(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103440633 A
(43)申请公布日 2013.12.11

(21)申请号 201310404195.1

(22)申请日 2013.09.06

(71)申请人 厦门美图网科技有限公司
地址 361008 福建省厦门市软件园二期望海路2号C栋4层

(72)发明人 张伟 傅松林 李志阳 张长定

(51)Int.Cl.
G06T 5/00(2006.01)

权利要求书3页 说明书8页

(54)发明名称

一种数字图像自动祛除斑点的方法

(57)摘要

本发明公开了一种数字图像自动祛除斑点的方法，其特征在于：通过对图像A依次进行灰度化、对比度增强、梯度极大值查找、皮肤排除、孤立点消除、高斯模糊、阈值处理、区域表求和得到结果D；最后根据结果D与梯度极大值查找的结果对图像A里的斑点进行泊松方程处理，得到自动祛痘祛斑祛痣的最终效果。本方案完全省却了数字图像祛斑操作中手动的动作，整个过程可自动完成，无需手动标定和选择斑点，节省了操作时间和步骤。

CN 103440633 A

1. 一种数字图像自动祛除斑点的方法，其特征在于：它包括以下步骤：

1) 接收一数字图像 A，对图像 A 进行灰度化处理，得到灰度图像 B；该处理采用下列两式中的一个：

Gray = 0.299*Red+0.587*Green+0.114*Blue；

Gray = (Red*306+Green*601+Blue*117+512)/1024；

其中，Gray 为该灰度图像 B 各像素点的灰度值，Red、Green、Blue 分别为图像 A 各像素点红、绿、蓝三个通道的颜色值；

2) 对灰度图像 B 进行对比度增强处理，公式为：

nResult=nColor+(nColor-128)*(1.0+Contrast)/255

其中，nResult 表示对比度增强后的灰度值，nColor 表示要进行对比度增强的灰度值，Contrast 为对比度增强的强度，范围 [0.0,1.0]；

3) 对灰度图像 B 进行梯度极大值查找，步骤如下：

首先用$\begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}$与灰度图像 B 里的逐个像素点进行卷积运算，然后将计算后的结果进行统计，并将统计结果保存到一个大小为 256 的数组里，根据统计结果的计算得到阈值 K，根据阈值 K 对卷积运算后的结果进行阈值化处理，大于等于阈值 K 的设置为 255，小于阈值 K 的设置为 0，最终得到梯度结果 C；

4) 对梯度结果 C 先后进行皮肤排除与孤立点消除处理；

图片 A 的宽为 w，图片的高为 h；皮肤识别后的数据为 pEdgeTable；i 表示当前像素点的行数，j 表示当前的列数；

皮肤排除的步骤：判断当前的像素点是否是皮肤；如果是皮肤的话，则将其所对应的梯度结果 C 上的值设置为 255；

该孤立点消除处理步骤为：

a. 建立边缘个数统计表，过程如下：建立一个数组 pSumTable，并都初始化为 0，数组的大小为 (w+1)*(h+1)；接着从第 2 行开始进行计算，并且每行计算时都是从第 2 列开始的（即第一行和第一列的数据都不用参与这个计算），每行从预设 passSum=0 开始赋值；

passSum=passSum+(pEdgeTable[j*(w+1)+i] & 0×01)；

pSumTable[j*(w+1)+i]=pSumTable[(j-1)*(w+1)+i]+passSum；

b. 剔除不连续的点：设定一个大小为 3×3 的窗口，首先判断 pEdgeTable 当前像素点的值是否等于 0，如果等于 0，则继续遍历下一个像素点，否则执行下面操作：

计算 pEdgeTable 当前像素点为中心的所述窗口内其他像素的值；假设上为 top，下为 bottom，左为 left，右为 right，则

top=max(0, j-1)；

bottom=min(h, j+2)；

right=min(w, i+2)；

left=max(0, i-1)；

假设上一个搜索窗口内的 Edge 个数为 preWinEdgeCount；当前搜索窗口内的 Edge 个数为 curWinEdgeCount；

先计算上一个搜索窗口内的 Edge 个数：

p1=(w+1)*top+left；

p2=(w+1)*bottom+left；

offset=right-left；

preWinEdgeCount=pSumTable[p2+offset]+pSumTable[p1]-pSumTable[p2]-pSumTable[p1+offset]；

