一种基于微分思想的图象变形算法研究与实现

邹北骥[1,2,3],周浩宇[2,3],吕格峰[2,3],孙家广[1]
(1. 清华大学计算机科学与技术系,北京 100084;2. 湖南大学计算机与通信学院,湖南长沙 410082;
3. 中国科学院软件研究所计算机科学重点实验室,北京 100080)

摘 要: 图象变形是图象处理中的基本问题,不仅要求图象变形效果好,而且要求变形算法效率高. 目前已有的变形算法,大多采用基于象素点的填充方法,这种方法有比较明显的缺点:(1) 当变形的图象区域是一个非规则区域时,图象变形处理所花费的时间增加,即算法的时间复杂度加大,效率低;(2) 在对象素点作映射变换时,可能出现变形后的图象区域中一个象素点对应变形前的图象区域中多个象素点,因此造成取舍困难,甚至造成取舍错误. 为解决这一问题,本文提出了一种基于微分思想的图象变形算法,该算法先将复杂的变形区域划分为一系列子区域,再将每个子区域划分为多个小矩形,将对应的小矩形按照标准矩形填充算法进行快速填充,由此实现图象变形. 实验结果表明,该算法实现简单,计算速度快,填充效果好,并已成功地应用于笔者开发的基于真实照片的人脸整形与美容图象处理系统中.

关键词: 图象变形; 填充; 算法
中图分类号: TP391 **文献标识码**: A **文章编号**: 0372-2112 (2003) 05-0674-05

Research on an Image Deformation Algorithm Based on Calculus and Its Implementation

ZOU Bei-ji[1,2,3], ZHOU Hao-yu[2,3], LV Ge-feng[2,3], SUN Jia-guang[1]
(1. *Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*;
2. *College of Computer and Communication, Hunan University, Changsha, Hunan* 410082, *China*;
3. *Laboratory of Computer Science, Institute of Software, Chinese Academy, Beijing* 100080)

Abstract: The image deformation often appears in the image processing system. So it is necessary to develop a highly efficient algorithm to get a good result of the deformation. But most of the existing deformation algorithms are based on Pixel-Fill-Method, and there are some obvious faults in these algorithms: (1) When the shape of the deformation region is complex, the efficiency of these algorithms is lower. (2) Some errors may occur when the pixels are mapped from the original image region to the destination image region since one target pixel may be relative to several source pixels. To solve these problems, a new deformation algorithm based on calculus is presented in this paper. In this algorithm a complex shape deformation region will be divided into several sub-regions firstly, and then each sub-region will be divided into many Small-Rectangles, at last a standard rectangle fill algorithm is applied to fill pixels in the corresponding rectangles. The experimental results from our algorithm show that there is a better effectof image deformation and a higher efficiency when using our algorithm. And our algorithm has been successfully applied to a human face beautifying system based on its real photo which was developed by the authors.

Key words: image deformation; fill; algorithm

1 引言

图象变形在许多计算机图形与图象处理系统中有着重要的应用. 它通常要求将一种形状的图象真实自然地变换为另一种形状的图象. 由于图象区域形状的改变,变换前后的图象区域除了形状发生变化外,其面积大小也将发生变化. 若变形后的区域面积增大,则需要补充更多的填充象素,反之填充的象素数比原区域图象象素数少. 但无论是区域面积增大还是减少,我们必须保证变形前后的图象有很大的相似性. 这种图象变形在笔者设计与开发的"基于真实照片的人脸整形与美容图象处理系统"中得到广泛应用,并获得良好的效果. 该系统设计了许多整形与美容的虚拟操作,如:对眼睛实施放大操作、对鼻子实施隆鼻操作、对嘴巴实施嘴唇修薄操作等,这些操作本质上是一致的,都是一种基于图形的图象变形,即用图

收稿日期:2002-08-09;修回日期:2002-12-02
基金项目:本项研究得到中国科学院软件研究所计算机科学重点实验室开放课题

形的方法产生曲线来确定变形前后的图象区域,用填充的方法获得变形后的图象,并保证与变形前的图象相似.

