $j++){
for($i=0;$i<$N; $i++){
    $im[$i][$j] = 0;
    $re1[$i][$j] = 0;
    $im1[$i][$j] = 0;
}}
&d2cdft( \@re,\@im,\$N,\$PI );
for($j=0;$j<$sy; $j++){
for($i=0;$i<$sx; $i++){
    $re1[$i][$j] = $s[$i][$j];
}}
&d2cdft( \@re1,\@im1,\$N,\$PI );
$N2 = int($N/2);
for($j=0;$j<$N; $j++){
    $u = $j;
    if($j > $N2) { $u = $j - $N;}
    for($i=0;$i<$N; $i++){
        $v = $i;
        if($i > $N2) { $v = $i - $N;}
        $gr = ($u**2 + $v**2)/$N;
        $Ure = $gr*( $re[$i][$j]*$re1[$i][$j] + $im[$i][$j]*$im1[$i][$j] );
        $im = $gr*( $im[$i][$j]*$re1[$i][$j] - $re[$i][$j]*$im1[$i][$j] );
        $re[$i][$j] = $Ure;
        $im[$i][$j] = $im;
    }
}
```

```
}}
```

После преобразования Фурье входного изображения `@re` элементы массива `$re[$i][$j]` и `$im[$i][$j]` будут представлять собой вещественную и мнимую части спектра исходного изображения. Преобразование Фурье осуществляется с помощью подпрограммы `d2cdft()` быстрого преобразования Фурье, описание которой было дано в предыдущих статьях цикла ([LXF135](#)).

Комплексные элементы массивов `$re1[$i][$j]` и `$im1[$i][$j]` согласованного фильтра также получены с помощью подпрограммы `d2cdft()`, которая применяется к функции эталона. Перемножение элементов массива спектра исходного изображения и элементов согласованного фильтра осуществляется по правилу умножения двух комплексных чисел. Результат перемножения комплексных чисел умножается на значения функции градиентного фильтра `$gr`, вычисляемой в каждой точке спектра как суммы квадратов пространственных частот (трансформант) `$u` и `$v`.

Чтобы получить результирующие корреляционные отклики, необходимо выполнить обратное преобразование Фурье с помощью подпрограммы `d2cdfti()` в соответствии с программным кодом:



Рис. 3. Обнаружение символов цифры 3: а)

исходное изображение; б) результат согласованной фильтрации с двоичным эталоном цифры 3; в) сечение корреляционного поля.

```
&d2cdfti( \@re,\@im,\$N,\$PI );  
$max = 0;  
for($j=0;$j<$N; $j++){  
  for($i=0;$i<$N; $i++){  
    if( $re[$i][$j] > $max){ $max = abs($re[$i][$j]);}  
  }  
}
```

```

for($j=0;$j<$N; $j++){
for($i=0;$i<$N; $i++){
    $out[$i][$j] = abs(255*($re[$i][$j]/$max));
}}

```

Двумерные массивы **@re** и **@im**, на входе подпрограммы **d2cdfti()**, являются результатом умножения спектра входного изображения на согласованный фильтр, как это было показано в предыдущем фрагменте кода. На выходе подпрограммы **d2cdfti()** двумерный массив **@re** будет представлять собой результат корреляции эталона с входным изображением. Выходное корреляционное поле записывается в массив **@out**, элементы которого **\$out[\$i][\$j]** нормируются к максимальному значению **\$max** и представляются в виде чисел от 0 до 255. В качестве примера рассмотрим обнаружение символов на номерных знаках автомобилей. На рис. 3 представлено исходное изображение и результаты обнаружения цифры 3 на номерном знаке автомашины с помощью согласованной градиентной фильтрации, программную реализацию которой мы рассмотрели. Эталон для обнаружения служит двоичное изображение цифры 3.

Размерность исходного изображения на рис. 3а составляет **1024 × 1024** отсчетов. Размерность эталона равна **sx=19** отсчетам по ширине и **sy=40** отсчетам по высоте. Отметим, что пики максимумов корреляции будут смещены относительно объектов на половину размеров эталона, то есть на величину **sx/2** по ширине и **sy/2** по высоте.

Сделаем акцент на одном важном моменте согласованной фильтрации. Фильтр (2) является пространственно-инвариантным, или, иначе говоря, его действие справедливо для всего входного изображения. Это вытекает из формулы (1) для свертки, в которой операция свертки входного изображения $s(x,y)$ с откликом $h(x,y)$ осуществляется по всей плоскости при перемещении эталона вдоль осей x_0 - x и y_0 - y . По этой причине согласованный фильтр обеспечит обнаружение сразу нескольких (множества) объектов, расположенных произвольным образом во входном изображении.

Водяные знаки

Проблема соблюдения авторских прав при размещении фотографий в сети Интернет или задача поиска изображений с помощью поисковых систем так или иначе требуют сопоставления множества фотоизображений с эталоном или описанием изображения. В этом случае, согласованная фильтрация может рассматриваться как один из инструментов решения подобных задач.

Наиболее популярные поисковые системы в настоящее время предоставляют возможность поиска изображений на основе их текстового описания, содержащегося в тэгах кода HTML-документа. Соответственно, степень релевантности поисковым запросам может не соответствовать изображениям, предоставляемым в выдаче результатов поиска. Пользователям приходится просматривать множество страниц, чтобы найти требуемое изображение.

Более продвинутые (но менее известные) ресурсы предлагают сервисы, использующие методы распознавания образов, направленные на классификацию объектов в изображении. Однако поиск цифровых фотографий по объектам, присутствующим в снимках, пока далёк от удовлетво-

рительного решения, если в качестве оптимума взять наш зрительный анализатор. И виной этому не столько алгоритмы распознавания, сколько сложность самой проблемы машинного распознавания образов.

Одним из компромиссов в решении задачи поиска фотографий в сети Интернет может послужить способ классификации изображений с использованием видимых водяных знаков.

Нанесение их на изображение уже сейчас применяется во многих ресурсах мировой сети. Как правило, видимой меткой служит адрес сайта в сети Интернет, логотип ресурса или некоторая отвлеченная метка типа знака ©. Примером нанесения прозрачных меток могут служить карты поисковой системы Google, в отображаемое поле которых встраиваются водяные знаки с прозрачной основой. Попробуем и мы реализовать подобный подход. При этом сначала встроим видимые водяные знаки в произвольное изображение, а затем попытаемся обнаружить их с помощью программы, приведенной выше. Программный код для встраивания символа © в фотоизображение будет таким:



Рис. 4. Фотоизображение со встроенными видимыми знаками ©.

мыми знаками ©.