接下来预设 r=2，并做循环，直到 r 大于 5 则跳出循环，以下为循环的步骤：

计算当前搜索窗口内的 Edge 个数：

top=max(0, j-r)；

bottom=min(h, j+r+1)；

right=min(w, i+r+1)；

left=max(0, i-r)；

p1=(w+1)*top+left；

p2=(w+1)*bottom+left；

offset=right-left；

curWinEdgeCount=pSumTable[p2+offset]+pSumTable[p1]-pSumTable[p2]-pSumTable[p1+offset]；

判断 curWinEdgeCount-preWinEdgeCount 是否小于 2，如果小于的话，则认定为不连续的边缘，归为奇异点，将 pEdgeTable 在该像素点的值设为 0；否则将 preWinEdgeCount=curWinEdgeCount；

5) 对梯度结果 C 进行高斯模糊处理，并做阈值处理；

高斯模糊是用正态分布计算图像中每个像素的变换：

$$G(u,v)=\frac{1}{2\pi\sigma^2}e^{-(u^2+v^2)/(2\sigma^2)}$$

其中 r 是模糊半径，σ 是正态分布的标准偏差，u 是原像素点在 x 轴上的位置偏移值。V 是原像素点在 y 轴上的位置偏移值；

阈值的公式为：如果大于等于阈值 K2 的话，则值设置为 255，否则设置为 0。这边 K2 的范围 [0,128]；

6) 根据灰度图像 B 进行区域表求和得到结果 D;

结果 D 的数组大小为图片的宽 w 与图片的高 h 的乘积，假设为数组

GraySumArea[w][h]，先预设数组的所有值为 0；

接着计算数组第一行的值，公式为：

GraySumArea[i][0]=GraySumArea[i-1][0]+grayi；

其中 i 从 1 开始直到 w 为止；grayi 为坐标 (i,0) 上像素点的灰度值；

接着计算数组第一列的值，公式为：

GraySumArea[0][j]=GraySumArea[0][j-1]+grayj；

其中 j 从 1 开始到 h 为止；grayj 为坐标为 (0, j) 上像素点的灰度值；

接着计算数组剩余的值，公式为：

GraySumArea[i][j]=grayij+GraySumArea[i][j-1]+GraySumArea[i-1]

[j]-GraySumArea[i-1][j-1]；

其中 i 从 1 开始到 w 为止；j 从 1 开始到 h 为止；grayij 为坐标为 (i，j) 上像素点的灰度值；

7) 根据结果 D 与梯度结果 C 对图像 A 里的斑点进行泊松方程处理，得到自动祛痘祛斑祛痣的最终效果；首先利用结果 D 进行判断每个像素点是否为斑点；判断规则为判断该像素点在结果 D 上的值与周围值的差值，判断差值是否小于 R 值，，如果是的话，则不是斑点，继续遍历下一个像素；否则继续判断该像素点在梯度结果 C 上的值，如果大于 T 值，则不是斑点；否则是斑点；其中，R 的范围为 [64，220]T 的为 [64，192]；

当该像素点为斑点时，则根据泊松方程 $\Delta \Phi = f$ 对该点进行肤色值的进行融合计算；经过泊松方程处理后得到该像素点新的颜色值，得到自动祛痘祛斑祛痣的最终效果。

2. 根据权利要求 1 所述一种数字图像自动祛除斑点的方法，其特征在于：所述步骤 2) 中，所述对比度增强的强度 Contrast 为 0.15。

3. 根据权利要求 1 所述一种数字图像自动祛除斑点的方法，其特征在于：所述步骤 5) 中的阈值 K2 设为 20。

4. 根据权利要求 1 所述一种数字图像自动祛除斑点的方法，其特征在于：所述步骤 7) 中 R 为 200，T 为 128。

一种数字图像自动祛除斑点的方法

技术领域

[0001] 本发明涉及一种数字图像的处理方法，具体涉及一种自动祛除数字图像中斑点的图像处理方法。

背景技术

[0002] 随着数字便携设备的普及，各种设备包括手机、平板电脑中也随之大量地应用了数字摄像装置，使用户拍摄照片的门槛大大降低，各种和生活有关的拍摄可以通过简单的操作完成，具有十分便利的特点。特别是对个人的自拍，已成为我们日常生活中常见的行为，甚至已经作为一种社交的技术手段。这类针对人像，特别是脸谱的拍摄，通常需要自拍后对图像进行调整和美化。这类操作比较常见的一种就是将图像里一些不想展现的元素祛除（例如痘、斑、痣等）。

