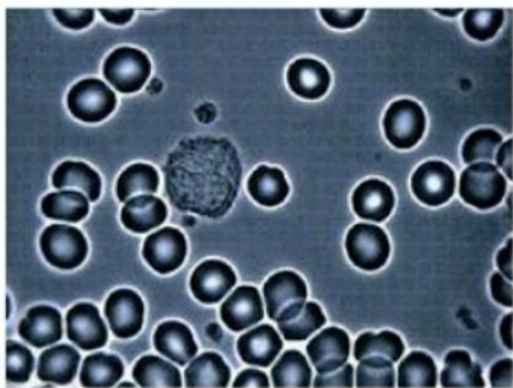


Выделение связанных компонент изображения

Паздникова М.В.

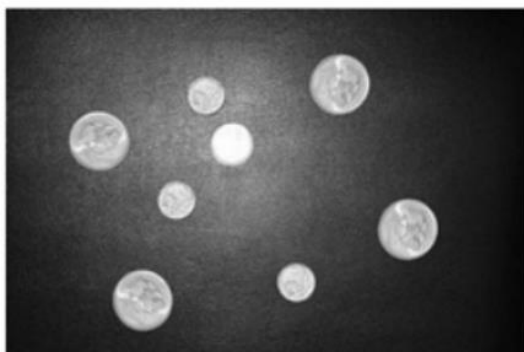
Примеры задач



Клетки крови



Ложки и сахар

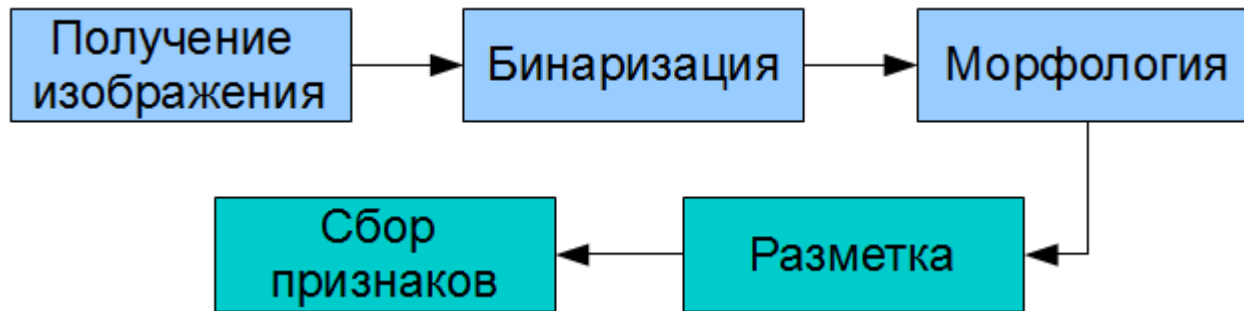


Монеты и купюры



Номера

Распознавание образов

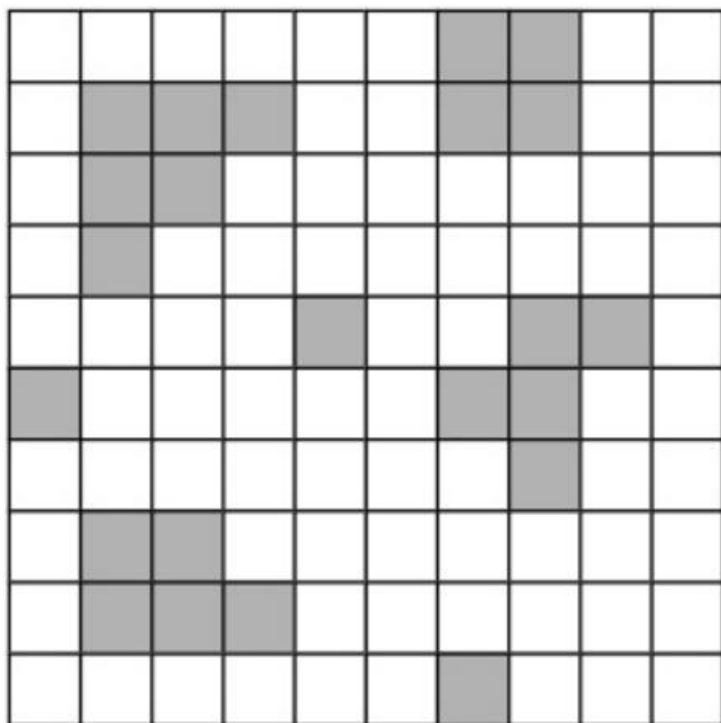


