



[12] 发明专利申请公开说明书

[21] 申请号 02146768.4

[43] 公开日 2003年3月26日

[11] 公开号 CN 1405735A

[22] 申请日 2002.11.8 [21] 申请号 02146768.4

[71] 申请人 北京工业大学

地址 100022 北京市朝阳区平乐园100号

[72] 发明人 贾克斌 沈兰荪 庄新月 张鸿源
徐 遑

[74] 专利代理机构 北京工大思海专利代理有限责
任公司

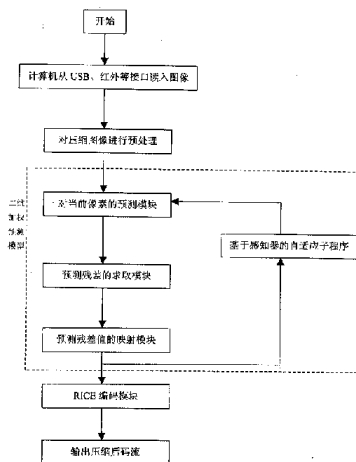
代理人 张 慧

权利要求书3页 说明书12页 附图8页

[54] 发明名称 基于感知器的彩色图像无损压缩方法

[57] 摘要

一种基于感知器的彩色图像无损压缩方法，涉及图像处理领域。本发明的特征在于具有以下处理步骤：(1)计算机从USB等接口读入待处理的目标图像后保存在内存中；(2)对待压缩图像进行现有的无损颜色空间变换；(3)采用二维加权预测模型对当前像素预测，求取预测残差值；(4)根据映射后的预测残差值，判断是否小于给定的误差限，如果超出阈值就需要自适应调整。(5)采用现有的RICE熵编码；(6)输出并保存图像的压缩结果。本发明所提供的方法复杂度低于传统压缩方法，处理后的图像具有较高的压缩比，并且在保持较高的压缩效果的同时，其压缩时间明显少于后者，具有较高的执行速度。



1、一种基于感知器的彩色图像无损压缩方法，是由数码摄像机或其他数字化仪完成采集待处理的目标图像，并将目标图像的光学信号转换为数字信号图像，计算机通过现有的 USB、红外等接口读入图像，在计算机处理器中对图像进行处理，处理后的结果图像输出到缓存器，可以直接在本地进行存储或通过网络存储设备进行远程传输，本发明的特征在于它还包括下述步骤：

- (1)计算机从 USB 等接口读入待处理的目标图像后保存在内存中；
- (2)对待压缩图像进行预处理，即对其进行现有的无损颜色空间变换；
- (3)采用一种二维加权预测模型对当前像素预测，求取预测残差值，对该预测残差值映射，其特征为：

①顺序输入颜色空间变换后的三个分量，即从左向右、从上向下依次顺序扫描和存放经过变换后的三个颜色分量的数据，图像（长度和宽度分别为 H 和 B）中的像素用 $x_k(i, j)$ 来表示；

由当前像素 $x_k(i, j)$ 左上方四个像素 $x_k(i, j-1)$ 、 $x_k(i-1, j-1)$ 、 $x_k(i-1, j)$ 、 $x_k(i-1, j+1)$ （i 表示行坐标，j 表示列坐标）的预测加权系数来预测当前值；

判断被预测的像素是否是边缘点，当被预测像素处于第一列、最后一列或第一行时，则进行特殊处理，即对于图像边缘外不存在像素的值用缺省值替代；对当前像素 $x_k(i, j)$ 的预测值 $\overline{x_k(i, j)}$ 的计算公式为：

$$\overline{x_k(i, j)} = w_1 x_k(i, j-1) + w_2 x_k(i-1, j-1) + w_3 x_k(i-1, j) + w_4 x_k(i-1, j+1) \quad (1)$$

其中 w_p ($p=1, 2, 3, 4$) 为预测加权系数，设定其初值为 $w_p=0.25$ ，从而得到当前像素的预测值 $\overline{x_k(i, j)}$ ；

②由预测残差值公式②，得到预测残差值 Δ ：

$$\Delta = x_k(i, j) - \overline{x_k(i, j)} \quad (2)$$

③在获得预测残差之后对其进行预测残差映射的方法为：假定每个单色像素是经过 8bit 量化，则预测残差值的分布范围应是 $[-255, +255]$ ，执行下面的映射算法，使其分布范围变为 $[0, +255]$ （这里预测残差值用 X 表示），

a) 根据预测模型计算得到预测残差值 X；

$$w_p(t+1) = w_p(t) + \Delta w_p(t) \quad (5)$$

$$\Delta w_p(t) = \alpha \times x_p \times d(t) \quad (p=1,2,3,4) \quad (6)$$

其中， α 是增益，通常 $0 \leq \alpha \leq 1$ ，用于控制修正速度； $\Delta w_p(t)$ 为 t 时刻连接权值 w_p 的修正值， x_p 为当前输入的第 p 个像素值；

③对每个样本重复步骤②，直至误差小于给定的误差限；

学习结束后的神经网络将学习样本模式以连接权的形式记忆下来，当神经网络输入某一模式时，神经网络将以公式③计算输出值 y ，因此整个学习记忆过程就是根据实际输出与希望输出之间的误差调整参数 w_p ，比较公式①和公式③可以看出，这里激励函数： $f(x)=x$ ，公式③中的 x_p 是当前输入的第 p 个像素值， w_p 是预测加权系数， y 是预测值；感知器实时监视预测残差值，如果这个值在编码过程中超出了预先设定的阈值，则产生一个反馈信号到感知器，通过重新学习和调整 w_p ，产生一组新的预测加权系数，实际图像编码过程中，公式为：

$$\Delta w_p = \alpha / 128 \times x_p \times \Delta \quad (p=1,2,3,4) \quad (7)$$

其中， Δw_p 为 t 时刻连接权值 w_p 的修正值， $\alpha = 0.001$ 为经验值， Δ 为上一像素点的预测残差值；这样，预测器使用修正过的预测加权系数，得到更加准确的预测值；

(6) 采用现有的 RICE 熵编码算法对映射后的预测残差进行编码；

(7) 输出并保存图像压缩的码流。

2、根据权利要求 1 所述的基于感知器的彩色图像无损压缩方法，其特征在于对待压缩图像进行预处理中所采用的无损颜色空间变换公式为：

$$\begin{cases} A_1' = \text{floor} (0.25 A_1 + 0.5 A_2 + 0.25 A_3) \\ A_2' = A_1 - A_2 \\ A_3' = A_3 - A_2 \end{cases} \quad (8)$$

其中， A_1 、 A_2 、 A_3 分别是变换前的三个颜色分量， A_1' 、 A_2' 、 A_3' 变换后得到的三个颜色分量值。

- b) 当预测残差值 X 小于-128 时, 将其加上 256 或者当预测残差值 X 大于 127 时, 将其减去 256 后得到一个新的预测残差值 $X1$;
- c) 新的预测残差值 $X1$ 如果小于零将其按照 $X2 = -X1 \times 2 - 1$, 否则将其按照 $X2 = 2 \times X1$ 得到映射后的预测残差值 $X2$;
- d) 输出映射后的预测残差值 $X2$;
- (4) 根据以上的映射后的预测残差值结果, 判断预测残差是否小于给定的误差限 8, 如果映射后的预测残差值超出这个阈值, 就需要修改预测加权系数, 即实现基于感知器的自适应调整子程序;
- (5) 当进行基于感知器的自适应调整子程序的特征如下:

使用现有技术中的神经网络中的双层感知器来实时监视和自适应地调整预测残差的漂移, 输出值可以由以下公式来表示:

$$y = f\left(\sum_{p=1}^n w_p x_p\right) \quad (3)$$

其中, $f(\cdot)$ 为激励函数, $w_1, w_2 \dots w_n$ 为从输入处理单元 p ($p=1,2,\dots,n$) 到输出处理单元的连接权值, w_0 为阈值, $x_1, x_2 \dots x_n$ 为处理单元 p 的输入, y 为输出值;

感知器的训练采用由一组样本组成的集合来进行, 在训练期间, 将这些样本重复送到感知器的输入层, 通过调整连接权值使感知器的输出达到理想输出, 其训练过程如下:

- ①初始化, 给每个 w_p 赋初值, 赋给 $w_p(t)$ 以 $(-1, +1)$ 区间的随机值; 这里 $w_p(t)$ 表示 t 时刻第 p 个输入上的连接权值, $1 \leq p \leq n$, $w_0(t)$ 为 t 时刻的阈值;
- ②连接权值的修正, 输入一个样本 $X = (x_1, x_2 \dots x_n)$ 和它的期望输出 d ;
- (a) 按公式③计算网络实际输出 $y(t)$;
- (b) 计算网络实际输出 $y(t)$ 与期望输出 d 之间的误差:
- $$d(t) = d - y(t) \quad (4)$$
- (c) 修正各输入输出间的连接权值 w_p :

基于感知器的彩色图像无损压缩方法

技术领域

本发明涉及图像处理领域，设计和实现了一种基于神经网络感知器的彩色图像无损压缩方法。

背景技术

随着近年来多媒体技术的广泛应用，在一些重要的图像信息的存储和传输应用中，图像无损压缩技术发挥着重要和不可替代的作用，它可以在很大程度上减轻对存储媒体容量和传输带宽的要求。特别是在反复多次进行存贮的应用系统中，为了保持图像原有的质量，一般要求必须采用无损压缩技术。

由于图像像素之间存在有冗余信息，利用图像数据之间的相关性，通过重新编码，去除图像数据中的冗余信息，就可以达到图像压缩的目的。至于彩色图像可以看作是多分量的单色数据图像顺序扫描或插值扫描排列而成。图1中(b)和(c)分别是标准图像(a)中各颜色平面内像素的自相关函数和各颜色平面之间像素的互相关函数。比较图(b)和图(c)可以看到，同一单色平面内相邻像素之间的相关性要高于不同单色平面内相应位置像素之间的相关性。对于图(a)来说，G平面与B平面像素之间的相关性要高于R与G或R与B之间的相关性。据彩色图像的这种特性，可以来设计彩色图像压缩方案。

针对于静止图像无损压缩，国际组织ITU-T(International Telecommunications Union--国际电信同盟)原名为CCITT(International Telephone and Telegraph Consultative Committee---国际电话与电报顾问委员会)和ISO(国际标准化组织)/IEC(国际电工委员会)于1992年联合制定了无损压缩的国际标准，将基于DPCM(Differential Pulse Code Modulation---线性预测编码)的预测器和Huffman编码推荐为无损压缩编码算法JPEG标准的早期版本。但这些传统的DPCM或对其进行简单改进的固定模式的预测器，并不能得到完全不相关的预测数据。使得算法压缩比较低，特别是压缩时间较长，远不能满足当前应用的需要。

图像数据的无损压缩方案一般包括建模（modeling）和编码(coding)两个部分。算术编码方法的出现，可以将这两部分从概念上分开，对模型映射后的数据码字从概率上进行分配。而研究人员可以集中精力研究和设计各种高效的预测模型，从而使得无损压缩无论在压缩比和压缩效率上都比早期的 JPEG 标准算法得到很大的提高。JBIG/JPEG（联合图像专家组）委员会接纳的新标准：JPEG-LS 是采用 LOCO-I (LOW COMplexity LOSSless COMpression for Images) 算法作为其核心内容的一种无损/近无损图像压缩算法。该算法采用二维预测，并通过引入上下文(Context)对预测值进行自适应校正，对得到的残差信号进行 Golomb 编码，当预测器发现有连续相同的像素时，就从正常模式切换到游程编码模式。这种算法对于一般图像压缩比可达 2.5 倍以上，压缩效率较高而受到广泛关注。但是 JPEG-LS 算法中的上下文模型（context modeling）增加了算法的复杂度。

本发明引入现有的神经网络中的感知器技术，通过感知器的自学习和自适应能力校正本发明提出的二维加权预测模型的参数，简化了预测模型部分的算法复杂度。熵编码部分中，考虑到目前常用的熵编码的方法主要有 Huffman、算术编码、LZW 等。这些熵编码虽然具有较好的编码效果，但其执行速度较慢。本发明采用了现有的 RICE 熵编码算法。该熵编码算法在对数据进行扫描过程中，就可以根据采样值实时进行计算和编码，无需求助于事先存贮的码表，提高了熵编码得速度。在编码过程中，它还具有自适应功能，以适应码长的变化。它的复杂度低，编码速度快，并且有较高的编码效率。

发明内容

本发明通过提出了一种二维加权预测模型（简称预测模型），设计并实现了整个彩色图像无损压缩方法。该方法克服了以往图像无损压缩算法中执行速度慢，算法复杂度高以及压缩效率低等缺点。兼顾了较低的算法复杂度的同时，达到了快速、高效的图像无损压缩效果，压缩比较高。根据彩色图像的颜色分量间与像素间相关性的分析，首先利用现有的无损颜色空间变换对颜色空间进行处理，消除颜色分量间的相关性。在本发明提出的二维加权预

测模型中，引入了现有的神经网络中的感知器技术。利用其本身具有的自学习和自适应特性，进行二维加权预测模型的自适应调整。使该算法在运行过程中具有很小的预测残差。进一步根据本发明所提出的预测残差映射算法，对预测残差值进行映射，减小了预测残差值的动态范围。保证了算法在具有较高压缩比的前提下，有较高的执行效率。得到预测残差后，需要进行熵编码，本文编码方法采用了现有技术中的 RICE 算法。该熵编码算法复杂度低，编码速度快，并且有较高的编码效率。在整个压缩算法的执行过程中，保证不引入量化误差，并在解码过程中完全恢复原来的数据，从而实现图像的无损压缩。

本发明的技术思路特征为：

1. 针对颜色平面间的相关性，采用现有的无损颜色空间变换的方法，对目标图像进行图像预处理来去除颜色空间的相关性，从而可以有效的提高压缩比。
2. 假设图像数据是按照从左至右、从上向下的顺序，依次输入三个经过空间变换后的颜色分量数据。由于同一分量相邻像素之间存在有很强相关性，根据当前像素左上方四个相邻像素的加权值，可以通过本发明所提出的预测算法，获得当前像素的预测值。
3. 如果这种预测值与当前的实际值很接近，则它们之间的差值称为预测残差。求取预测残差值，如果该值足够小，可使编码器用很短的码字对其进行编码。
4. 实际算法中，得到预测残差值并没有直接进行编码，而是先经过了一个本发明所提出的映射过程。通过数据映射算法，减小了预测残差值的动态范围，使用于编码的预测残差信号的分布范围更加集中，从而可以获得更小的熵并获得更好的编码效果。
5. 根据以上的映射后的预测残差值结果，判断预测残差是否小于给定的误差限 δ ，如果映射后的预测残差值超出这个阈值，就需要修改预测加权系数，即实现基于感知器的自适应调整子程序。
6. 在预测和编码过程中，采用了一种感知器的技术来自适应地校正二维加权预测模型的参数。通过感知器的自学习和自适应能力对预测残差值进行监视，