[0003] 虽然现在的图像处理软件很多都可以满足用户的大部分需求，具备这类祛斑的功能，但是现有的祛斑操作仍然步骤繁多，不够便利和快速。如此，如何尽量少地让用户操作，如何更智能、快速地祛除数字图像的暗斑，成为一个亟待解决的问题。

发明内容

[0004] 针对现有数字图像处理方法缺少快速、自动的处理的解决办法，本发明提出一种数字图像自动祛斑的方法，其方案如下：

[0005] 一种数字图像自动祛除斑点的方法，它包括以下步骤：

[0006] 1) 接收一数字图像 A，对图像 A 进行灰度化处理，得到灰度图像 B；该处理采用下列两式中的一个：

[0007] Gray=0.299*Red+0.587*Green+0.114*Blue；

[0008] Gray=(Red*306+Green*601+Blue*117+512) / 1024；

[0009] 其中，Gray 为该灰度图像 B 各像素点的灰度值，Red、Green、Blue 分别为图像 A 各像素点红、绿、蓝三个通道的颜色值；

[0010] 2) 对灰度图像 B 进行对比度增强处理，公式为：

[0011] nResult=nColor+(nColor-128)*(1.0+Contrast) / 255

[0012] 其中，nResult 表示对比度增强后的灰度值，nColor 表示要进行对比度增强的灰度值，Contrast 为对比度增强的强度，范围 [0.0,1.0]；

[0013] 3) 对灰度图像 B 进行梯度极大值查找，步骤如下：

[0014] 首先用 $\begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}$ 与灰度图像 B 里的逐个像素点进行卷积运算，然后将计算后的结果进行统计，并将统计结果保存到一个大小为 256 的数组里，根据统计结果的计算得到阈值 K，根据阈值 K 对卷积运算后的结果进行阈值化处理，大于等于阈值 K 的设置为 255，小于阈值 K 的设置为 0，最终得到梯度结果 C；

[0015] 4) 对梯度结果 C 先后进行皮肤排除与孤立点消除处理；

[0016] 图片 A 的宽为 w，图片的高为 h；皮肤识别后的数据为 pEdgeTable；i 表示当前像素点的行数，j 表示当前的列数；

[0017] 皮肤排除的步骤：判断当前的像素点是否是皮肤；如果是皮肤的话，则将其所对应的梯度结果 C 上的值设置为 255；该皮肤判断的方法可采用多种现有技术，如下述文献提及的方法 M. J. Jones and J. M. Rehg，“Statistical Color Models with Application to Skin Detection”Proc. CVPR. 1999.

[0018] 该孤立点消除处理步骤为：

[0019] a. 建立边缘个数统计表，过程如下：建立一个数组 pSumTable，并都初始化为 0，数组的大小为 (w+1)*(h+1)；接着从第 2 行开始进行计算，并且每行计算时都是从第 2 列开始的（即第一行和第一列的数据都不用参与这个计算），每行从预设 passSum=0 开始赋值；

[0020] passSum=passSum+(pEdgeTable[j*(w+1)+i]&0×01)；

[0021] pSumTable[j*(w+1)+i]=pSumTable[(j-1)*(w+1)+i]+passSum；

[0022] b. 剔除不连续的点：设定一个大小为 3×3 的窗口，首先判断 pEdgeTable 当前像素点的值是否等于 0，如果等于 0，则继续遍历下一个像素点，否则执行下面操作：

[0023] 计算 pEdgeTable 当前像素点为中心的所述窗口内其他像素的值；假设上为 top，下为 bottom，左为 left，右为 right，则

[0024] top=max(0, j-1)；

[0025] bottom=min(h, j+2)；

[0026] right=min(w, i+2)；

[0027] left=max(0, i-1)；

[0028] 假设上一个搜索窗口内的 Edge 个数为 preWinEdgeCount；当前搜索窗口内的 Edge 个数为 curWinEdgeCount；

[0029] 先计算上一个搜索窗口内的 Edge 个数：

[0030] p1=(w+1)*top+left；

[0031] p2=(w+1)*bottom+left；

[0032] offset=right-left；

[0033] preWinEdgeCount=pSumTable[p2+offset]+pSumTable[p1]-pSumTable[p2]-pSumTable[p1+offset]；