Получили бинарное изображение

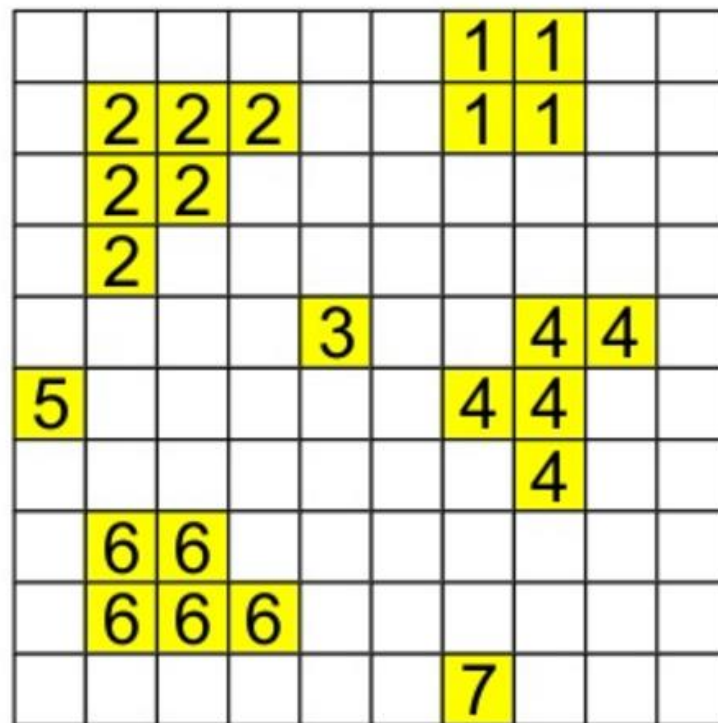


Нужна карта разметки

Разметка связанных областей

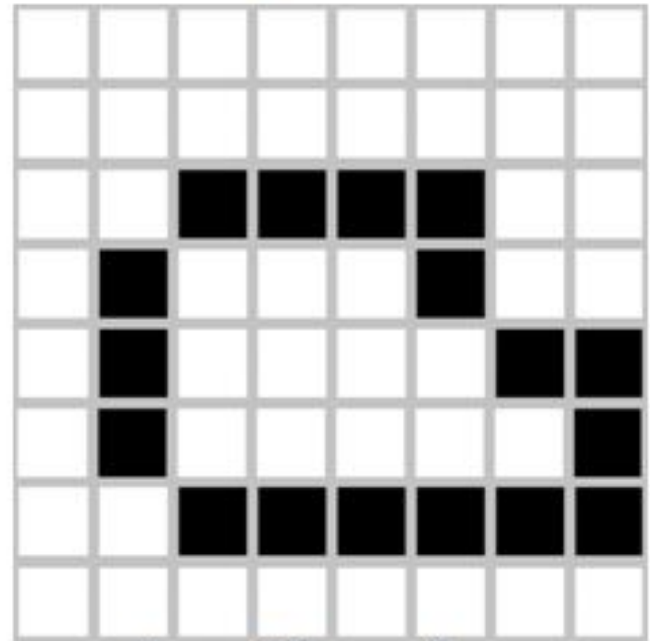
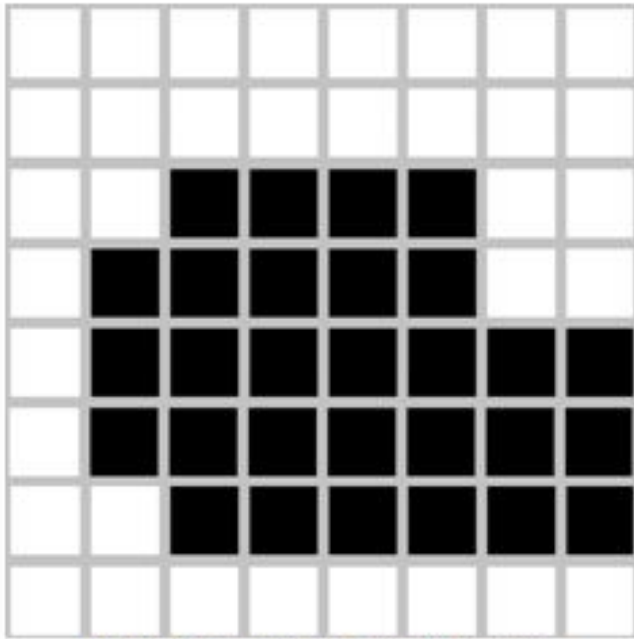


Бинарное изображение



Размеченное изображение

СВЯЗНОСТЬ



Виды связности

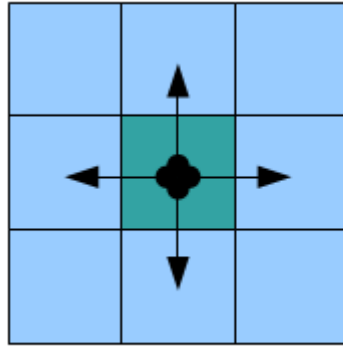


Рис. 1-а)

