



(12)发明专利

(10)授权公告号 CN 107464532 B

(45)授权公告日 2019.09.27

(21)申请号 201710644421.1

(22)申请日 2017.07.31

(65)同一申请的已公布的文献号
申请公布号 CN 107464532 A

(43)申请公布日 2017.12.12

(73)专利权人 天津大学
地址 300192 天津市南开区卫津路92号

(72)发明人 张涛 巫红英 赵鑫

(74)专利代理机构 天津市北洋有限责任专利代
理事务所 12201

代理人 杜文茹

(51)Int.Cl.
G09G 3/34(2006.01)

(56)对比文件

- CN 102930833 A, 2013.02.13, 全文.
- CN 102903340 A, 2013.01.30, 全文.
- JP 特开2008-71603 A, 2008.03.27, 全文.
- CN 103792704 A, 2014.05.14, 全文.
- CN 103943071 A, 2014.07.23, 全文.

审查员 宋澄

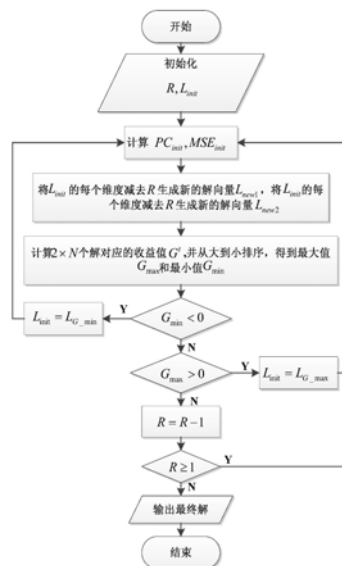
权利要求书2页 说明书9页 附图2页

(54)发明名称

一种基于贪婪算法的区域背光动态调光方法

(57)摘要

一种基于贪婪算法的区域背光动态调光方法,在通过基于图像亮度典型特征值统计确定一组初始背光亮度后,再应用贪婪算法重新设计出一种新型的区域背光动态调光算法。即,寻找最佳区域背光亮度调节方案的过程,算法终止后的最优解即为最终的区域背光亮度调节方案。在解的编码上,有如下对应关系:当背光分为N个区域,每个区域的亮度值为整数时,用一个N维向量L来表示问题的解。从而使得通过本发明的算法求得的区域背光亮度分配方案获得更高的图像质量。本发明确定的区域背光亮度分布方案不但能保证调光后图像的显示质量,还能较大程度地降低能耗,较好地平衡了图像显示质量与能耗之间的相互制约的关系。



1. 一种基于贪婪算法的区域背光动态调光方法,其特征在于,包括如下步骤:
- 1) 初始化图像像素点亮度的搜索步长R,确定能耗约束条件及迭代终止条件;
 - 2) 根据图像区域的亮度最大值生成初始解向量 L_{init} ;
 - 3) 分别计算出初始解向量 L_{init} 对应的目标函数值 MSE_{init} 以及初始能耗值 PC_{init} ,其中,所述的目标函数值 MSE_{init} 是采用如下公式计算:

$$MSE_{init} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (y'_{init,i,j} - y_{i,j})^2$$

所述的初始能耗值 PC_{init} 是采用如下公式计算:

$$PC_{init} = \frac{1}{N} \sum_{k=1}^N b_{init_k} \times 100\% \\ BL_{full}$$

式中, b_{init_k} 是第k个区域的初始背光亮度, $y_{i,j}$ 是像素点(i,j)区域调光前的像素亮度, $y'_{init,i,j}$ 是根据初始解向量 L_{init} 对像素点(i,j)区域调光后的像素亮度, BL_{full} 是背光全亮时的背光亮度,N是图像区域个数,m是输入图像的高,n输入图像的宽;

- 4) 通过初始解向量 L_{init} 和搜索步长R,生成 $2 \times N$ 个新的解向量 L_{new} ;