变形后的图象区域的填充是变形算法的关键,填充算法的好坏直接影响变形后的图象效果,同时填充效率也是一个重要问题.目前国内外就图象填充问题开展了很多的研究工作,提出了一系列的算法.文献[1]提出了一种由矩形图象变换为任意四边形图象的填充算法,该算法是基于象素点并通过线性插值的方法来填充变形后的图象区域的;文献[2]提出的一种图象变形映射算法主要是将矩形区域的二维图象向任意的二维曲线轮廓区域映射而达到图象变形的目的,同样这种算法也是基于象素点进行填充的.这些算法由于是基于象素点来计算填充象素的位置,因而时间复杂度大,效率低.在研究这些算法的基础上,本文提出了一种基于微分思想的图象变形算法,即将变形前后的图象区域细分为一系列的矩形,在图象变形前后的区域之间建立矩形之间的映射关系,达到快速填充的目的.将这种填充算法应用于人脸整形与美容系统中,不仅效率高,而且效果好.

在笔者开发的基于真实照片人脸整形与美容系统中,设计了多种操作算子,如:针对眼睛部位的美容,最常用的操作是将眼睛放大.如图 1 所示,图 1(*a*)是放大前的眼睛,图 1(*b*)是放大后的眼睛,图 1(*b*)

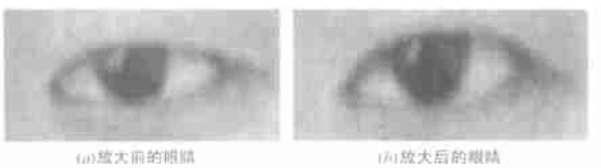

图 1　眼睛的放大

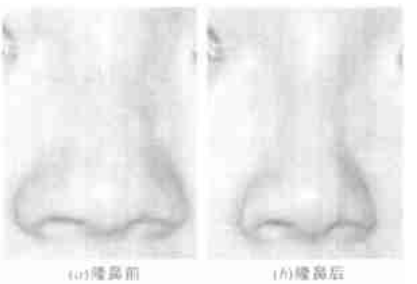

图 2　隆鼻

图象处理角度来看,是对原眼睛所在区域(非规则区域)的图象进行变形,即将其放大,实质上就是将原眼睛所在区域的图象象素按某种规则填充到眼睛放大后所在的图象区域(也是非规则区域).在这一过程中,填充是一个关键问题,它不仅要求填充前后两区域的图象非常相似,而且要求有一个较高的填充效率.再如隆鼻操作,如图 2 所示,图 2(*a*)是隆鼻前鼻子的图象,图 2(*b*)是隆鼻后鼻子的图象,隆鼻前的图象区域(非规则区域)被修改为隆鼻后的图象区域(也是非规则区域),这一处理也是图象变形,其过程也是一个填充过程.

2　传统的基于象素点填充的图象变形算法

正如前文所述,对变形后的图象区域进行填充是图象变形的关键问题,因此提高象素填充的效率和填充的效果尤为重要.目前的一些图象变形算法所采用的填充方法大都是基于象素点的填充.其大体思路概括如下:

首先介绍一下象素点的相对位置:如图 3 所示,对于图象区域中的任一个象素点 Q,过 Q 点沿垂直方向和水平方向各作一条垂线,分别与图象区域边界相交于点 a, b, c, d,记 $|aQ| = h1$, $|bQ| = h2$, $|cQ| = v1$, $|dQ| = v2$, $\alpha = h1/h2$, $\beta = v1/v2$,则象素点的相对位置由 (α,β) 决定.

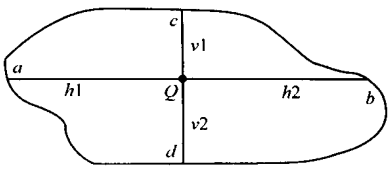

图 3

对变形后图象区域内的象素点进行填充的计算过程和依据是:对于变形后图象区域中的任意一个象素点 A,根据它的相对位置 (α,β),在变形前图象区域中找到与它具有相同相对位置的象素点 B 或具有相同相对位置且在同一邻域内的象素点集 B_i $(i = 0,1,2,3\ldots)$.若为后者,则需要根据其他算法从该点集中选择一个象素点(或者利用二次插值对 B_i 求加权平均值),将其值赋给象素点 A.