如果其值超过设定的阈值，则进行自适应调整，使得整个编码过程中，预测残差值保持在一个很小的范围内，从而达到压缩的目的。

7.对映射后的预测残差值进行 RICE 编码。

本发明的技术方案参见图 3、图 4。这种基于感知器的彩色图像无损压缩方法，是由数码摄像机或其他数字化仪器完成采集待处理的目标图像，并将目标图像的光学信号转换为数字信号图像输入到计算机进行处理、传输等操作。计算机处理主要是通过现有的 USB 接口软件，在现有的彩色图像无损压缩及神经网络中的感知器技术的基础上对图像进行压缩处理。处理后的结果图像输出到缓存器，可以直接在本地进行存储或通过网络存储设备进行远程传输等操作。本发明的特征在于它还包括下述步骤：

- (1)计算机从 USB 等接口读入待处理的目标图像后保存在内存中；
- (2)对待压缩图像进行预处理，即对其进行现有的无损颜色空间变换；
- (3)采用一种二维加权预测模型对当前像素预测，求取预测残差值，对该预测残差值映射，其特征为：

这部分处理的流程图见图 6 中区域 A。

①为了对整个算法进行描述，假设图像采用 RGB 颜色空间并经过颜色空间变换的预处理。顺序输入颜色空间变换后的三个分量，即从左向右、从上向下依次顺序扫描和存放经过变换后的三个颜色分量的数据。图像（长度和宽度分别为 H 和 B）中的像素用 $x_k(i, j)$ 来表示。这里 $i=0, 1, \dots, B-1$ 表示某个颜色平面中像素所在的行， $j=0, 1, \dots, H-1$ 表示列， $k=R, G, B$ ，表示变换后像素所在的颜色平面。

由当前像素 $x_k(i, j)$ 左上方四个像素 $x_k(i, j-1)$ 、 $x_k(i-1, j-1)$ 、 $x_k(i-1, j)$ 、 $x_k(i-1, j+1)$ (i 表示行坐标， j 表示列坐标) 的加权值来预测当前值，它们的分布位置见图 5，图中 x_1 、 x_2 、 x_3 、 x_4 为代表 $x_k(i, j-1)$ 、 $x_k(i-1, j-1)$ 、 $x_k(i-1, j)$ 、 $x_k(i-1, j+1)$ 四个像素， X 为当前像素。判断被预测的像素是否是边缘点，当被预测像素处于第一列、最后一列或第一行时，则进行特殊处理，即对于图像边缘外不存在像素的值用缺省值替代；对当前像素 $x_k(i, j)$ 的预测值

如果仅仅采用上述固定的预测模型，并不能保证在整个编码过程中预测值不发生偏移，从而使用于编码的预测残差发生漂移。在本发明所提出的算法中，采用了神经网络中的双层感知器来实时监视和自适应地调整预测残差值的漂移。

感知器的训练过程以及预测加权系数的修改方法如下：

双层感知器的网络结构如图 2 所示，它是由输入层和输出层组成。输入层的各个处理单元与输出层的每个处理单元互连，在每个连接中均附有权值，这些权值可以通过学习规则来调整。它的学习过程是改变神经元之间的连接强度的过程。

输出值可以由以下公式来表示：

$$y = f\left(\sum_{p=1}^n w_p x_p\right) \quad (3)$$

其中， $f(\cdot)$ 为激励函数， $w_1, w_2 \dots w_n$ 为从输入处理单元 p ($p=1,2,\dots,n$) 到输出处理单元的连接权值， w_0 为阈值， $x_1, x_2 \dots x_n$ 为处理单元 p 的输入， y 为输出值；

感知器的训练采用由一组样本组成的集合来进行，在训练期间，将这些样本重复送到感知器的输入层，通过调整连接权值使感知器的输出达到理想输出，其训练过程如下：

①初始化，给每个 w_p 赋初值，赋给 $w_p(t)$ 以 $(-1, +1)$ 区间的随机值；这里 $w_p(t)$ 表示 t 时刻第 p 个输入上的连接权值， $1 \leq p \leq n$ ， $w_0(t)$ 为 t 时刻的阈值；

②连接权值的修正，输入一个样本 $X = (x_1, x_2 \dots x_n)$ 和它的期望输出 d ；

(a)按公式③计算网络实际输出 $y(t)$ ；

(b) 计算网络实际输出 $y(t)$ 与期望输出 d 之间的误差：

$$d(t) = d - y(t) \quad (4)$$

(c) 修正各输入输出间的连接权值 w_p ：

$$w_p(t+1) = w_p(t) + \Delta w_p(t) \quad (5)$$

$\overline{x_k(i, j)}$ 公式为:

$$\overline{x_k(i, j)} = w_1 x_k(i, j-1) + w_2 x_k(i-1, j-1) + w_3 x_k(i-1, j) + w_4 x_k(i-1, j+1) \quad ①$$

其中 w_p ($p=1,2,3,4$) 为预测加权系数, 设定其初值为 $w_p=0.25$, 从而得到当前像素的预测值 $\overline{x_k(i, j)}$;

②由预测残差值公式②, 得到预测残差值 Δ :

$$\Delta = x_k(i, j) - \overline{x_k(i, j)} \quad ②$$

如果在整个编码过程中, 都能够保证预测很准, 就可以始终得到足够小的预测残差值, 从而有利于后续的熵编码。这部分处理过程的流程图见图 6 中区域 C。

③在获得预测残差之后对其进行预测残差映射方法为: 假定每个单色像素是经过 8bit 量化, 则预测残差值的分布范围应是 $[-255, +255]$ 。执行下面的映射算法, 使预测残差值的分布范围变为 $[0, +255]$ (这里预测残差值用 X 表示)。映射的整个流程见图 7。

- a) 根据预测模型计算得到预测残差值 X;
- b) 当预测残差值 X 小于 -128 时, 将其加上 256 或者当预测残差值 X 大于 127 时, 将其减去 256 后得到一个新的预测残差值 X1;
- c) 新的预测残差值 X1 如果小于零将其按照 $X2 = -X1 \times 2 - 1$, 否则将其按照 $X2 = 2 \times X1$ 得到映射后的预测残差值 X2;
- d) 输出映射后的预测残差值 X2;

通过数据映射算法, 减小了预测残差值的动态范围, 使用于编码的预测残差信号的分布范围更加集中, 从而可以获得更小的熵并获得更好的编码效果;

(4)根据以上映射后的预测残差结果, 判断预测残差是否小于给定的误差限 δ , 如果小于这个阈值则直接进行熵编码; 如果预测残差值超出这个阈值, 就需要修改预测加权系数, 即实现基于感知器的自适应调整的子程序;

(5)当进行基于感知器的自适应调整的子程序的特征如下:

$$\Delta w_p(t) = \alpha x_p \times d(t) \quad (p=1,2,3,4) \quad \textcircled{6}$$