```
$op = .075;
$dh0 = 64;
$dw0 = 64;
$dh = 128;
$dw = 128;
$L = $N/128;
for($h=0;$h<$L; $h++){
    $m = $h*$dh + $dh0;
    for($w=0;$w<$L; $w++){
        $n = $w*$dw + $dw0;
        for($j=0;$j<$sy;$j++){
            for($i=0;$i<$sx;$i++){
                $sum = $re[$i+$n][$j+$m] + $op*$rel[$i][$j];
                if($sum > 255){ $sum = 255;}
                $re[$i+$n][$j+$m]=$sum;
            }
        }
    }
}
```

```
}}  
}}
```

Входное и выходное изображения помещаются в двумерный массив **@re**. Параметр **\$op** определяет непрозрачность водяного знака относительно исходного фотоизображения и равен в данном случае **7,5 %**. Параметры **\$dh0** и **\$dw0** дают начальные смещения (отступы) по высоте и ширине фотоизображения, соответственно. Значения параметров **\$dh** и **\$dw** задают шаг встраивания водяных знаков по высоте и ширине снимка. Параметр **\$L** определяет число встраиваемых меток как по высоте, так и по ширине фотоизображения. С помощью параметров **\$sy** и **\$sx** задаются размеры встраиваемого знака ©, а так же размеры эталона © на этапе обнаружения. Значение **255** служит ограничителем на случай возможного превышения значения яркости. На рис. 4 представлен результат встраивания прозрачного знака ©. Водяные знаки равномерно размещены по полю изображения, образуя матрицу, имеющую восемь рядов и восемь столбцов.

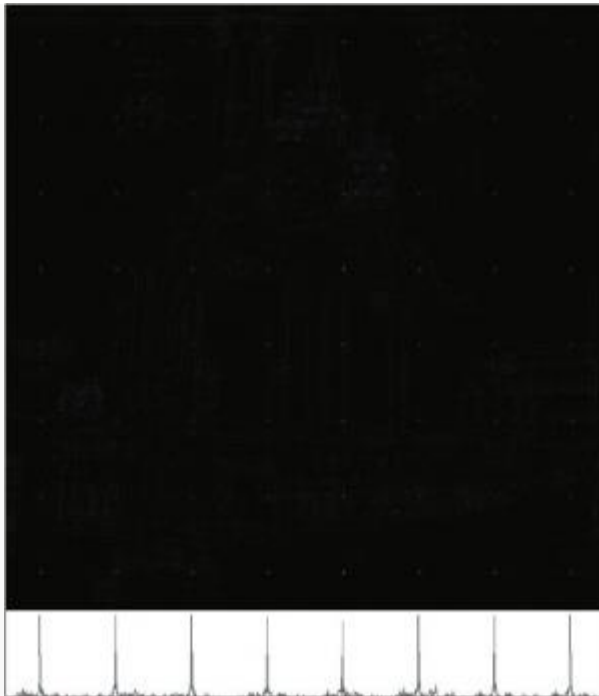


Рис. 5. Результаты обнаружения знаков © с помощью согласованной фильтрации: а) поле корреляции; б) одно из сечений поля корреляции.

На рис. 5 представлены результаты обнаружения искомых символов ©. В качестве эталона использовалось двоичное изображение знака ©.

Анализ полученных результатов позволяет нам сделать вывод, что при прозрачности водяных знаков, равной **92,5 %**, ошибки обнаружения (ошибки I рода) отсутствуют, а также отсутствуют ошибки ложного обнаружения, то есть II рода. Если мы будем увеличивать прозрачность водяных знаков, тем самым уменьшая их видимость, то неминуемо появятся ошибки ложного обнаружения, а затем и ошибки I рода. Уровень возникновения ошибок зависит от конкретного входного изображения. Для фотоизображения на рис. 4 ошибки II рода начинают появляться при величине прозрачности водяных знаков, равной **95 %**, а ошибки I рода – при **97 %**.

В нашей статье мы продемонстрировали возможности *Perl* по решению прикладных задач обнаружения объектов. Отметим, что цифровые методы в целом не обеспечивают высокой скорости расчета корреляции двумерных массивов из-за большого объема вычислений. И чем больше размер входных изображений, тем ниже эта скорость. Именно поэтому в системах навигации крылатых ракет применяются оптические корреляторы, упомянутые в начале. Но гибкость и оперативность программирования на *Perl*, в сочетании с мощью математических модулей *Perl*, позволяют создавать высокоэффективные алгоритмы обработки видеоданных.