分析这种基于象素点的填充算法,存在两个主要问题:

(1) 当图象变形区域为非矩形区域时,象素填充的时间复杂度大.因为首先当变形后的图象区域为非矩形区域时,只能通过穷搜索找到每个象素点对应的四个边界点,才能计算出相对位置 (α,β);其次,若变形前的图象区域也为非矩形区域,则区域内任一行或任一列象素点的个数都可能不同.为了在变形前的图象区域找出具有相同的相对位置 (α,β) 的象素点或点集,首先对变形前的区域进行水平方向扫描,找出每一列中 β 值符合要求的象素点,对于这些象素点,再进行垂直方向的扫描,找出 α 值符合条件的象素点,从而可以求出变形前图象区域中具有与变形后图象区域指定象素点具有相同 α 值和 β 值的象素点集 M.同样地,也可以先进行垂直方向扫描,再进行水平方向扫描,但无论如何总是需要扫描两次.由此可以得出这种基于象素点填充的算法时间复杂度为 $O(m \times n \times p \times q)$,(其中 m、n 分别为变形前图象区域中象素的平均列数与行数,p、q 分别为变形后图象区域中象素的平均列数与行数).

(2) 当变形前的图象区域为非矩形时,有可能出现象素点集 M 中的象素点并非处在相同邻域,即在变形前的图象区域中有多个处在不同位置的象素点的相对位置相同且都等于 (α,β),如图 4 所示的 P、Q,它们有相同的 α 值和 β 值.

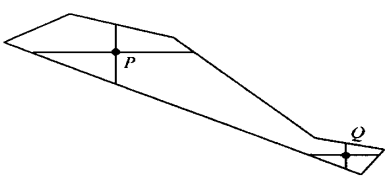

图 4

因此对于这种基于象素点进行填充的算法怎样取舍具有相同相对位置的象素点也就显得很重要了,当然对这些象素点进行判断和取舍也必然在一定程度上加大了时间复杂度.

虽然这种基于象素点填充的变形算法在对非矩形区域的图象实施变形时存在以上问题,但当变形前后的图象区域均为矩形时,时间复杂度可明显降低,这时变形后图象区域中每个象素点的相对位置 (α,β) 计算变得很简单,不必再进行穷

搜索,可以直接通过每个象素点的座标 $(x2, y2)$ 和矩形左上点座标 $(x1, y1)$ 及矩形的高 H 和宽 W 来计算,如图 5 (b) 所示;同时对于某一相对位置 (α,β),在变形前的图形区域中只可能有同一邻域内的象素点等于 (α,β),这样可以从根本上避免对处于不同位置具有相同相对位置的象素点进行取舍的问题. 下面推导这种图象变形中象素填充的有关计算式:

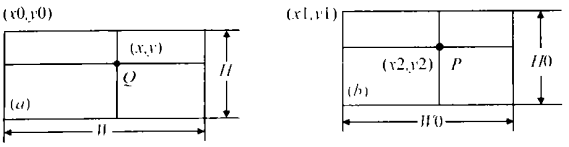

图 5 (a) 变形前的区域为矩形;(b) 变形后的区域也为矩形

如图 5 (b),对于变形后图象区域中的一个象素点 P,可通过式 (1) 求出它的相对位置 (α,β):

$$\begin{cases}\beta = (y2 - y1)/(H0 - (y2 - y1)) \\ \alpha = (x2 - x1)/(W0 - (x2 - x1))\end{cases} \quad (1)$$

如图 5 a,若变形前图象区域中某一个象素点 Q 与 P 具有相同的相对位置,由于变形前图象区域是矩形,则区域内任一行或任一列象素点的个数都相同,因此可以由式 (2) 计算出 Q 点的座标

$$\begin{cases}x = x0 + (1 - 1/(1+\alpha)) \times H \\ y = y0 + (1 - 1/(1+\beta)) \times H\end{cases} \quad (2)$$

对于这种变形区域是规则形状的图象变形,算法时间复杂度降为 $O(m \times n)$ (其中 m, n 分别为变形后图象矩形区域的宽度和高度). 为此,我们称这种基于象素点的矩形区域填充算法为标准矩形填充算法.