其中， α 是增益，通常 $0 \leq \alpha \leq 1$ ，用于控制修正速度； $\Delta w_p(t)$ 为 t 时刻连接权值 w_p 的修正值， x_p 为当前输入的第 p 个像素值；

③对每个样本重复步骤②，直至误差小于给定的误差限；

学习结束后的神经网络将学习样本模式以连接权的形式记忆下来，当神经网络输入某一模式时，神经网络将以公式③计算输出值 y ，因此整个学习记忆过程就是根据实际输出与希望输出之间的误差调整参数 w_p ，比较公式①和公式③可以看出，这里激励函数： $f(x)=x$ ，公式③中的 x_p 是当前输入的第 p 个像素值， w_p 是预测加权系数， y 是预测值；感知器实时监视预测残差值，如果这个值在编码过程中超出了预先设定的阈值，则产生一个反馈信号到感知器，通过重新学习和调整 w_p ，产生一组新的预测加权系数，实际图像编码过程中，公式为：

$$\Delta w_p = \alpha / 128 \times x_p \times \Delta \quad (p=1,2,3,4) \quad \textcircled{7}$$

其中， Δw_p 为 t 时刻连接权值 w_p 的修正值， $\alpha=0.001$ 为经验值， Δ 为上一像素点的预测残差值；这样，预测器使用修正过的预测加权系数，得到更加准确的预测值；

由于感知器有学习记忆、自适应调整连接权值的能力，这种模型显然有可能用来自适应调整图像压缩算法中的预测加权系数，获得比固定预测加权系数更高的压缩比。通过实际输出和希望输出之间的差值来自适应地调整预测加权系数，也就是感知器的学习过程。

感知器实时监视预测残差值，如果这个值在编码过程中超出了预先设定的阈值，则产生一个反馈信号到感知器，通过重新学习和调整 w_p ，产生一组新的预测加权系数。采用公式⑤、⑥修改预测加权系数，这部分处理过程的流程图见图 6 中区域 B。

可以看到在自适应调整算法中，感知器模型采用的是一种边学习边工作的模式，无需进行先期训练。对图像扫描一次，预测和编码同时完成。具有

很高的执行效率；

(6)采用现有的 RICE 熵编码算法对映射后的预测残差进行编码；

(7)输出并保存图像压缩的码流。

当进行预处理，即对图像进行现有的无损颜色空间变换，其原理如下：

对于多分量（彩色）的图像，通过一定的无损颜色空间变换，可以在一定程度上去除颜色分量间的相关性，JPEG-LS、JPEG2000 和其他彩色图像压缩方法都精心设计了颜色空间变换方法。本发明中的空间变换是根据公式①进行。其中， A_1 、 A_2 、 A_3 分别是变换前的三个颜色分量， A_1' 、 A_2' 、 A_3' 变换后得到的三个颜色分量值。

$$\begin{cases} A_1' = \text{floor} (0.25 A_1 + 0.5 A_2 + 0.25 A_3) \\ A_2' = A_1 - A_2 \\ A_3' = A_3 - A_2 \end{cases} \quad \textcircled{8}$$

本发明所提的方法复杂度低于传统压缩方法，处理后的图像具有较高的压缩比，并在保持较高的压缩效果的同时，其压缩时间明显少于后者，具有较高的执行速度。熵编码部分中，采用了 RICE 熵编码算法，该熵编码算法在对数据进行扫描过程中，可以根据采样值实时进行计算和编码，无需求助于事先存贮的码表，提高了熵编码得速度。在编码过程中，它还具有自适应功能，以适应不同码长的变化。它的复杂度低，编码速度快，并有较高的编码效率。

附图说明：

图 1 是彩色图像相关性分析图：

(a) 标准图像；(b) 同一分量平面内像素的自相关函数；(c) 不同分量平面间像素的互相关函数；

图 2 是双层感知器结构图；

图 3 是基于感知器的快速彩色图像无损压缩系统框图；

1、数字摄像机，2、USB 接口，3、计算机处理器，4、输出缓存，5、图像的无损压缩编码，6、压缩的结果文件，7、硬盘，8、网络存储设备，9、解码，10、显示器；

图 4 是本发明的方法流程图；

图 5 是预测器中四个像素及当前像素的位置图；

图 6 是对当前像素的预测模块及自适应调整流程图；

图 7 是预测残差值的映射模块流程图；

图 8 是解码流程图；

图 9 是实施例的原始图像，及其 RGB 各分量；

图 10 是经预处理后的三个颜色分量，a 图为 A1'分量，b 图为 A2' 分量，c 图为 A3' 分量；

图 11 是图像压缩算法子程序流程图；

图 12 是本发明的程序流程图。

具体实施方式

图 3 中数码摄像机和 USB 接口都是市售的，主要完成采集需要进行压缩的各种图像，将待处理图像的光学信号转换为数字信号图像输入到计算机，便于计算机处理、传输等操作。计算机处理主要是通过现有的 USB 接口软件读入采集到的图像，处理后的图像文件输出到计算机的缓存器，便于在本地存储（存入硬盘）；通过网络存储设备进行数据的传输、存储、处理；或者直接进行解码，其中解码的流程图见图 8，然后显示。图像的无损压缩编码是通过软件来实现的。下面结合具体实例详细描述图像压缩的整个过程，参见图 12。整个过程步骤见前述发明的技术思路特征，实施过程可参见图 12 和图 11：

首先通过数码相机对目标图像进行采集，数码摄像机将小区车库管理过程中在车进、出时分别抓拍一帧图像，然后通过 USB 接口传到计算机硬盘中，这样就完成了实施例监控图像的采集过程。

具体实施中将图像传输到计算机后，在计算机中完成以下程序：

第一步：图 9 是实施例的原始图像，即用户要压缩的图像(24 色真彩，jpg 格式)及其 RGB 各分量，图像大小为 (142×107)。对该图像进行无损颜色空间变换，得到去除颜色分量间相关性后的结果图像（这一部分独立于主程序）。A1、A2、A3 三个分量经处理后的结果分别对应为图 10 中 a 图、b 图、c 图所

示：

第二步：打开上一步的结果图像。

第三步：创建一个压缩文件，用来存放压缩后的数据。

第四步：提取经过无损颜色空间变换的图像的图像信息。

第五步：执行图像压缩算法子程序，具体步骤见图 11 的图象压缩算法流程图：

(1)读入一行数据，判断是否为边界点，若是边界点，则对其进行特殊处理，即对于图像边缘外不存在像素的值用缺省值替代；若不是边界点，则对预测器赋值，预测器的预测加权系数数组的初值为： $W[1]=0.375$ ； $W[2]=0.375$ ； $W[3]=0.125$ ； $W[4]=0.125$ 。

(2)采用一种二维加权预测模型对当前像素预测，求取预测残差值，对该预测残差值映射：

①利用同一分量相邻像素之间存在有很强相关性这一特点，根据当前像素左上方四个相邻像素（假设图像数据是按照从左至右、从上向下的顺序，依次输入三个经过无损颜色空间变换后的颜色分量数据），可以通过一定的预测算法，获得当前像素的预测值。

②计算预测值与实际值的差称为预测残差。当该值足够小时，使编码器用很短的码字对其进行编码；

③对得到的预测残差进行残差映射处理。通过数据映射算法，减小了预测残差值的动态范围，使用于编码的预测残差信号的分布范围更加集中，从而可以获得更小的熵并获得更好的编码效果。