[0034] 接下来预设 r=2，并做循环，直到 r 大于 5 则跳出循环，以下为循环的步骤：

[0035] 计算当前搜索窗口内的 Edge 个数：

[0036] top=max(0, j-r)；

[0037] bottom=min(h, j+r+1)；

[0038] right=min(w, i+r+1)；

[0039] left=max(0, i-r)；

[0040] p1=(w+1)*top+left；

[0041] p2=(w+1)*bottom+left；

[0042] offset=right-left；

[0043] curWinEdgeCount=pSumTable[p2+offset]+pSumTable[p1]-pSumTable[p2]-pSum

Table[p1+offset]；

[0044] 判断 curWinEdgeCount-preWinEdgeCount 是否小于 2，如果小于的话，则认定为不连续的边缘，归为奇异点，将 pEdgeTable 在该像素点的值设为 0；否则将 preWinEdgeCount =curWinEdgeCount；

[0045] 6) 对梯度结果 C 进行高斯模糊处理，并做阈值处理；

[0046] 高斯模糊是用正态分布计算图像中每个像素的变换：

[0047] $$G(u,v)=\frac{1}{2\pi\sigma^2}e^{-(u^2+v^2)/(2\sigma^2)}$$

[0048] 其中 r 是模糊半径，σ 是正态分布的标准偏差，u 是原像素点在 x 轴上的位置偏移值。V 是原像素点在 y 轴上的位置偏移值；

[0049] 阈值的公式为：如果大于等于阈值 K2 的话，则值设置为 255，否则设置为 0。这边 K2 的范围 [0,128]；

[0050] 6) 根据灰度图像 B 进行区域表求和得到结果 D；

[0051] 结果 D 的数组大小为图片的宽 w 与图片的高 h 的乘积，假设为数组

[0052] GraySumArea[w][h]，先预设数组的所有值为 0；

[0053] 接着计算数组第一行的值，公式为：

[0054] GraySumArea[i][0]=GraySumArea[i-1][0]+grayi；

[0055] 其中 i 从 1 开始直到 w 为止；grayi 为坐标 (i,0) 上像素点的灰度值；

[0056] 接着计算数组第一列的值，公式为：

[0057] GraySumArea[0][j]=GraySumArea[0][j-1]+grayj；

[0058] 其中 j 从 1 开始到 h 为止；grayj 为坐标为 (0，j) 上像素点的灰度值；

[0059] 接着计算数组剩余的值，公式为：

[0060] GraySumArea[i][j] ＝ grayij+GraySumArea[i][j-1]+GraySumArea[i-1][j]-GraySumArea[i-1][j-1]；

[0061] 其中 i 从 1 开始到 w 为止；j 从 1 开始到 h 为止；grayij 为坐标为 (i，j) 上像素点的灰度值；

[0062] 7) 根据结果 D 与梯度结果 C 对图像 A 里的斑点进行泊松方程处理，得到自动祛痘祛斑祛痣的最终效果；首先利用结果 D 进行判断每个像素点是否为斑点；判断规则为判断该像素点在结果 D 上的值与周围值的差值，判断差值是否小于 R 值，，如果是的话，则不是斑点，继续遍历下一个像素；否则继续判断该像素点在梯度结果 C 上的值，如果大于 T 值，则不是斑点；否则是斑点；其中，R 的范围为 [64,220])T 的为 [64,192]；

[0063] 当该像素点为斑点时，则根据泊松方程 Δ Φ=f 对该点进行肤色值的进行融合计算；经过泊松方程处理后得到该像素点新的颜色值，得到自动祛痘祛斑祛痣的最终效果。

[0064] 作为本方案的优选者，可以有如下改进：所述步骤 2) 中，所述对比度增强的强度 Contrast 为 0.15。所述步骤 5) 中的阈值 K2 设为 20。所述步骤 7) 中 R 为 200，T 为 128。

