

электронный журнал

МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК

Издатель: ФГБОУ ВПО «Московский государственный технический университет им. Н.Э. Баумана»

УДК 519.685

Оценка сложности алгоритма автоматического расположения видеокамер системы наблюдения

Студент

кафедры «Информационные системы и телекоммуникации»: Д.А. Локтев

Научный руководитель: А.Н. Алфимцев,

к. т. н., доцент кафедры «Информационные системы и телекоммуникации»

МГТУ им. Н.Э. Баумана

loktevdan@yandex.ru

В настоящее время постиндустриального развития цивилизации все большее значение приобретают системы и средства автоматизации и управления процессами. В состав таких систем входит программное, техническое и информационное обеспечение, в каждой подсистеме важную роль играет последовательность выполнения тех или иных действий по сбору, регистрации, передачи и обработки информации. Каждая такая последовательность задается алгоритмом, который имеет входные параметры, основные неизвестные, условия ветвления направлений обработки информации и конечные характеристики обработки данных. Таким образом, важнейшей особенностью большинства систем автоматизации и управления на первом уровне логического обеспечения являются алгоритмы и формализация знаний и процедур. Алгоритмизация как составная часть системы автоматизации процессов и управления системами является составной частью многих исследований и разработок, связанных с применением ЭВМ в различных областях хозяйственной деятельности человека, например, при создании современных мультимедийных интерфейсов человеко-машинного взаимодействия [1,2], при реализации прикладных программ по расчету в области механики деформируемых тел [3] и при проектировании систем информационной безопасности и защиты информации [4]. Изучению свойств и качества работы алгоритмов посвящена большая обзорная статья [5], в которой описаны результаты использования методов получения нижних оценок, а в качестве объектов алгоритмизации рассматриваются конечные и бесконечные множества. В работе [6] теоретически и эмпирически оцениваются комбинированные алгоритмы, моделирующие системы ограничений, и сравниваются с традиционными методами оценок применительно к задачам программирования в ограничениях. Но все же чаще в работах отечественных и зарубежных исследователей алгоритм используется в качестве средства решения конкретной задачи и не рассматривается в качестве самостоятельного объекта исследования, вследствие, этого некоторые реализованные решения могут быть далеки от оптимальных. Любой алгоритм обладает набором определенных свойств, из которых принято выделять результативность, определенность, дискретность и массовость. В данной же работе основным является сам алгоритм, а не задача, решаемая с его помощью, предложенный алгоритм исследуется с точки зрения его сложности, для определения которой рассчитывается скорость реализации алгоритма, объем оперативной памяти и

свободное место на ПЗУ, требующиеся при этом, таким образом, исследователи свойств алгоритмов оказываются перед дилеммой объемно-временной сложности.

В общем случае сложность алгоритма зависит от количества (объема) входных данных, количества точек ветвления (условий ветвления алгоритма), количества внутренних процедур, используемых функций, подключаемых библиотек, уровня вложенности вычислений, количества итераций внутренних и внешних циклов, формы предоставления конечных результатов и т.д. В данной работе реализована идея, согласно которой, порядок сложности алгоритма зависит только от наиболее быстро возрастающей его ветви, при этом постоянные множители в получаемых выражениях предлагается не использовать.

Таким образом, далее будем учитывать максимально возможное количество итераций, то есть при условном операторе будем учитывать, что выполняется условие с максимальным количеством действий.

Представим исследуемый алгоритм в виде следующих элементов (блоков):

- 1) Ввод трех координат для N (то же самое, что и i_{\max}) точек – $O(3N)$.
- 2) Ввод фокусного расстояния, угла обзора, количества камер для установки – $O(k_{\text{okam}}+2)$, что приблизительно равно $O(k)$.
- 3) Поиск высоты и ширины плана (для коэффициента пропорциональности, чтобы рисунок всегда помещался на экран) – приблизительно $O(2N)$, так как два последовательных цикла:

```
//ищем visotu plana
float ymin=y[0];
float ymax=y[0];
for (i=1; i<=imax; i++)
{
    if (y[i]>ymax) {ymax=y[i];}
    if (y[i]<ymin) {ymin=y[i];}
}
float maxvisota=ymax-ymin;
```

```
//ищем birinu plana
float xmin=x[0];
float xmax=x[0];
for (i=1; i<=imax; i++)
{
    if (x[i]>xmax) {xmax=x[i];}
    if (x[i]<xmin) {xmin=x[i];}
}
float maxbirina= xmax-xmin;
```