5) 将 $2 \times N$ 个新的解向量记为: $L_{new}^1, L_{new}^2, \dots, L_{new}^t, \dots, L_{new}^{2N-1}, L_{new}^{2N}$,分别计算 $2 \times N$ 个新的解向量的收益值,其中第t个新的解向量 L_{new}^t 的收益值 G^t 计算公式如下:

$$G^t = \begin{cases} \frac{|MSE_{new}^t - MSE_{init}|}{PC_{new}^t - PC_{init}}, & PC_{new}^t - PC_{init} > 0, MSE_{new}^t - MSE_{init} < 0 \\ \frac{|MSE_{new}^t - MSE_{init}|}{PC_{new}^t - PC_{init}}, & PC_{new}^t - PC_{init} < 0, MSE_{new}^t - MSE_{init} < 0 \\ 0, & MSE_{new}^t - MSE_{init} \leq 0 \end{cases}$$

其中,第t个新的解向量 L_{new}^t 的目标函数值 MSE_{new}^t 的计算公式如下:

$$MSE_{new}^t = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (y''_{new,i,j} - y_{i,j})^2$$

第t个新的解向量 L_{new}^t 的能耗值 PC_{new}^t 的计算公式如下:

$$PC_{new}^t = \frac{1}{N} \sum_{k=1}^N b_{new_k}^t \times 100\% \\ BL_{full}$$

式中, $b_{new_k}^t$ 是第t个新的解向量 L_{new}^t 中的第k个区域的背光亮度, $y_{i,j}$ 是像素点(i,j)区域调光前的像素亮度, $y''_{new,i,j}$ 是根据第t个新的解向量 L_{new}^t 对像素点(i,j)区域调光后的像素亮度, BL_{full} 是背光全亮时的背光亮度,N是图像区域个数,m是输入图像的高,n输入图像的宽;

- 6) 判断 $2 \times N$ 个新的解向量是否满足设定的能耗约束条件,若都不满足设定的能耗约束

条件,则继续步骤7),否则将满足能耗约束条件的解按照对应的收益值从大到小进行排序,若排序后的最小值 $G_{\min} < 0$,则用最小值 G_{\min} 所对应的解向量 $L_{G_{\min}}$ 来替换初始解向量 L_{init} ,即 $L_{init} = L_{G_{\min}}$ 后返回步骤3),若排序后的最小值 $G_{\min} \geq 0$ 且最大值 $G_{\max} > 0$,则用最大值 G_{\max} 所对应的解向量 $L_{G_{\max}}$ 来替换初始解向量 L_{init} ,即 $L_{init} = L_{G_{\max}}$ 后返回步骤3),否则,继续步骤7);

7) 令搜索步长 $R = R - 1$,判断 R 是否大于或等于1,若是,则返回步骤3),否则,跳转至步骤8);

8) 方法终止,输出初始解向量 L_{init} 作为最终解,即为最终的区域背光亮度分配方案。

2. 根据权利要求1所述的一种基于贪婪算法的区域背光动态调光方法,其特征在于,步骤1)所述的能耗约束为背光模组消耗的电量。

3. 根据权利要求1所述的一种基于贪婪算法的区域背光动态调光方法,其特征在于,步骤2)包括:

(1) 将图像按照背光区域划分方式分成相同的 N 个区域;

(2) 分别求每个区域所有像素点的亮度最大值,其中第 k 个区域的亮度最大值表示为 l_{init}^k ;

(3) 通过所有 N 个区域的亮度最大值生成一个初始解向量 L_{init} ,初始解向量 L_{init} 是一个 N 维向量, $L_{init} = [l_{init}^1, l_{init}^2, \dots, l_{init}^k, \dots, l_{init}^N]$,其中 l_{init}^k 是向量第 k 维的值。

4. 根据权利要求1所述的一种基于贪婪算法的区域背光动态调光方法,其特征在于,步骤4)包括:

分别将初始解向量 L_{init} 中的每个维度值 l_{init}^k 减去搜索步长 R ,获得 N 个新的解向量:

$L_{new1}^1, L_{new1}^2, \dots, L_{new1}^k, \dots, L_{new1}^N$; 其中 $L_{new1}^k = [l_{new1}^1, l_{new1}^2, \dots, l_{new1}^k, \dots, l_{new1}^N]$,为第 k 个新的解向量, l_{new1}^k 是第 k 个新的解向量中第 k 维向量的值,计算公式如下:

$$l_{new1}^k = \begin{cases} l_{init}^k - R, & l_{init}^k - R > 0 \\ 0, & l_{init}^k - R \leq 0 \end{cases}$$

然后再将初始解向量 L_{init} 中每个维度值 l_{init}^k 加上 R ,继续获得另外 N 个新的解向量:

$L_{new2}^1, L_{new2}^2, \dots, L_{new2}^k, \dots, L_{new2}^N$; 其中 $L_{new2}^k = [l_{new2}^1, l_{new2}^2, \dots, l_{new2}^k, \dots, l_{new2}^N]$,为第 k 个新的解向量, l_{new2}^k 是第 k 个新的解向量中第 k 维向量的值,计算公式如下:

$$l_{new2}^k = \begin{cases} l_{init}^k + R, & l_{init}^k - R > 0 \\ 0, & l_{init}^k - R \leq 0 \end{cases}$$

至此,共生成了 $2 \times N$ 个新的解向量。

一种基于贪婪算法的区域背光动态调光方法

技术领域

[0001] 本发明涉及一种调光方法。特别是涉及一种基于贪婪算法的区域背光动态调光方法。

背景技术

[0002] 1、区域背光亮度调节问题

[0003] 作为被动式发光器件,液晶显示器(Liquid Crystal Display,LCD)本身不发光,需要一个称为背光模组的组件给它提供光源以显示画面。当前LED背光液晶显示器在显示器行业占据着绝对的优势,随着背光技术发展的日益成熟,LCD的成本也随之逐渐下降,各大厂商现主要依靠能耗和显示质量来占领市场份额。由于LCD的能耗大约90%都消耗在背光源上,对于消费者来说,低能耗和优越的显示画质是他们购买带有LCD电子产品时最为看重的两个因素。传统的LCD,背光源亮度始终处于常亮状态,即无论显示较亮还是较暗的画面,背光源都全部点亮,这导致功耗较高,且画面对比度也较低进而使得显示效果不佳。为了降低功耗同时提高画面显示质量,区域背光动态调光技术被提出。该技术可以根据显示图像的内容,实时调节背光源LED的亮度,即当显示较暗画面时,可以适当降低LED的亮度,甚至在显示全黑画面时,可以将对应背光LED全部关闭。这样不仅可以降低功耗,还能有效提高LCD的对比度从而提升显示质量。目前,显示领域都在研发LED区域背光动态调光技术,以期获得最佳的节能和画质。

[0004] LED区域背光动态调光算法主要包含两个部分:分区背光亮度的提取和液晶像素的合理补偿。目前大多的分区背光亮度提取方法都是通过对显示图像内容的典型特征值统计,找出一种合理的确定方法。这些算法的优势是实现起来较为简单,但它们在确定分区背光亮度时往往只考虑了能耗与显示质量两者中的一个而忽略了另一个因素,这与研究LED区域背光动态调光技术的目标相违背。我们知道,分区背光亮度降低的越多越有利用降低能耗,但过度地降低背光亮度可能影响到图像的显示质量。因此,在确定分区背光亮度时应该综合考虑两方面的因素:能耗和显示质量。这两者之前不是完全对立的关系,在一定程度上,它们之前存在相互促进的作用。考虑到这两方面因素,可以将确定分区的最佳背光亮度看作是一个寻优的过程,即找到满足能耗与显示质量都达到最佳状态下的折衷点,如图1所示。因此,可以将区域背光亮度调节问题转化为一个最优化问题,即在所有的分区亮度分布方案中,一定存在一种方案满足在显示质量一定的情况下能耗最低或是能耗一定的情况拥有最高的显示质量。

[0005] 假设背光模组采用区域背光模式的LCD系统,其液晶面板大小为 $m \times n$,对于像素 i ,人眼从液晶屏上所感受的亮度为 y_{ii} ,则它表示为:

$$[0006] \quad y_{ii} = a_{ii} \cdot b_{ii} \quad (1)$$

[0007] 写成Hadamard积矩阵的形式:

$$[0008] \quad Y = a \circ b \quad (2)$$

[0009] 式中, a_{ii} 和 b_{ii} 分别是对应于像素 i 的液晶面板光透射系数和背光亮度。 $a \in \mathbb{R}^{m \times n}$ 是