[0065] 本方案带来的有益效果有：

[0066] 1. 完全省却了数字图像祛斑操作中手动的动作，整个过程可自动完成，无需手动标定和选择斑点，节省了操作时间和步骤；

[0067] 2. 在触控屏数字设备中避免了不准确的手势触控带来失效模式，可以仅适用菜单

/ 按钮进行快速操作，不会产生失误。

具体实施方式

[0068] 本实施例一种数字图像自动祛除斑点的方法，它包括以下步骤：

[0069] 1) 接收一人脸图像 A 数字图像 A，对图像 A 进行灰度化处理，得到灰度图像 B；该处理采用下列两式中的一个：

[0070] Gray=0.299*Red+0.587*Green+0.114*Blue；

[0071] Gray=(Red*306+Green*601+Blue*117+512)/1024；

[0072] 其中，Gray 为该灰度图像 B 各像素点的灰度值，Red、Green、Blue 分别为图像 A 各像素点红、绿、蓝三个通道的颜色值；

[0073] 2) 对灰度图像 B 进行对比度增强处理，公式为：

[0074] nResult=nColor+(nColor-128)*(1.0+Contrast)/255

[0075] 其中，nResult 表示对比度增强后的灰度值，nColor 表示要进行对比度增强的灰度值，Contrast 为对比度增强的强度，取 0.15；

[0076] 3) 对灰度图像 B 进行梯度极大值查找，步骤如下：

[0077] 首先用 $\begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}$ 与灰度图像 B 里的逐个像素点进行卷积运算，然后将计算后的结果进行统计，并将统计结果保存到一个大小为 256 的数组里，根据统计结果的计算得到阈值 K，根据阈值 K 对卷积运算后的结果进行阈值化处理，大于等于阈值 K 的设置为 255，小于阈值 K 的设置为 0，最终得到梯度结果 C；

[0078] 4) 对梯度结果 C 先后进行皮肤排除与孤立点消除处理；

[0079] 图片 A 的宽为 w = 480，图片的高为 h = 640；皮肤识别后的数据为 pEdgeTable；i 表示当前像素点的行数，j 表示当前的列数；

[0080] 皮肤排除的步骤：判断当前的像素点是否是皮肤；如果是皮肤的话，则将其所对应的梯度结果 C 上的值设置为 255；

[0081] 本方案采用基于肤色模型的皮肤识别判断方法，步骤如下：

[0082] (1) 对图像进行人脸识别，获取人脸区域；

[0083] (2) 对步骤 (1) 获取的人脸区域进行均值计算，获取平均肤色；

[0084] (2.1) 初始化原始皮肤模型；

[0085] (2.1.1) 创建肤色模型，大小为 256*256；

[0086] (2.1.2) 依次对肤色模型进行赋值，具体伪代码如下；

[0087] 预设临时变量 AlphaValue、nMax、i、j 为整数类型。

[0088] 肤色模型变量为 SkinModel[256][256]

[0089] For(i = 0；i < 256；i++)

[0090] {

[0091] 判断 i 是否大于 128，如果大于 128，则 AlphaValue 为 255，否则为 i*2；

[0092] 计算获得 nMax 的值，计算公式为 nMax = min(256，AlphaValue*2)；

[0093] For(j = 0；j < nMax；j++)

[0094] {

[0095] 计算对应位置的肤色模型的值，计算公式为 SkinModel[i][j] = AlphaValue-(j / 2)；

[0096] }

[0097] For(j = nMax. j < 256 ;j++)