3 基于微分思想的图象变形算法

针对上述基于象素点填充算法中存在的问题,本文提出了一种基于微分思想的图象变形算法,其基本思想是:已知变形前图象区域为 $D1$、变形后图象区域为 $D2$ 和某一方向 P,那么,将区域 $D1$ 沿方向 P 划分为 n 个小矩形,每个矩形均有两条边的方向为 P,且方向为 P 的边的长度很小,根据微分思想,可以近似地认为区域 $D1$ 就是由这 n 个小矩形所构成;同样地,也将区域 $D2$ 划分为 n 个这样的小矩形. 区域 $D1$ 中的小矩形与区域 $D2$ 中的小矩形一一对应,得到 n 组小矩形. 然后,使用标准矩形填充算法,将每组中对应的小矩形进行快速填充从而实现图象变形. 根据该思想,本算法要解决如下几个问题:

(1) 对变形区域形状的要求:变形区域可以是任意区域,但需满足一定条件. 记图象变形区域为 D,既可指变形前的图象区域 $D1$,又可指变形后的图象区域 $D2$. 对于变形区域 D,本算法作出如下要求:区域是单连通域,且满足以下条件,即:至少存在一条直线段 L,使得区域 D 边界上的每个点向这条直线作出的垂线段 Lt, Lt 上每个点都不在区域 D 外. 区域与直线段的两个相距最远的交点就是直线段的两个端点. 如图 6 所示,其中图 6 (a) 为满足上述条件的单连通区域,而图 6 (b) 为不满足条件的单连通区域. 在以下分析中,设区域 $D1$ 对应的直线段为 $L1$,区域 $D2$ 对应的直线段为 $L2$.

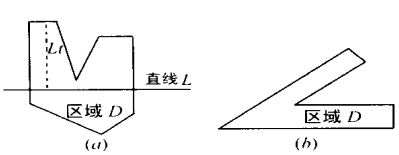

图 6

(2) 方向为 P 的直线段 $L1$、$L2$ 的选取是该算法实现中一个关键,对于符合上述条件的单连通区域 D,找出满足上述条件的一条直线段 L,如果符合这个条件的直线段有若干个,那么选取其方向能够简化计算的直线段. 通常选取其方向平行于 X 轴或平行于 Y 轴的直线段. 现设直线段 $L1$ 与区域 $D1$ 的两端的交点为 $L1a$、$L1b$;直线段 $L2$ 与区域 $D2$ 的两端的交点为 $L2a$、$L2b$.

(3) 矩形的划分规则:在变形前的图象区域 $D1$ 中,将直线段 $L1$ 划分为 n 段,每一段的长度为 Δl,这样总共有 $(n+1)$ 个划分点 T_0、T_1、…、T_n,经过每个划分点作垂直于直线段 $L1$ 的垂线,设这些垂线与区域边界的交点为 P_0、Q_0、P_1、Q_1、…、P_n、Q_n,如图 7 所示. 那么,当 Δl 足够小时,也即矩形 $P_iP_{(i+1)}Q_{(i+1)}Q_i$ 的边 $P_iP_{(i+1)}$ 足够短时,区域 $D1$ 可以近似地看成由 n 个矩形 $P_0P_1Q_1Q_0$、$P_1P_2Q_2Q_1$、…、$P_{n-1}P_nQ_nQ_{n-1}$ 所组成. 一般情况下,我们可设定 Δl 值为 2～3 个象素点.

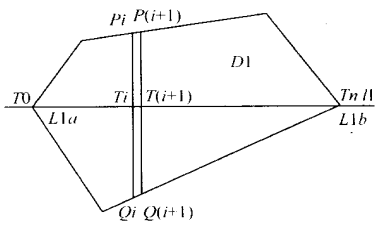

图 7

同样地,对于区域 $D2$ 也进行类似的划分:将直线段 $L2$ 等分为 n 段,这样总共有 $(n+1)$ 个划分点 T_0'、T_1'、…、T_n',经过每个划分点作垂直于直线段 $L2$ 的垂线,设这些垂线与边界的交点为 P_0'、Q_0'、P_1'、Q_1'、…、P_n'、Q_n',那么,区域 $D2$ 可以看作是由 n 个矩形 $P_0'P_1'Q_1'Q_0'$、$P_1'P_2'Q_2'Q_1'$、…、$P_{n-1}'P_n'Q_n'Q_{n-1}'$ 所组成的.