- 4) Рисование плана – один цикл, то есть сложность данного «куска» программы (с учетом математических операций, включенных в данный цикл) – $O(8N+8)$, что приблизительно равно $O(8N)$.

```
//risuem plan
for (i=1; i<=imax-1;i++)
{
    Form1->Image1->Canvas->MoveTo(x[i-1]*kpr,CH-y[i-1]*kpr);
    Form1->Image1->Canvas->LineTo(x[i]*kpr,CH-y[i]*kpr);
};
Form1->Image1->Canvas->MoveTo(x[imax-1]*kpr,CH-y[imax-1]*kpr);
```

Form1->Image1->Canvas->LineTo(x[0]*kpr,CH-y[0]*kpr);

5) Поиск ближайшей точки до двери – здесь количество дверей равно $p/2$, сл-но, сложность алгоритма $=O(7N)+O(p/2*(2N+2))$.

```
//ibem blizai buyu to4ku (ne drugoi dveri) do dveri
int p=0;
for (i=0; i<=imax-1; i++)
{
if ((z[i]==z[i+1]) && (z[i]==1)) //eto dver
{
xdv[p]=x[i];
xdv[p+1]=x[i+1];
ydv[p]=y[i];
ydv[p+1]=y[i+1];
to4ki[p]=i;
to4ki[p+1]=i+1;
p=p+2;
}
}
for (h=0; h<=p-2; h=h+2)
{
if (ydv[h]==ydv[h+1])
{
for (i=0; i<=imax-1; i++)
{
if (z[i]!=1)
{
if ((abs(ydv[h]-y[i])<minrasst)&&((ydv[h]-y[i])!=0))
{
minrasst=abs(ydv[h]-y[i]);
}
}
}
if (minrasst<rp)
{
z[to4ki[h]]==0;
}
}
if (xdv[h]==xdv[h+1])
{
for (i=0; i<=imax-1; i++)
{
if (z[i]!=1)
{
if ((abs(xdv[h]-x[i])<minrasst)&&((xdv[h]-x[i])!=0))
{
minrasst=abs(xdv[h]-x[i]);
}
}
}
if (minrasst<rp)
{
```

```

z[to4ki[h]]==0;
}
}
}

```

6) Расчет количества дверей – $O(N)$

```

//kol-vo dveri
for (i=0; i<=imax-1; i++)
{
if ((z[i]==z[i+1]) && (z[i]==1)) //eto dver
{
koldv=koldv+1;
}
}

```

7) Расчет макс количества камер – сложность равняется $O(3)$

```

int maxkolkamer=koldv*360/beta; //max kol-vo kamer

```

8) Показ дверей на плане здания в виде двух отрезков (с учетом математических операций) – $O(20N)$

```

//pokaz dveri
for (i=0; i<=imax-1; i++)
{
if ((z[i]==z[i+1]) && (z[i]==1)) //eto dver
{
if (x[i]==x[i+1])
{
Form1->Image1->Canvas->MoveTo((x[i]-5)*kpr,CH-y[i]*kpr);
Form1->Image1->Canvas->LineTo((x[i]+5)*kpr,CH-y[i]*kpr);
Form1->Image1->Canvas->MoveTo((x[i+1]-5)*kpr,CH-y[i+1]*kpr);
Form1->Image1->Canvas->LineTo((x[i+1]+5)*kpr,CH-y[i+1]*kpr);
}
if (y[i]==y[i+1])
{
Form1->Image1->Canvas->MoveTo(x[i]*kpr,CH-(y[i]-5)*kpr);
Form1->Image1->Canvas->LineTo(x[i]*kpr,CH-(y[i]+5)*kpr);
Form1->Image1->Canvas->MoveTo(x[i+1]*kpr,CH-(y[i+1]-5)*kpr);
Form1->Image1->Canvas->LineTo(x[i+1]*kpr,CH-(y[i+1]+5)*kpr);
}
}
}
}