液晶面板中每个液晶像素光透射系数构成的矩阵, $b \in \mathbb{R}^{m \times n}$ 是所有像素点背光亮度组成的矩阵。如果 $b_{ii} = 0$, 则说明像素 ii 的背光亮度为 0, 此时没有光照射到该像素上; 如果 $b_{ii} = 1$, 则说明此时对应于像素 ii 的背光是全开的。由于液晶本身的漏光特性, a_{ii} 的值不可能为 0。

[0010] 理论上, 背光模组可以做到每一个像素点对应于一个背光分区, 即每个像素都对应一个背光源 LED 灯, 但考虑到成本、散热性能等多方面的因素, 实际应用中都是一个分区包含成百上千个像素点, 这种情况下就存在分区之间光的串扰问题, 即像素点 i 的实际背光亮度不仅受来自它自身所在分区背光亮度的影响, 还受其他背光分区亮度的影响。所以, 在模拟光扩散的过程中, 可以将所有分区背光亮度对像素 i 背光亮度的贡献表示为点扩散函数 (Point Spread Function, PSF) 系数矩阵 H , 则 b_i 即是矩阵 H 中对应的点扩散函数向量 h 与相应的 LED 灯发光强度乘积的总和, 故有:

$$[0011] \quad b_{ii} = \sum_{jj=1}^N h_{ii,jj} \cdot r_{jj} \quad (3)$$

[0012] 写成矩阵相乘的形式:

$$[0013] \quad b = Hr \quad (4)$$

[0014] 其中, N 是 LCD 背光模组的分区总个数; $h_{ii,jj}$ 是第 jj 个分区对应于像素 ii 的点扩散函数系数, 表示第 jj 个分区背光亮度对像素 i 的贡献; r_{jj} 是第 jj 个分区的背光亮度。

[0015] 假设图像的显示质量表示为 Q , 能耗表示为 PC , 则寻优过程中的目标函数和约束条件可表示如下:

$$[0016] \quad \max: f(x) = Q \quad (5)$$

$$[0017] \quad \text{subject to: } PC \leq PC_{limit}$$

[0018] 其中, P_{limit} 是能耗约束的上限值。

[0019] 为了降低计算复杂度, 将图像处理中较为常用的图像质量客观评价指标均方误差 (Mean Square Error, MSE) 作为目标函数, 能耗看作约束条件, 则有:

$$[0020] \quad \begin{aligned} \max: & \quad f(x) = MSE \\ \text{subject to: } & \quad PC \leq PC_{limit} \end{aligned} \quad (6)$$

$$[0021] \quad MSE = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (y'_{i,j} - y_{i,j})^2 \quad (7)$$

$$[0022] \quad PC = \frac{1}{N} \sum_{k=1}^N b_k \times 100\% \quad (8)$$

[0023] 式中, b_i 是第 k 个分区的背光亮度, $y_{i,j}$ 和 $y'_{i,j}$ 分别是像素 (i, j) 区域调光前后的像素亮度, BL_{full} 是背光全亮时的背光亮度, 一般取 $BL_{full} = 255$ 。

[0024] 2、贪婪算法

[0025] 贪婪算法 (Greedy Algorithms) 是一种较为常用的启发式算法, 在对问题求解时, 总是做出在当前看来是最好的选择。也就是说, 不从整体最优上加以考虑, 他所做出的是在某种意义上的局部最优解。贪婪算法的关键在于贪婪策略的选择, 算法将会根据这个策略来判断当前哪一种选择最为合适。贪婪算法具备求解速度快, 算法易实现的那个特点, 因此常被用来求解最优化问题。贪婪算法的基本原理如下:

- [0026] 1、建立数学模型来描述问题；
 [0027] 2、把求解的问题分成若干个子问题；
 [0028] 3、对每一子问题求解，得到子问题的局部最优解；
 [0029] 4、把子问题的解局部最优解合成原来解问题的一个解。