当 $L1$ 的长度 (记为 $|L1|$) 小于或等于 $L2$ 的长度 (记为 $|L2|$) 时,将 $|L1|$ 整除 Δl,商为 n,余数设为 k,在实际系统中,Δl 的值一般是 2～3 个象素点,而 $|L1|$ 一般都在 100 个象素点以上,所以 n 的值远大于 Δl,而 $k < \Delta l$. 为了保证区域 $D1$ 恰好划分为 n 个矩形,需要将余下的 k 列象素点分配给 n 个矩形中的 k 个矩形,分配规则如下:若存在某一个图象变形子区域对变形后的图象效果影响特别大,则这一子区域内的小矩形应分配到更多列的象素点,即把这 k 列象素点分配给这一子区域;反之,则把这 k 列象素点均匀地分配到整个图像变形区域中去. 如:在人脸整形与美容图象处理系统中眼睛部分的变形,因为变形区域中间部分 (眼球) 对其变形效果影响最大,所以应该将这 k 列象素点分配给线段 $L1$ 中心的 k 个小矩形,于是这些小矩形沿 P 方向的边宽度变为 $(\Delta l + 1)$,其他的 $(n-k)$ 个小矩形沿 P 方向的边宽度仍为 Δl,如图 8 所示. 然后,将 $|L2|$ 整除 n,得到 $\Delta l'$,显然 Δl 大于或等于 Δl;如果 Δl 是整数,那么每个矩形沿 P 方向的边长度为 $\Delta l'$;如果 $\Delta l'$ 不为整,则对 Δl 取整,得到新的 $\Delta l'$,那么每个矩形沿 P 方向

的初始长度可设为 $\Delta l'$,对于剩下的($|L2|-n*\Delta l'$)记为 k' 列象素,再依照前面提到的分配原则进行分配,将这剩下的 k' 列象素点,分配给线段中心的 k 个小矩形,这时 k 个小矩形的沿 P 方向的边的宽度变为($\Delta l'+1$),如图 9 所示.

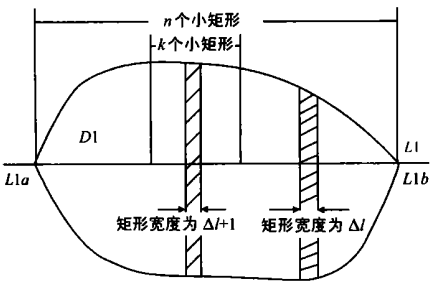

图 8　变形前的图象区域的矩形划分

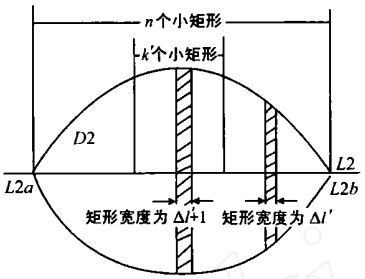

图 9　变形后的图象区域的矩形划分

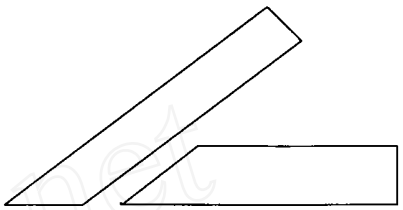

图 10

当$|L1|$ 大于$|L2|$ 时,可以采用与上面类似的方法进行矩形划分.

(4)将 $D1$ 中的 n 个小矩形与 $D2$ 中的 n 个小矩形一一对应起来,可以得到 n 组对应的小矩形,对于每组中的两个小矩形,采用标准矩形填充算法,即用式(1)和式(2)计算后进行快速填充.

(5)对于不满足(1)中所述条件的变形区域,可以将其分解为若干个小区域,使每个小区域都满足其条件,然后再使用本算法进行填充,实现图象变形. 如图 6(b),可以先分解为图 10 所示的两个子区域,然后再运用算法分别对两个区域进行变形处理.