[0098] {

[0099] 初始对应位置的肤色模型的值为 0；

[0100] }

[0101] }；

[0102] (2.2) 计算整个图像的颜色均值，作为初始皮肤的阈值；

[0103] (2.2.1) 遍历整个图像的像素点，将红色通道、绿色通道、蓝色通道的颜色值累加，得到颜色累加值；

[0104] (2.2.2) 将颜色累加值除以像素点的总数，得到红色通道、绿色通道、蓝色通道的均值，作为初始皮肤的阈值。

[0105] (2.3) 根据步骤 (2.2) 获取得的初始皮肤的阈值计算人脸区域的平均肤色。

[0106] (2.3.1) 根据如下公式计算平均肤色的黑白值：

[0107] GRAY1 = 0.299*RED+0.587*GREEN+0.114*BLUE

[0108] 其中，GRAY1 为灰度图的当前像素点的灰度值；RED、GREEN、BLUE 分别为图像的当前像素点的红、绿、蓝通道的颜色值；

[0109] (2.3.2) 将步骤 (2.3.1) 中的黑白值作为阈值，用来排除人脸区域非皮肤的部分；

[0110] 并依次遍历人脸区域里的像素点的颜色值，根据如下公式获得平均肤色：

[0111] skin = SkinModel[red][blue]；

[0112] 其中，skin 为经过皮肤模型的颜色映射后的皮肤值；SkinModel 为步骤 (2.1) 的初始化原始皮肤模型；red 为红色通道的颜色值；blue 为蓝色通道的颜色值。

[0113] (3) 根据步骤 (2) 获取的平均肤色计算当前图像的肤色概率映射表；

[0114] (3.1) 创建肤色概率映射表，大小为 256*256；

[0115] (3.2) 依次对肤色概率映射表进行赋值，具体伪代码如下；

[0116] 预设临时变量 i、j、SkinRed_Left、AlphaValue、Offset、TempAlphaValue、OffsetJ 为整数类型；

[0117] 肤色概率映射表的变量为 SkinProbability[256][256]；

[0118] SkinRed 为步骤 (2.2.2) 计算得到的红色通道的均值；SkinBlue 为步骤 (2.2.2) 计算得到的蓝色通道的均值；

[0119] 预设 SkinRed_Left 的值，计算公式为：SkinRed_Left = SkinRed-128；

[0120] For(i = 0 ;i < 256 ;i++)

[0121] {

[0122] 计算 Offset 的值，公式为 Offset = max(0, min(255, i-SkinRed_Left))；

[0123] 判断 Offset 的值是否小于 128，如果小于的，话则 AlphaValue = Offset*2；如果大于等于 128 的话，则 AlphaValue = 255；

[0124] For(j = 0 ;j < 256 ;j++)

[0125] {

[0126] 计算 OffsetJ 的值,公式为 OffsetJ = max(0, j-SkinBlue) ;

[0127] 计 算 TempAlphaValue 的 值, 公 式 为 TempAlphaValue = max(AlphaValue-(OffsetJ*2),0) ;

[0128] 判断 TempAlphaValue 的值。如果大于 160 的话,则 SkinProbability[i][j] 的值为 255 ;

[0129] 如 果 小 于 90 的 话, 则 SkinProbability[i][j] 的 值 为 0 ; 否 则 SkinProbability[i][j] 的值为 TempAlphaValue+30 ;

[0130] }

[0131] } ;

[0132] (4) 根据步骤 (3) 获取的肤色概率映射表对当前图像进行肤色识别,并获得当前图像的肤色概率的结果图,通过如下公式进行实现 :

[0133] skinColor = SkinProbability[red][blue]

[0134] 其中, skinColor 为结果图的肤色概率值 ;SkinProbability 为肤色概率映射表 ; red 为像素点的红色通道的颜色值 ;blue 为像素点的蓝色通道的颜色值。

[0135] 该孤立点消除处理步骤为 :

[0136] a. 建立边缘个数统计表,过程如下 :建立一个数组 pSumTable,并都初始化为 0,数组的大小为 (w+1)*(h+1) ;接着从第 2 行开始进行计算,并且每行计算时都是从第 2 列开始的 (即第一行和第一列的数据都不用参与这个计算),每行从预设 passSum = 0 开始赋值 ;

[0137] passSum = passSum+(pEdgeTable[j*(w+1)+i] & 0×01) ;

[0138] pSumTable[j*(w+1)+i] = pSumTable[(j-1)*(w+1)+i]+passSum ;

[0139] b. 剔除不连续的点 :设定一个大小为 3×3 的窗口,首先判断 pEdgeTable 当前像素点的值是否等于 0,如果等于 0,则继续遍历下一个像素点,否则执行下面操作 :

[0140] 计算 pEdgeTable 当前像素点为中心的所述窗口内其他像素的值 ;假设上为 top,下为 bottom,左为 left,右为 right,则

[0141] top = max(0, j-1) ;

[0142] bottom = min(h, j+2) ;

[0143] right = min(w, i+2) ;

[0144] left = max(0, i-1) ;