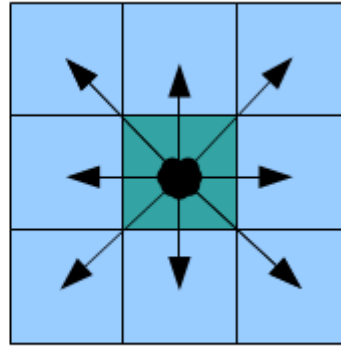


Рис. 1-б)

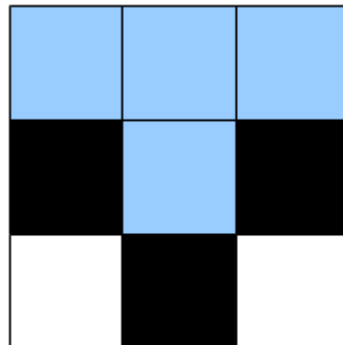


Рис. 2-а)

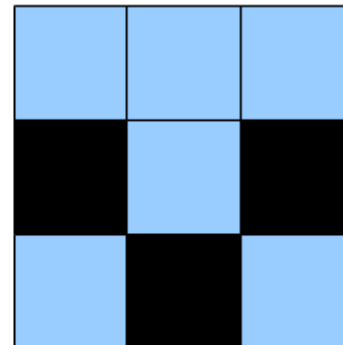


Рис. 2-б)

Рекурсивный алгоритм

- просмотр с точки (1,1);
- поочередно все точки строки: если все белые, переходим к следующей строке; иначе:
 - а) помечается точка;
 - б) определяется цвет четырех соседних точек;
 - в) если цвет всех точек белый или все помечены, закончить, иначе начальная позиция перемещается в первую найденную соседнюю черную точку;
 - г) повторяются пункты а, б и в;
- повторить до последней точки последней строки изображения.

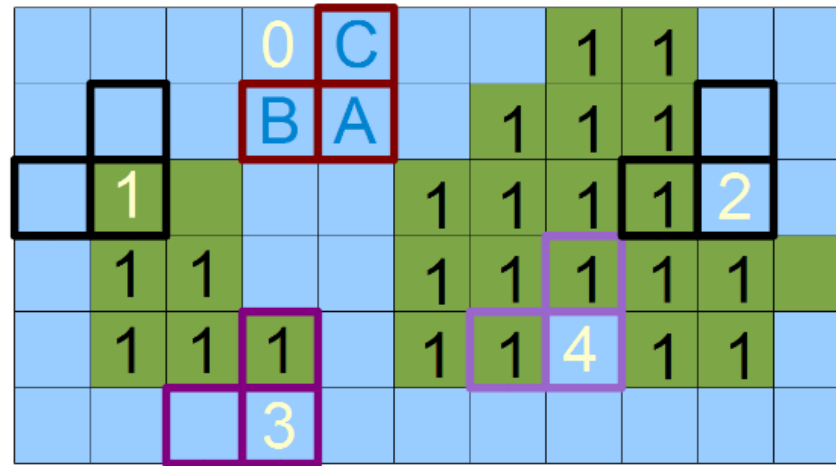
```
void Labeling(BIT* img[], int* labels[])
{
    int L = 1;
    for(int y = 0; y < H; ++y)
        for(int x = 0; x < W; ++x)
            Fill(img, labels, x, y, L++);
}

void Fill(BIT* img[], int* labels[], int x, int y, int L)
{
    if( (labels[x][y] == 0) & (img[x][y] == 1) )
    {
        labels[x][y] = L;
        if( x > 0 )
            Fill(img, labels, x - 1, y, L);
        if( x < W - 1 )
            Fill(img, labels, x + 1, y, L);
        if( y > 0 )
            Fill(img, labels, x, y - 1, L);
        if( y < H - 1 )
            Fill(img, labels, x, y + 1, L);
    }
}
```

Итеративный алгоритм



Итеративный алгоритм



- 0: пропускаем пиксель.
- 1: создание нового объекта — новый номер.
- 2: помечаем текущий пиксель A меткой, расположенной в B.
- 3: помечаем текущий пиксель A меткой, расположенной в C.
- 4: пиксель A может быть помечен либо как B либо как C.

Сведение к бинарному изображению



Алгоритмы, основанные на границах областей



Общая схема работы данной группы методов:

- Обнаружение границ регионов.
- Обнаружение регионов исходя из найденных границ.