4　算法的修正

虽然笔者提出的这种算法是基于微分思想的,但由于在实际的图象变形中处理的最小单位是象素点,因此不可能做到把划分的矩形宽度趋近于 0,而只能是 1～2 个象素点,这样将导致相邻两个矩形的高度会有 1～2 个象素点的差值,反映到最后的变形效果上就是在图象变形区域边界上存在锯齿状. 如图 11(a)所示. 为了解决这个问题,首先把包含了变形前图象区域的一个矩形范围放大 k ($k>2$)倍(因为是标准矩形,所以可以利用基于象素点填充算法来进行快速放大),然后利用基于微分思想的变形算法对放大图象区域进行变形,变形完成后,再把放大的图象缩小 $1/k$. 经过这一处理,本来存在的 1～2 个象素点的高度差就变成了 $1/k$～$2/k$ 的差值,基本消除了图象变形边缘的锯齿状. 如图 11(b)所示.

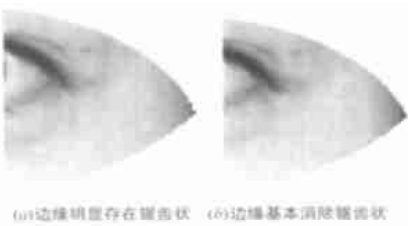

图 11　锯齿的消除

5　算法分析

在以下的分析中,m、n 分别为图象变形前区域的平均列数与行数,p、q 分别为图象变形后区域的平均列数与行数.

首先分析传统的基于象素点填充的图象变形算法. 变形后图象区域平均有 $m\times n$ 个象素点,当变形前图象区域为非规则形状时,对于变形后图象区域中的每一个象素点,都需要在变形前的图象区域中进行 $m\times n$ 次搜索,以在变形前的图象区域中找出具有相同相对位置(α,β)的一个象素点或一组象素点. 所以,当 m、n、p、q 中有任意一个量发生变化时,运算所花费的时间都会变化,时间复杂度为 $O(m\times n\times p\times q)$.

对于本文提出的基于微分思想的矩形填充算法,假设在变形后图象区域中沿着水平方向划分小矩形,设每个小矩形沿着水平方向的平均边长为 t,那么总共可以划分为(p/t)个小矩形,每个矩形的平均高度为 n,每个小矩形中象素点数目为($q\times t$),对每个象素点,可以利用矩形相对位置匹配索引的公式,快速求出变形前图象区域中对应小矩形中的对应的源象素点,而不需要象传统的基于象素点填充的变形算法那样对变形前的图象区域进行 $m\times n$ 次搜索. 所以其时间复杂度为 $O(n\times t\times m/t)$,即 $O(n\times m)$,它无关于变形前图象区域的复杂程度和面积大小,比基于象素点填充的变形算法降低了两个时间复杂度级别,同时也避免了具有相同相对位置的象素点取舍的问题,因此,本文提出的基于微分思想的图象变形方法比基于象素点填充的图象变形算法有明显的优势.

6　实验结果

为了证明本文所提出的基于微分思想的图象变形算法有良好的变形效果和较高的效率,笔者设计了一个模拟程序(程序运行硬件环境:P Ⅲ600MHz,64M RAM),通过与基于象素点填充的图象变形算法相比较来进行验证. 实验分为三组,第一、二组实验用的原始图象如图 12 所示,该图象区域象素的平均列数和平均行数分别为 252、91,第三组实验用的原始图象如图 13 所示,其象素平均列数和平均行数为前者的两倍,分别为 504 和 182.

第一组实验:采用基于象素点填充的图象变形算法和本

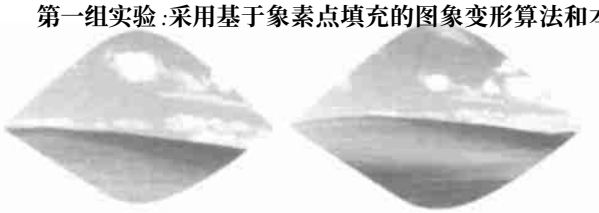

图 12　第一、二组实验用的原始图象

图 13　第三组实验用的原始图象

文所提出的图象变形算法并针对同一个变形后的图象区域进行实验,其变形效果分别如图 14、15 所示,实验数据列于表 1 的第一行,可以看到,本文提出的变形算法,其填充效果和所花费时间与基于象素点填充的变形算法相比有明显提高.