发明内容

[0030] 本发明所要解决的技术问题是，提供一种在相同或者更低的能耗下可以使区域调光后的图像获得更高的显示质量的基于贪婪算法的区域背光动态调光方法。

[0031] 本发明所采用的技术方案是：一种基于贪婪算法的区域背光动态调光方法，包括如下步骤：

[0032] 1) 初始化图像像素点亮度的搜索步长R，确定能耗约束条件及迭代终止条件；

[0033] 2) 根据图像区域的亮度最大值生成初始解向量L_{init}；

[0034] 3) 分别计算出初始解向量L_{init}对应的目标函数值MSE_{init}以及初始能耗值PC_{init}，其中，所述的目标函数值MSE_{init}是采用如下公式计算：

$$[0035] \quad MSE_{init} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (y'_{init,i,j} - y_{i,j})^2$$

[0036] 所述的初始能耗值PC_{init}是采用如下公式计算：

$$[0037] \quad PC_{init} = \frac{1}{N} \sum_{k=1}^N b_{init,k} \times 100\%$$

BL_{full}

[0038] 式中， $b_{init,k}$ 是第k个区域的初始背光亮度， $y_{i,j}$ 是像素点(i, j)区域调光前的像素亮度， $y'_{init,i,j}$ 是根据初始解向量L_{init}对像素点(i, j)区域调光后的像素亮度， BL_{full} 是背光全亮时的背光亮亮度，N是图像区域个数，m是输入图像的高，n输入图像的宽；

[0039] 4) 通过初始解向量L_{init}和搜索步长R，生成2×N个新的解向量L_{new}；

[0040] 5) 将2×N个新的解向量记为： $L_{new}^1, L_{new}^2, \dots, L_{new}^t, \dots, L_{new}^{2N-1}, L_{new}^{2N}$ ，分别计算2×N个新的解向量的收益值，其中第t个新的解向量L_{new}^t的收益值G^t计算公式如下：

$$[0041] \quad G^t = \begin{cases} \frac{|MSE_{new}^t - MSE_{init}|}{PC_{new}^t - PC_{init}}, & PC_{new}^t - PC_{init} > 0, MSE_{new}^t - MSE_{init} < 0 \\ \frac{|MSE_{new}^t - MSE_{init}|}{PC_{new}^t - PC_{init}}, & PC_{new}^t - PC_{init} < 0, MSE_{new}^t - MSE_{init} < 0 \\ 0 & , MSE_{new}^t - MSE_{init} \leq 0 \end{cases}$$

[0042] 其中，第t个新的解向量L_{new}^t的目标函数值MSE_{new}^t的计算公式如下：

$$[0043] \quad MSE_{new}^t = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (y'_{new,i,j} - y_{i,j})^2$$

[0044] 第t个新的解向量L_{new}^t的能耗值PC_{new}^t的计算公式如下：

$$[0045] \quad PC_{new}^t = \frac{\frac{1}{N} \sum_{k=1}^N b_{new_k}^t}{BL_{full}} \times 100\%$$

[0046] 式中, $b_{new_k}^t$ 是第t个新的解向量 L_{new}^t 中的第k个区域的背光亮度, $y_{i,j}$ 是像素点 (i, j) 区域调光前的像素亮度, $y_{new_{i,j}}^t$ 是根据第t个新的解向量 L_{new}^t 对像素点 (i, j) 区域调光后的像素亮度, BL_{full} 是背光全亮时的背光亮度, N 是图像区域个数, m 是输入图像的高, n 输入图像的宽;

[0047] 6) 判断 $2 \times N$ 个新的解向量是否满足设定的能耗约束条件, 若都不满足设定的能耗约束条件, 则继续步骤7), 否则将满足能耗约束条件的解按照对应的收益值从大到小进行排序, 若排序后的最小值 $G_{min} < 0$, 则用最小值 G_{min} 所对应的解向量 $L_{G_{min}}$ 来替换初始解向量 L_{init} , 即 $L_{init} = L_{G_{min}}$ 后返回步骤3), 若排序后的最小值 $G_{min} \geq 0$ 且最大值 $G_{max} > 0$, 则用最大值 G_{max} 所对应的解向量 $L_{G_{max}}$ 来替换初始解向量 L_{init} , 即 $L_{init} = L_{G_{max}}$ 后返回步骤3), 否则, 继续步骤7);