Алгоритмы, основанные на границах областей

Достоинства:

- более гибок, чем простая бинаризация
- более стабилен к изменениям характеристик изображения

Недостатки:

- Слабая устойчивость (достаточно небольшого разрыва границы для несрабатывания алгоритма)
- Выделение областей не является точным (часть региона "отъедается" границей)
- Скорость работы методов невысока

Алгоритмы, основанные на поиске регионов

- Объявляем левый верхний пиксель изображения новым классом.
- Для пикселей первой строки считаем отклонение от класса левого пикселя и, сравнивая с порогом, либо добавляем пиксель к классу соседа, либо заводим новый класс.
- Первый пиксель каждой последующей строки сравниваем с классом верхнего пикселя. Далее будем сравнивать пиксель с классами двух соседей: левого и верхнего.
 - Если отклонение от обоих классов больше порога, то заводим новый класс, если отклонение больше только для одного класса, то добавляем пиксель к тому классу отклонение от которого меньше порога.
 - Если отклонение допустимо для обоих классов, то возможно 2 варианта:
 - $L(g(C_1) - g(C_2)) < \delta$ - тогда объединяем эти 2 класса (если они не один и тот же класс) и добавляем к объединённому классу текущий пиксель.
 - $L(g(C_1) - g(C_2)) > \delta$ - в этом случае добавляем пиксель к тому классу, от которого отклонение минимально.

Алгоритмы, основанные на поиске регионов



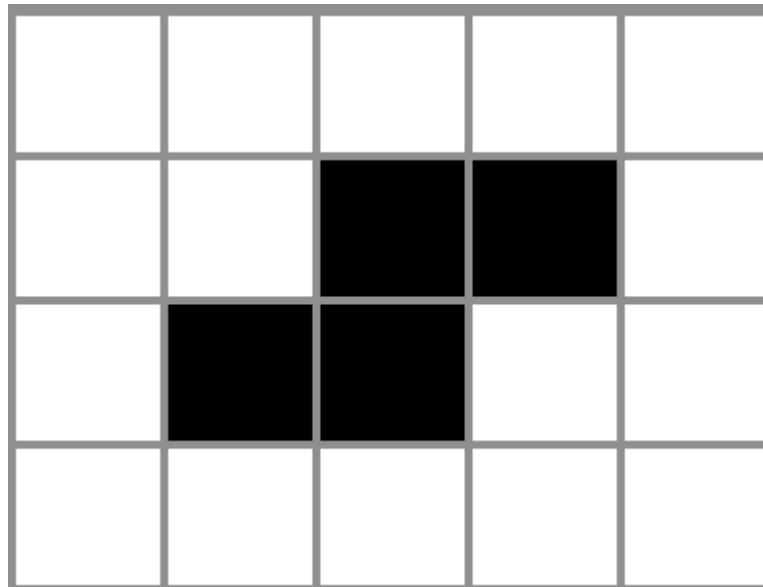
Алгоритмы, основанные на поиске регионов

Достоинства:

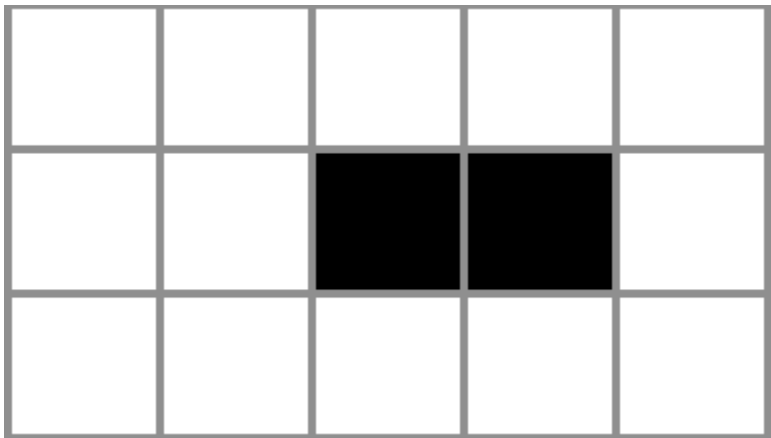
- Широкий спектр применения
- Гибкость (можно эффективно менять чувствительность алгоритма)
- Скорость (быстрее алгоритма с поиском границы метода)
- Устойчивость (более устойчив к ошибкам)
- Точность (найденные регионы не «обкусаны»)

Практическое задание

- Рекурсивным алгоритмом выделить связную область данного бинарного изображения (граничные пиксели можно не рассматривать):



Практическое задание (пример)



- (2;2) – белый;
- (2;3) – черный, не помечен:
 - Помечаем;
 - Рассмотрим соседей:
 - (1;3) – белый;
 - (2;2) – белый;
 - (2;4) – черный, не помечен:
 - Помечаем;
 - Рассмотрим соседей:
 - (1;4) – белый;
 - (2;3) – черный, помечен;
 - (2;5) – белый;
 - (3;4) – белый;
 - (3;3) – белый;
- (2;4) – черный, помечен;
- Конец.