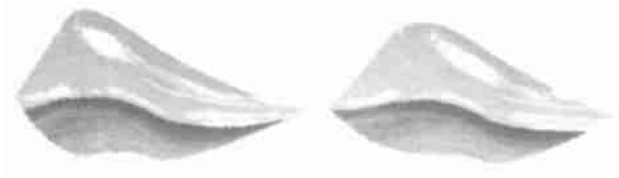

图 14 基于象素点填充算法实现的变形效果

图 15 本文所提出的算法实现的变形效果

第二组实验:改变变形后图象区域的象素列数,使其为前者的两倍,结果两种算法所耗费的时间约为第一组实验时的两倍.这是因为这两种算法的时间消耗均与图象变形后的区域相关,当变形后区域发生改变时,所消耗的时间也会随之改变.图 16、图 17 分别为第二组实验采用上述两种算法所得到的图象变形效果,其实验数据列于表 1 的第二行.

图 16 基于象素点填充算法实现的变形效果图

图 17 本文所提出的算法实现的变形效果

第三组实验:同时改变原始图象区域和变形后的图象区域的大小,使象素的平均行数和平均列数都是第一组实验的两倍,于是,用基于象素填充的变形算法所耗费的时间将为第一组实验的 16 倍,这是由其时间复杂为 $O(N^4)$ 所决定的,而采用基于本文算法所耗费的时间只是第一组实验的 4 倍,图 18、图 19 分别为第三组实验采用上述两种算法所得到的图象变形效果,其实验数据列于表 1 的第三行.

图 18 基于象素点填充算法实现的变形效果

图 19 本文所提出的算法实现的变形效果

表 1

组号	m	n	p	q	基于象素点填充的变形算法所耗费的时间(ms)	本文所提出的变形算法所耗费的时间(ms)
1	252	91	267	78	3469	990
2	252	91	534	78	7021	1996
3	504	182	534	156	55504	4016

(表中 m、n 分别为变形前图象区域的象素平均列数与行数,p、q 分别为变形后图象区域的象素平均列数与行数).

7 结束语

本文介绍的基于微分思想的图象变形算法,它的核心思想就是问题的逐步细化,对于一个复杂的变形区域,首先将其分解为若干较简单的变形区域,这些较简单的区域再划分为很多小矩形,之后再对对应的矩形进行填充,它的效率明显高于基于象素填充的算法,同时变形效果好,易于实现,可适用于任意复杂的变形区域.

参考文献:

[1] 周秉锋. 一个基于线性变换的数字图像自由拉伸算法 [J]. 北京大学学报(自然科学版). 1997,33(6):770 - 775.

[2] 何国辉. 一种图象变形映射算法的研究. 中国图象图形学报 [J]. 1998,3(5):371 - 374.

[3] James D Foley, Andries van Dam, Steven K Feiner, et al. Computer Graphics: Principles and Practice (Second Edition) [M]. New York: Addison-Wesley Publishing Company Inc, 1993. 820 - 832:92 - 99.

[4] Catmull E. 3-D Transformations of images in scan line order [J]. Computer Graphics, 1980, 14(3):279 - 285.

[5] Smith A R. Planar2-pass texture mapping and warping [J]. Computer Graphics, 1987, 21(4):263 - 272.

[6] Reeves WT. Particle systems: A technique for modeling a class of fuzzy objects [J]. Computer Graphics, 1982, 17(3):359 - 376.

[7] Rosenfeld M. Special effects production with computer graphics and video techniques [J]. Computer Graphics, 1987, 21(4):197 - 206.

[8] Oka M K, Akio O, Yoshitaka K, Takashi T. Real-time manipulation of texture-mapped surfaces [J]. Computer Graphics, 1987, 21(4):181 - 188.

[9] Beier T, Neely S. Feature-based image metamorphosis [J]. Computer Graphics, 1992, 26(2):35 - 42.

[10] Grimson W. From Images to Surfaces [M]. Boston: MIT Press, 1981.

作者简介:

邹北骥 男,1961 年生,江西南昌人,清华大学计算机科学系在职博士后,湖南大学计算机与通信学院教授,主要研究领域:计算机图形学、图象处理与多媒体技术. Email: bjzou@vip.163.com

周浩宇 男,1979 年生于湖南长沙,硕士研究生,研究方向为计算机图形图象处理.

吕格峰 男,1980 年生于湖南长沙,硕士研究生,研究方向为计算机图形图象处理.

孙家广 男,1946 年生于江苏镇江,中国工程院院士,清华大学教授,主要研究领域:计算机图形学,CAD 技术,软件工程等.