

图像压缩

黑白图像存储

像素点灰度值：0~255，为8位二进制数

图像的灰度值序列： $\{p_1, p_2, \dots, p_n\}$ ， p_i 为第 i 个像素点灰度值

图像存储：每个像素的灰度值占8位，
总计空间为 $8n$



有没有更好的存储
方法



图像变位压缩的概念

变位压缩存储：将 $\{p_1, p_2, \dots, p_n\}$ 分成 m 段

$$S_1, S_2, \dots, S_m$$



同一段的像素占用位数相同

第 t 段有 $l[t]$ 个像素，每个占用 $b[t]$ 位

段头：记录 $l[t]$ (8位) 和 $b[t]$ (3位) 需要 11 位

总位数为

$$b[1] \cdot l[1] + b[2] \cdot l[2] + \dots + b[m] \cdot l[m] + 11m$$

图像压缩问题

约束条件：第 t 段像素个数 $l[t] \leq 256$

第 t 段占用空间： $b[t] \times l[t] + 11$

$$b[t] = \left\lceil \log(\max_{p_k \in S_t} p_k + 1) \right\rceil \leq 8$$

问题：给定像素序列 $\{p_1, p_2, \dots, p_n\}$ ，
确定最优分段，即

$$\min_T \left\{ \sum_{t=1}^m (b[t] \times l[t] + 11) \right\},$$

$T = \{S_1, S_2, \dots, S_m\}$ 为分段

实例

灰度值序列

$P=\{10,12,15,255,1,2,1,1,2,2,1,1\}$

分法1: $S_1=\{10, 12, 15\}$, $S_2=\{255\}$,
 $S_3=\{1, 2, 1, 1, 2, 2, 1, 1\}$

分法2: $S_1=\{10,12,15,255,1,2,1,1,2,2,1,1\}$

分法3: 分成12组, 每组一个数

存储空间

分法1: $11 \times 3 + 4 \times 3 + 8 \times 1 + 2 \times 8 = 69$

分法2: $11 \times 1 + 8 \times 12 = 107$

分法3: $11 \times 12 + 4 \times 3 + 8 \times 1 + 1 \times 5 + 2 \times 3 = 163$

子问题界定与计算顺序

子问题前边界为1，后边界为 i

对应像素序列为 $\langle p_1, p_2, \dots, p_i \rangle$

优化函数值 $S[i]$ 为最优分段存储位数

计算顺序

$i = 1$ 

$i = 2$ 

...

$i = n$ 

算法设计

递推方程：设 $S[i]$ 是 $\{p_1, p_2, \dots, p_i\}$ 的最优分段需要的存储位数， S_m 是最后分段

$$S[i] = \min_{1 \leq j \leq \min\{i, 256\}} \{ S[i-j] + j \times b[i-j+1, i] \} + 11$$

$$b[i-j+1, i] = \left\lceil \log(\max_{p_k \in S_m} p_k + 1) \right\rceil \leq 8$$



$S[i-j]$ 个位

j 个灰度

$j \times b[i-j+1, i]$ 位

Compress (P, n)

伪码

1. $Lmax \leftarrow 256; header \leftarrow 11; S[0] \leftarrow 0$

2. for $i \leftarrow 1$ to n do

3. $b[i] \leftarrow length(P[i])$

4. $bmax \leftarrow b[i]$

5. $S[i] \leftarrow S[i-1] + bmax$

6. $l[i] \leftarrow 1$

7. for $j \leftarrow 2$ to $\min\{i, Lmax\}$ do

8. if $bmax < b[i-j+1]$

9. then $bmax \leftarrow b[i-j+1]$

10. if $S[i] > S[i-j] + j * bmax$

11. then $S[i] \leftarrow S[i-j] + j * bmax$

12. $l[i] \leftarrow j$

13. $S[i] \leftarrow S[i] + header$

子问题后
边界 i

最后
段长 j

找到更
好分段

$P = \langle 10, 12, 15, 255, 1, 2 \rangle$.

$S[1]=15, S[2]=19, S[3]=23, S[4]=42, S[5]=50$

$l[1]=1, l[2]=2, l[3]=3, l[4]=1, l[5]=2$

10	12	15	255	1	2
----	----	----	-----	---	---

$S[5]=50$ $1 \times 2 + 11$ 63

10	12	15	255	1	2
----	----	----	-----	---	---

$S[4]=42$ $2 \times 2 + 11$ 57

10	12	15	255	1	2
----	----	----	-----	---	---

$S[3]=23$ $3 \times 8 + 11$ 58

10	12	15	255	1	2
----	----	----	-----	---	---

$S[2]=19$ $4 \times 8 + 11$ 62

10	12	15	255	1	2
----	----	----	-----	---	---

$S[1]=15$ $5 \times 8 + 11$ 66

10	12	15	255	1	2
----	----	----	-----	---	---


$6 \times 8 + 11$ 59

追踪解

算法 Traceback (n, l)

输入：数组 l

输出：数组 C

1. $j \leftarrow 1$ // j 为正在追踪的段数
2. while $n \neq 0$ do
3. $C[j] \leftarrow l[n]$ 
4. $n \leftarrow n - l[n]$
5. $j \leftarrow j + 1$

$C[j]$: 从后向前追踪的第 j 段的长度

时间复杂度: $O(n)$

小结

- 图像变位存储问题的建模
- 子问题边界的界定
- 递推方程及初值
- 伪码
- 标记函数与解的追踪
- 时间复杂度