(3)根据以上的映射后的预测残差值结果，判断预测残差是否小于给定的误差限 8，如果映射后的预测残差值超出这个阈值，就需要修改预测加权系数，即调用基于感知器的自适应调整子程序。

(4)进行基于感知器的自适应调整子程序。

(5)对映射后的预测残差进行现有的 RICE 熵编码。

在此过程中，采用本发明提出的利用现有感知器技术来自适应地校正二维

加权预测模型参数的方法。通过感知器的自学习和自适应能力对预测残差值进行监视，如果其值超过设定的阈值，则进行自适应调整并进行映射，使得整个编码过程中，预测残差值保持在一个很小的范围内，从而达到压缩的目的。自适应调整二维加权预测模型的参数的整个流程图见图 6。在此取神经元数目 n 为 1，输入节点数 m 为 4，增益 α 为 0.00001。

(6)判断是否到行尾，如果该行已经结束则保存当前行；否则，再次判断是否为边界点，重复上述步骤；

(7)判断是否到列尾，若到列尾，则结束该子程序；否则重复(1)到(7)步；

第六步：打印：“Mission completed!”、“CompressedCounter=编码后的字节数”、“See compressed data in file compress.dat”

第七步：关闭用户要压缩的文件以及经压缩后的文件。

第八步：得到压缩后的文件，存储在本地的硬盘中，从而完成了对车库的监控记录存储。

为了检验本发明所提出的算法的性能，与文献中传统的 Huffman 算法、LZW 算法、FELICS 算法，特别是新的无损压缩标准 JPEG-LS 进行了比较。用于测试的图像全都是 RGB 空间的 24 位（每个像素分量 8 bits）彩色图像。包括两幅风景图像、一幅航空图像、一幅计算机合成图像。为了使结果更具有说明性，还选用了两幅标准彩色图像 peppers 和 lena 用于对比实验。Huffman 算法、LZW 算法和 JPEG-LS 算法的结果是通过运行从网页上下载的执行程序得到的，FELICS 算法的结果是从文献 Barequet R, Feder M. SICLIC: A simple inter-color lossless image coder. In *Proc. of data compression conference*, Edited by J. A. Storer, Snowbird Utah, USA, March, 1999, pp.500-510.中得到的。整个实验是在 300MHz 的 Pentium II 、内存为 64M 的 PC 机上用 C 语言完成的。

表 1 给出了对比压缩结果，它是用每个像素的位数来表示的（=8 bit/压缩比）。从表 1 中可以看到，本发明所提的方法其压缩结果明显优于 LZW 算法和传统的 Huffman 算法，比 FELICS 算法的结果也好。虽然 JPEG-LS 的压缩结果要好于本发明算法的结果，但是二者的结果相当接近，特别是对于两幅

风景图像（平滑的自然图像），本发明的结果只比 JPEG-LS 算法差 3%左右。

从表 2 所示的压缩时间来看，本发明算法执行速度明显快于传统的 Huffman 和 LZW 算法，比低复杂性高效率的 JPEG-LS 算法也快 13-17%。从算法的结构上比较，本算法和 JPEG-LS 在预测和编码部分都相当，但是 JPEG-LS 算法中增加了上下文模型（context modeling），其复杂度远大于本发明中所使用的感知器预测模型，虽然它提高了压缩比，但执行速度受到影响。

表 1: 压缩结果的比较

图像	宽*高	压缩结果(bits/pixel)				
		Huffman	LZW	FELICS	JPEG-LS	本发明
风景 1	2048*1536	2.92	5.01	*	2.13	2.20
风景 2	2056*2400	2.27	4.20	*	1.83	1.89
航空	2200*1196	2.65	3.92	*	1.90	2.22
计算机合成	1636*1070	2.47	3.08	*	2.21	2.55
Peppers	512*512	4.67	5.84	5.14	3.88	4.28
Lena	512*512	5.12	6.78	4.84	4.52	4.76

表 2: 压缩时间的比较

图像	宽*高	压缩时间(s)			
		Huffman	LZW	JPEG-LS	本发明
风景 1	2048*1536	6.88	9.87	6.21	5.10
风景 2	2056*2400	9.35	13.32	8.68	7.27
航空	2200*1196	6.83	7.90	6.62	5.55
计算机合成	1636*1070	5.24	8.68	4.46	3.89
Peppers	512*512	2.18	3.12	0.60	0.51
Lena	512*512	2.87	3.89	1.05	0.90