```

9) Нахождение координат камер и их прорисовка на плане в виде треугольников (с учетом математических операций) – $O(50N)$

```

for (i=0; i<=imax-1; i++)
{
if ((z[i]==z[i+1]) && (z[i]==1) && (q<=k-1)) //eto dver
{
if (x[i]==x[i+1])

```

```

    {
    ykam[q]=y[i]+fabs(y[i]-y[i+1])/2;
    xkam[q]=x[i]+rp;
    xnizkam[q]=xkam[q]-rp;
    xverkam[q]=xkam[q]-rp;
    ynizkam[q]=ykam[q]+sin(beta*M_PI/180/2)*r;
    yverkam[q]=ykam[q]-sin(beta*M_PI/180/2)*r;
    //treugolnik
    Form1->Image1->Canvas->MoveTo(xkam[q]*kpr,CH-ykam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xnizkam[q]*kpr,CH-ynizkam[q]*kpr);
    Form1->Image1->Canvas->MoveTo(xnizkam[q]*kpr,CH-ynizkam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xverkam[q]*kpr,CH-yverkam[q]*kpr);
    Form1->Image1->Canvas->MoveTo(xkam[q]*kpr,CH-ykam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xverkam[q]*kpr,CH-yverkam[q]*kpr);
        q=q+1;
    }
    if (y[i]==y[i+1])
    {
    xkam[q]=x[i]+fabs((x[i+1]-x[i])/2);
    ykam[q]=y[i]+rp;
    xnizkam[q]=xkam[q]-sin(beta*M_PI/180/2)*r;
    xverkam[q]=xkam[q]+sin(beta*M_PI/180/2)*r;
    ynizkam[q]=ykam[q]-rp;
    yverkam[q]=ykam[q]-rp;
    //treugolnik
    Form1->Image1->Canvas->MoveTo(xkam[q]*kpr,CH-ykam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xnizkam[q]*kpr,CH-ynizkam[q]*kpr);
    Form1->Image1->Canvas->MoveTo(xnizkam[q]*kpr,CH-ynizkam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xverkam[q]*kpr,CH-yverkam[q]*kpr);
    Form1->Image1->Canvas->MoveTo(xkam[q]*kpr,CH-ykam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xverkam[q]*kpr,CH-yverkam[q]*kpr);
        q=q+1;
    }
}
}
}

```

10) Дальнейшее нахождение координат камер и их прорисовка на плане в виде треугольников (с учетом математических операций) – $O(k*50N)$ (k – кол-во камер)

```

//если kol-vo kamer>kol-va dverei
while (q<=k-1)
{
    for (i=0; i<=imax-1; i++)
    {
    if ((z[i]==z[i+1]) && (z[i]==1)&& (q<=k-1)) //eto dver
    {
    if (x[i]==x[i+1])
    {
    ykam[q]=y[i]+fabs((y[i]-y[i+1])/2);
    xkam[q]=x[i]+rp;
    xnizkam[q]=xverkam[q-koldv];
    ynizkam[q]=yverkam[q-koldv];