[0048] 7) 令搜索步长 $R = R - 1$, 判断R是否大于或等于1, 若是, 则返回步骤3), 否则, 跳转值步骤8);

[0049] 8) 方法终止, 输出初始解向量 L_{init} 作为最终解, 即为最终的区域背光亮度分配方案。

[0050] 步骤1) 所述的能耗约束为背光模组消耗的电量。

[0051] 步骤2) 包括:

[0052] (1) 将图像按照背光区域划分方式分成相同的N个区域;

[0053] (2) 分别求每个区域所有像素点的亮度最大值, 其中第k个区域的亮度最大值表示为 l_{init}^k ;

[0054] (3) 通过所有N个区域的亮度最大值生成一个初始解向量 L_{init} , 初始解向量 L_{init} 是一个N维向量, $L_{init} = [l_{init}^1, l_{init}^2, \dots, l_{init}^k, \dots, l_{init}^N]$, 其中 l_{init}^k 是向量第k维的值。

[0055] 步骤4) 包括:

[0056] 分别将初始解向量 L_{init} 中的每个维度值 l_{init}^k 减去搜索步长R, 获得N个新的解向量:

$L_{new1}^1, L_{new1}^2, \dots, L_{new1}^k, \dots, L_{new1}^N$; 其中 $L_{new1}^k = [l_{new1}^1, l_{new1}^2, \dots, l_{new1}^k, \dots, l_{new1}^N]$, 为第k个新的解向量, l_{new1}^k 是第k个新的解向量中第k维向量的值, 计算公式如下:

$$[0057] \quad l_{new1}^k = \begin{cases} l_{init}^k - R, & l_{init}^k - R > 0 \\ 0, & l_{init}^k - R \leq 0 \end{cases}$$

[0058] 然后再将初始解向量 L_{init} 中每个维度值 l_{init}^k 加上R, 继续获得另外N个新的解向量:

$L_{new2}^1, L_{new2}^2, \dots, L_{new2}^k, \dots, L_{new2}^N$; 其中 $L_{new2}^k = [l_{new2}^1, l_{new2}^2, \dots, l_{new2}^k, \dots, l_{new2}^N]$, 为第k个新的解向量, l_{new2}^k 是第k个新的解向量中第k维向量的值, 计算公式如下:

$$[0059] \quad l_{new2}^k = \begin{cases} l_{init}^k + R, & l_{init}^k - R > 0 \\ 0, & l_{init}^k - R \leq 0 \end{cases}$$

[0060] 至此,共生成了 $2 \times N$ 个新的解向量。

[0061] 本发明的一种基于贪婪算法的区域背光动态调光方法,用本发明方法确定的区域背光亮度分布方案不但能保证调光后图像的显示质量,还能较大程度地降低能耗,较好地平衡了图像显示质量与能耗之间的相互制约的关系。

附图说明

[0062] 图1是能耗、图像质量、背光亮度三者之间的对应关系图;

[0063] 图2是本发明一种基于贪婪算法的区域背光动态调光方法的流程图。

具体实施方式

[0064] 下面结合实施例和附图对本发明的一种基于贪婪算法的区域背光动态调光方法做出详细说明。

[0065] 本发明的一种基于贪婪算法的区域背光动态调光方法,包括如下步骤:

[0066] 1) 初始化图像像素点亮度的搜索步长 R ,确定能耗约束条件及迭代终止条件,所述的能耗约束为背光模组消耗的电量;

[0067] 2) 根据图像区域的亮度最大值生成初始解向量 L_{init} ;包括:

[0068] (1) 将图像按照背光区域划分方式分成相同的 N 个区域;

[0069] (2) 分别求每个区域所有像素点的亮度最大值,其中第 k 个区域的亮度最大值表示为 l_{init}^k ;