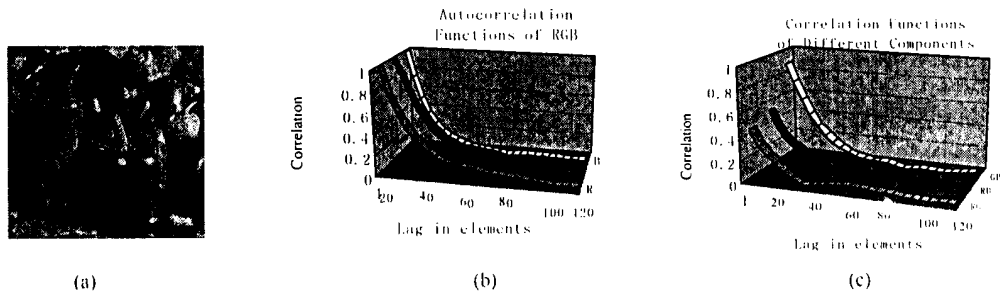


图 1

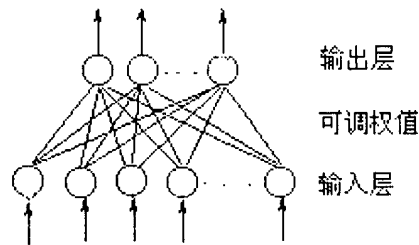


图 2

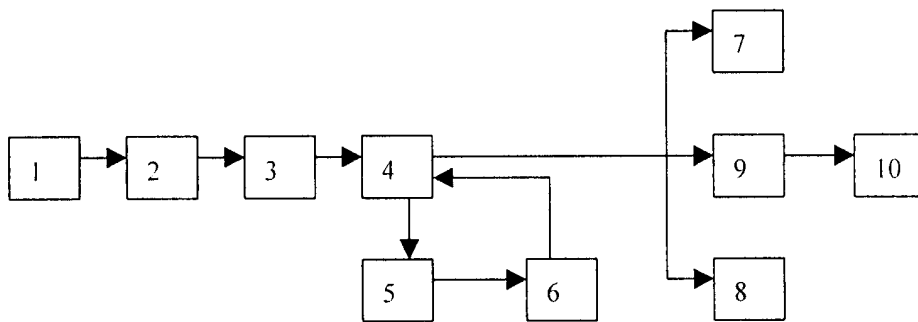


图 3

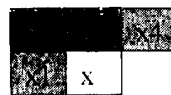


图 5