```

```

    xverkam[q] = ((xverkam[q-koldv] - xkam[q]) * cos(beta*M_PI/180) - (yverkam[q-
koldv] - ykam[q]) * sin(beta*M_PI/180)) + xkam[q];
    yverkam[q] = ((xverkam[q-koldv] - xkam[q]) * sin(beta*M_PI/180) + (yverkam[q-
koldv] - ykam[q]) * cos(beta*M_PI/180)) + ykam[q];
    //треугольник
    Form1->Image1->Canvas->MoveTo(xkam[q]*kpr,CH-ykam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xnizkam[q]*kpr,CH-ynizkam[q]*kpr);
    Form1->Image1->Canvas->MoveTo(xnizkam[q]*kpr,CH-ynizkam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xverkam[q]*kpr,CH-yverkam[q]*kpr);
    Form1->Image1->Canvas->MoveTo(xkam[q]*kpr,CH-ykam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xverkam[q]*kpr,CH-yverkam[q]*kpr);
    q=q+1;
}
if (y[i]==y[i+1])
{
    xkam[q]=x[i]+fabs((x[i+1]-x[i])/2);
    ykam[q]=y[i]+rp;
    xnizkam[q]=xverkam[q-koldv];
    ynizkam[q]=yverkam[q-koldv];
    xverkam[q] = ((xverkam[q-koldv] - xkam[q]) * cos(beta*M_PI/180) - (yverkam[q-
koldv] - ykam[q]) * sin(beta*M_PI/180)) + xkam[q];
    yverkam[q] = ((xverkam[q-koldv] - xkam[q]) * sin(beta*M_PI/180) + (yverkam[q-
koldv] - ykam[q]) * cos(beta*M_PI/180)) + ykam[q];
    //треугольник
    Form1->Image1->Canvas->MoveTo(xkam[q]*kpr,CH-ykam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xnizkam[q]*kpr,CH-ynizkam[q]*kpr);
    Form1->Image1->Canvas->MoveTo(xnizkam[q]*kpr,CH-ynizkam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xverkam[q]*kpr,CH-yverkam[q]*kpr);
    Form1->Image1->Canvas->MoveTo(xkam[q]*kpr,CH-ykam[q]*kpr);
    Form1->Image1->Canvas->LineTo(xverkam[q]*kpr,CH-yverkam[q]*kpr);
    q=q+1;
}
}
}

```

Таким образом, суммарное значение сложности алгоритма, рассчитанное по описанной методике и представляющее собой верхнюю оценку вычислительной сложности равно $O(88N) + O(p/2*(2N+2)) + O(k*50N)$.

$p/2$ - количество дверей в здании, то есть p - количество координат, которыми обозначаются двери. Максимально теоретическое p равно общему количеству координат N , а максимальное количество камер k зависит от угла обзора β и количества дверей $p/2$ и равно: $p/2*360/\beta$. Минимальный угол обзора по горизонтали для видеокамер с размером матрицы 1/4" равен 2,444 градусов. Таким образом, можно записать верхнюю оценку вычислительной сложности как:

$$O(88N) + O(p/2*(2N+2)) + O(k*50N) = O(88N) + O(p/2*(2N+2)) + O(p/2*360/\beta*50N) = O(88N) + O(N^2+N) + O(N/2*360/2,444*50N) = O(88N) + O(N^2+N) + O(3682,5N^2) \approx O(3683N^2).$$

Из приведенных оценок отдельных элементов алгоритма видно, какие из них вносят наибольший вклад в конечное значение сложности, согласно этим данным можно выделить процедуры, которые могут быть оптимизированы на этапе тестирования программного приложения для зданий и помещений сложной геометрической формы.

Литература

1. Алфимцев А.Н., Девятков В.В. Интеллектуальные мультимодальные интерфейсы. – Калуга, ООО «Полиграф-Информ», 2011. – 328 с.
2. Девятков В.В., Алфимцев А.Н. Распознавание манипулятивных жестов // Вестник МГТУ им. Н.Э. Баумана, Сер. Приборостроение, № 3, 2007. – С. 56 – 75.
3. Локтев А.А., Залетдинов А.В. Определение точек взаимодействия прямых и отраженных волн в пластинке. // Вестник МГСУ, № 4, 2010. – С. 303 – 308.
4. Айдаров Ю.Р. Новый алгоритм анализа протоколов информационной безопасности и оценка его вычислительной сложности. // Вестник Пермского университета. Серия: Математика. Механика. Информатика, № 4, 2008. – С. 165 – 168.
5. Кудрявцев В.Б. Андреев А.Е. О сложности алгоритмов. // Фундаментальная и прикладная математика, Т.15. № 3. 2010. – С. 135 – 181.
6. Семенов В.А., Сидяка О.В. Теоретические и экспериментальные оценки сложности методов локального распространения в задачах программирования в ограничениях // Труды Института системного программирования РАН, т. 19., 2010. – С. 117 – 133.