[0145] 假设上一个搜索窗口内的 Edge 个数为 preWinEdgeCount ;当前搜索窗口内的 Edge 个数为 curWinEdgeCount ;

[0146] 先计算上一个搜索窗口内的 Edge 个数 :

[0147] p1=(w+1)*top+left ;

[0148] p2=(w+1)*bottom+left ;

[0149] offset=right-left ;

[0150] preWinEdgeCount=pSumTable[p2+offset]+pSumTable[p1]-pSumTable[p2]-pSumTable[p1+offset] ;

[0151] 接下来预设 r = 2,并做循环,直到 r 大于 5 则跳出循环,以下为循环的步骤 :

[0152] 计算当前搜索窗口内的 Edge 个数 :

[0153] top=max(0, j-r) ;

[0154] bottom=m in(h, j+r+1) ;

[0155] right=min(w, i+r+1) ;

[0156] left=max(0, i-r) ;

[0157] p1=(w+1)*top+left ;

[0158] p2=(w+1)*bottom+l eft ;

[0159] offset=right-left ;

[0160] curWinEdgeCount=pSumTable[p2+offset]+pSumTable[p1]-pSumTable[p2]-pSumTable[p1+offset] ;

[0161] 判断 curWinEdgeCount-preWinEdgeCount 是否小于 2,如果小于的话,则认定为不连续的边缘,归为奇异点,将 pEdgeTable 在该像素点的值设为 0 ;否则将 preWinEdgeCount =curWinEdgeCount ;

[0162] 5) 对梯度结果 C 进行高斯模糊处理,并做阈值处理 ;

[0163] 高斯模糊是用正态分布计算图像中每个像素的变换 :

[0164] $$G(u,v)=\frac{1}{2\pi\sigma^2}e^{-(u^2+v^2)/(2\sigma^2)}$$

[0165] 其中 r 是模糊半径, σ 是正态分布的标准偏差,u 是原像素点在 x 轴上的位置偏移值。V 是原像素点在 y 轴上的位置偏移值 ;

[0166] 阈值的公式为 :如果大于等于阈值 K2 的话,则值设置为 255,否则设置为 0, ;2 取 20 ;

[0167] 6) 根据灰度图像 B 进行区域表求和得到结果 D ;

[0168] 结果 D 的数组大小为图片的宽 w 与图片的高 h 的乘积,假设为数组

[0169] GraySumArea[w][h],先预设数组的所有值为 0 ;

[0170] 接着计算数组第一行的值,公式为 :

[0171] GraySumArea[i][0]=GraySumArea[i-1][0]+grayi ;

[0172] 其中 i 从 1 开始直到 w 为止 ;gray i 为坐标 (i,0) 上像素点的灰度值 ;

[0173] 接着计算数组第一列的值,公式为 :

[0174] GraySumArea[0][j]=GraySumArea[0][j-1]+grayj ;

[0175] 其中 j 从 1 开始到 h 为止 ;grayj 为坐标为 (0, j) 上像素点的灰度值 ;

[0176] 接着计算数组剩余的值,公式为 :

[0177] GraySumArea[i][j]=grayij+GraySumArea[i][j-1]+GraySumArea[i-1][j]-GraySumArea[i-1][j-1] ;

[0178] 其中 i 从 1 开始到 w 为止 ;j 从 1 开始到 h 为止 ;grayij 为坐标为 (i, j) 上像素点的灰度值 ;

[0179] 7) 根据结果 D 与梯度结果 C 对图像 A 里的斑点进行泊松方程处理,得到自动祛痘祛斑祛痣的最终效果 ;首先利用结果 D 进行判断每个像素点是否为斑点 ;判断规则为判断该像素点在结果 D 上的值与周围值的差值,判断差值是否小于 R 值,如果是的话,则不是斑点,继续遍历下一个像素 ;否则继续判断该像素点在梯度结果 C 上的值,如果大于 T 值,则不是斑点 ;否则是斑点 ;其中,R 为 200,T 为 128 ;

[0180] 当该像素点为斑点时，则根据泊松方程 $\Delta \Phi = f$ 对该点进行肤色值的进行融合计算；经过泊松方程处理后得到该像素点新的颜色值，得到自动祛痘祛斑祛痣的最终效果，在不影响头像肤色、毛发等细节的前提下，肉眼已经不可见其上的痣点。