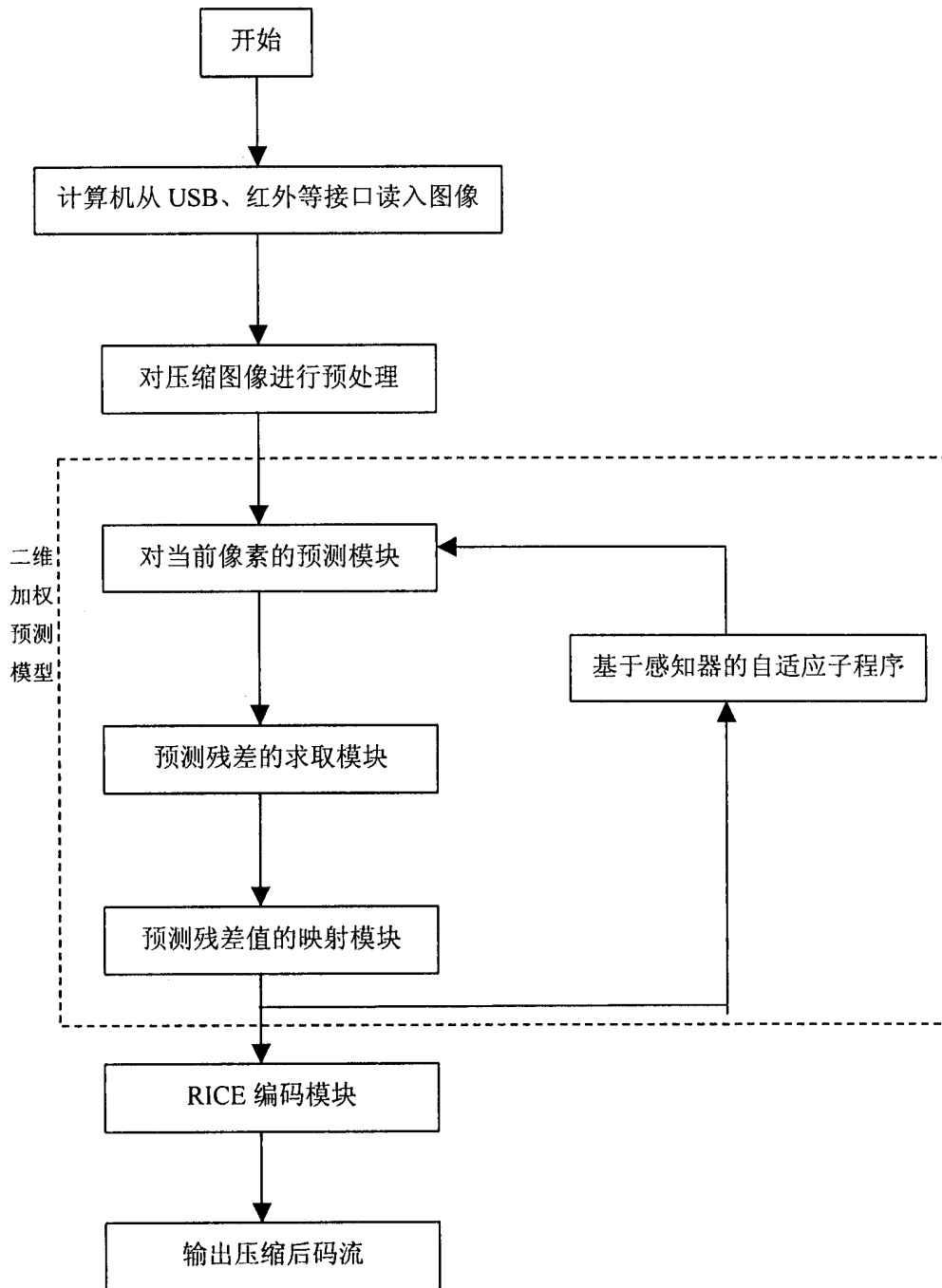


图 4

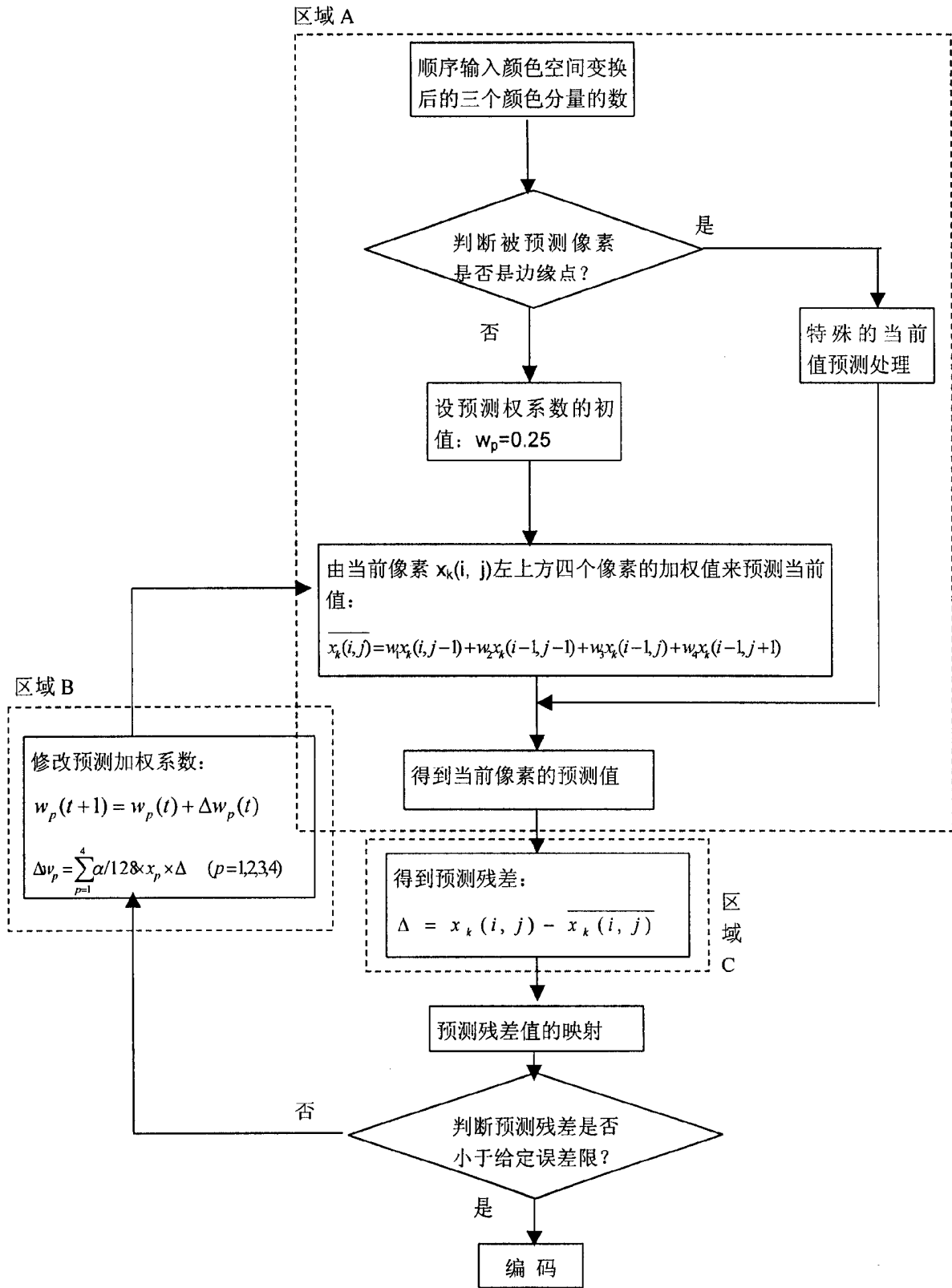


图 6

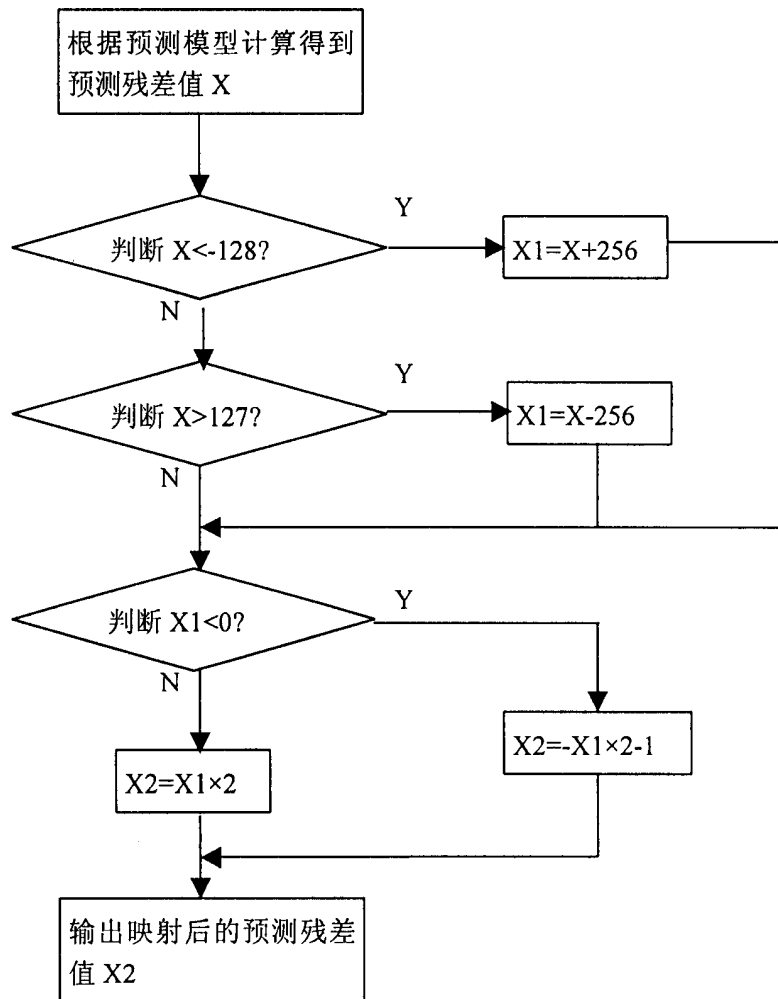


图 7

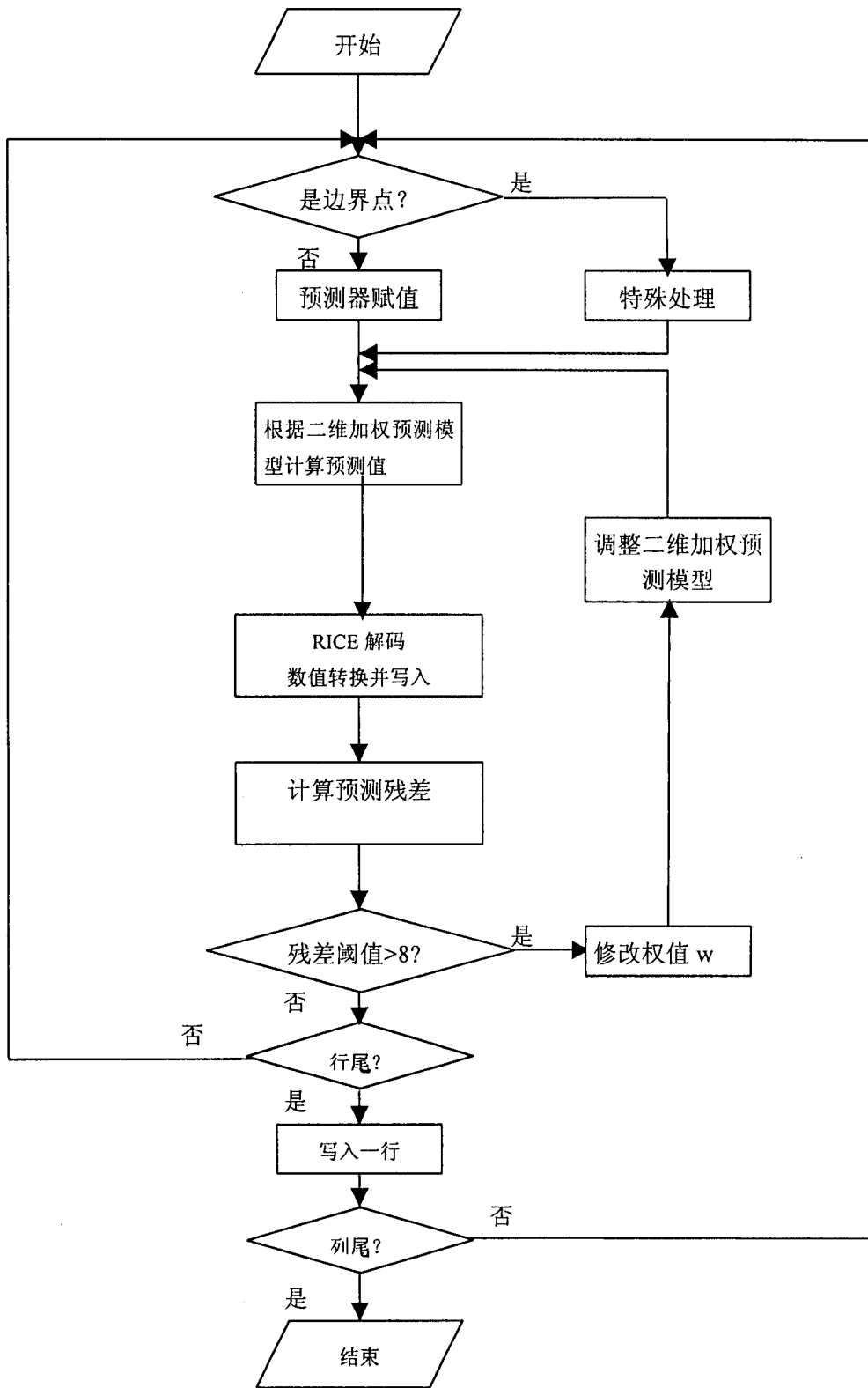


图 8

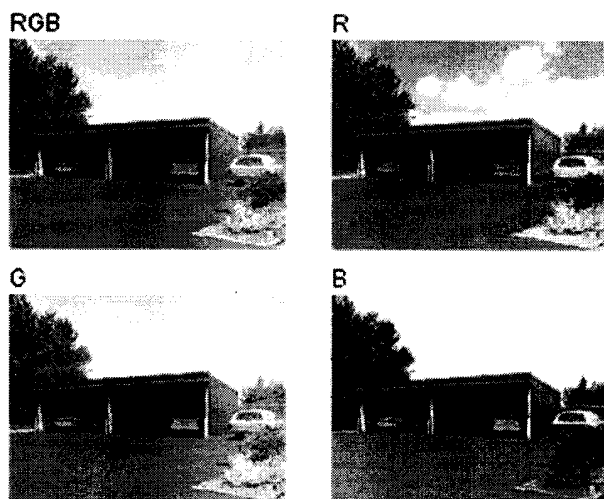


图 9

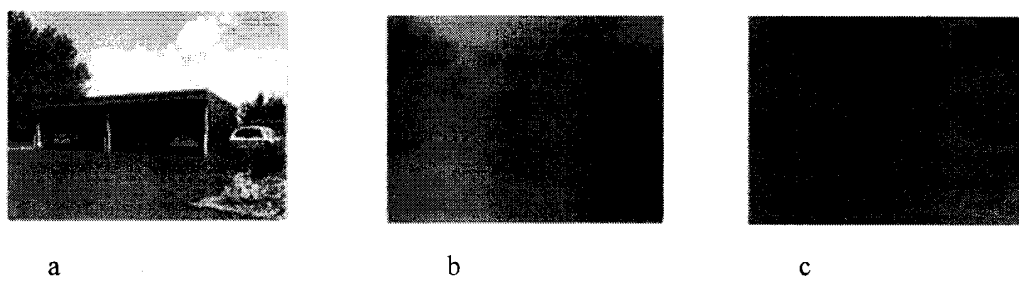


图 10

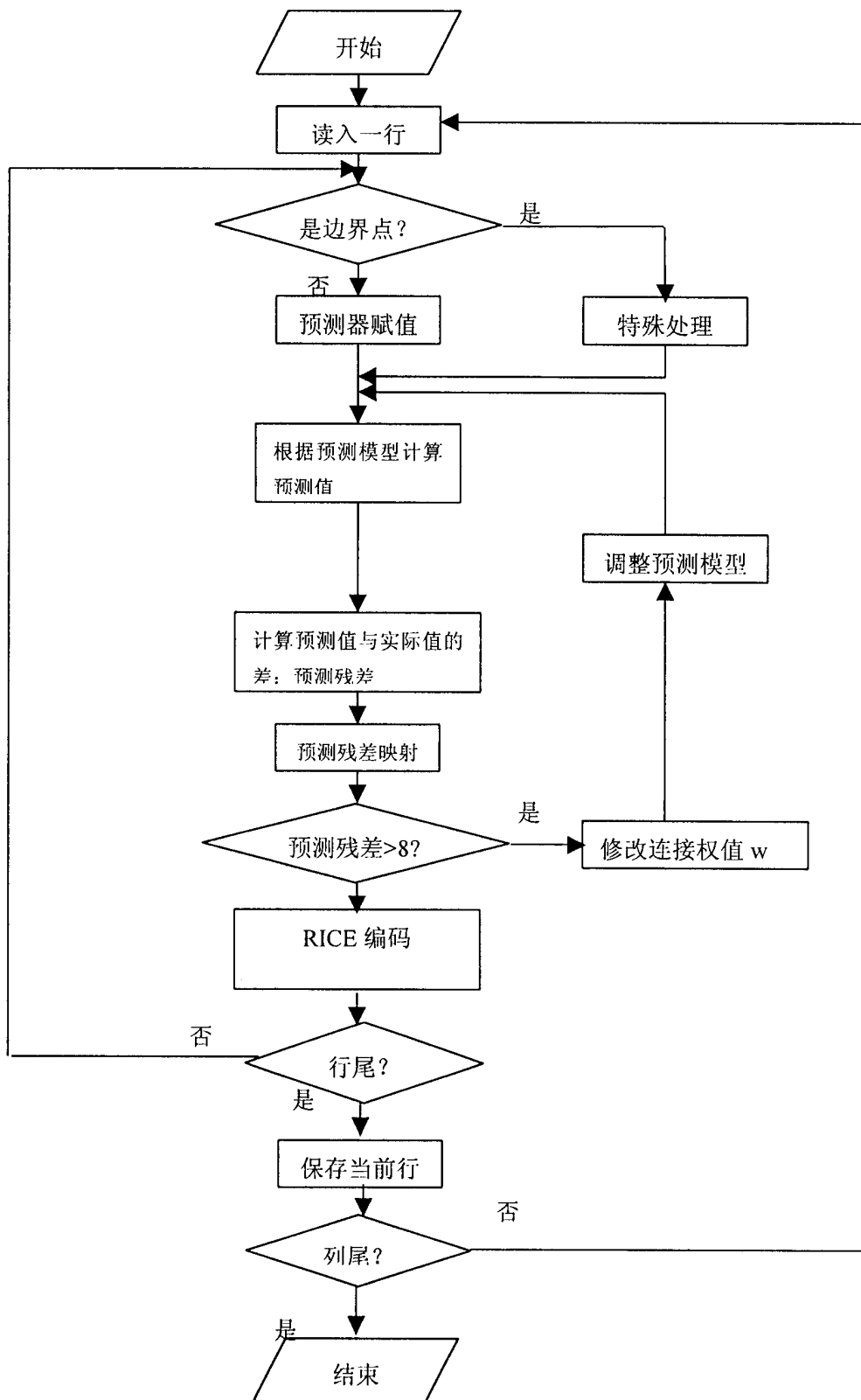


图 11

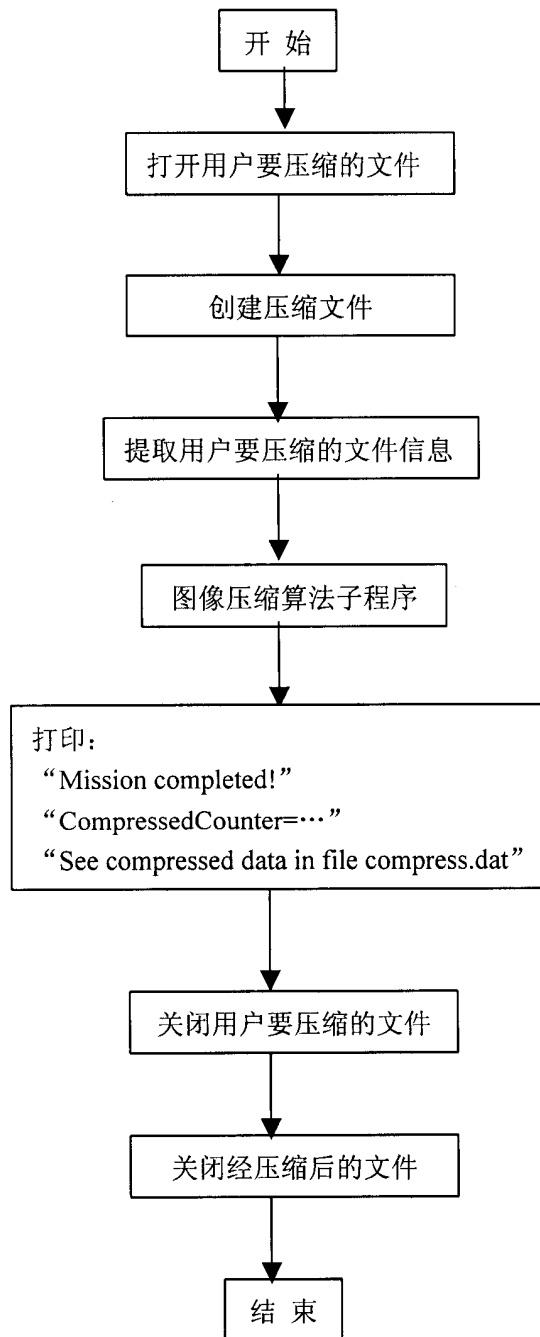


图 12