[0070] (3) 通过所有 N 个区域的亮度最大值生成一个初始解向量 L_{init} ,初始解向量 L_{init} 是一个 N 维向量, $L_{init} = [l_{init}^1, l_{init}^2, \dots, l_{init}^k, \dots, l_{init}^N]$,其中 l_{init}^k 是向量第 k 维的值。

[0071] 3) 分别计算出初始解向量 L_{init} 对应的目标函数值 MSE_{init} 以及初始能耗值 PC_{init} ,其中,

[0072] 所述的目标函数值 MSE_{init} 是采用如下公式计算:

$$[0073] \quad MSE_{init} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (y'_{init,i,j} - y_{i,j})^2$$

[0074] 所述的初始能耗值 PC_{init} 是采用如下公式计算:

$$[0075] \quad PC_{init} = \frac{1}{N} \sum_{k=1}^N b_{init_k} \times 100\%$$

[0076] 式中, b_{init_k} 是第 k 个区域的初始背光亮度, $y_{i,j}$ 是像素点 (i,j) 区域调光前的像素亮度, $y'_{init,i,j}$ 是根据初始解向量 L_{init} 对像素点 (i,j) 区域调光后的像素亮度, BL_{full} 是背光全亮时的背光亮度, N 是图像区域个数, m 是输入图像的高, n 输入图像的宽;

[0077] 4) 通过初始解向量 L_{init} 和搜索步长 R ,生成 $2 \times N$ 个新的解向量 L_{new} ;包括:

[0078] 分别将初始解向量 L_{init} 中的每个维度值 l_{init}^k 减去搜索步长 R ,获得 N 个新的解向量: $L_{new1}^1, L_{new1}^2, \dots, L_{new1}^k, \dots, L_{new1}^N$;其中 $L_{new1}^k = [l_{new1}^1, l_{new1}^2, \dots, l_{new1}^k, \dots, l_{new1}^N]$,为第 k 个新的解向量, l_{new1}^k 是第 k 个新的解向量中第 k 维向量的值,计算公式如下:

$$[0079] \quad l_{new1}^k = \begin{cases} l_{init}^k - R, & l_{init}^k - R > 0 \\ 0, & l_{init}^k - R \leq 0 \end{cases}$$

[0080] 然后再将初始解向量 L_{init} 中每个维度值 l_{init}^k 加上 R ,继续获得另外 N 个新的解向量: $L_{new2}^1, L_{new2}^2, \dots, L_{new2}^k, \dots, L_{new2}^N$;其中 $L_{new2}^k = [l_{new2}^1, l_{new2}^2, \dots, l_{new2}^k, \dots, l_{new2}^N]$,为第 k 个新的解向量, l_{new2}^k 是第 k 个新的解向量中第 k 维向量的值,计算公式如下:

$$[0081] \quad l_{new2}^k = \begin{cases} l_{init}^k + R, & l_{init}^k - R > 0 \\ 0, & l_{init}^k - R \leq 0 \end{cases}$$

[0082] 至此,共生成了 $2 \times N$ 个新的解向量。

[0083] 5) 将 $2 \times N$ 个新的解向量记为: $L_{new}^1, L_{new}^2, \dots, L_{new}^t, \dots, L_{new}^{2N-1}, L_{new}^{2N}$,分别计算 $2 \times N$ 个新的解向量的收益值,其中第 t 个新的解向量 L_{new}^t 的收益值 G^t 计算公式如下:

$$[0084] \quad G^t = \begin{cases} \frac{|MSE_{new}^t - MSE_{init}|}{PC_{new}^t - PC_{init}}, & PC_{new}^t - PC_{init} > 0, MSE_{new}^t - MSE_{init} < 0 \\ \frac{|MSE_{new}^t - MSE_{init}|}{PC_{new}^t - PC_{init}}, & PC_{new}^t - PC_{init} < 0, MSE_{new}^t - MSE_{init} < 0 \\ 0, & MSE_{new}^t - MSE_{init} \leq 0 \end{cases}$$

[0085] 其中,第 t 个新的解向量 L_{new}^t 的目标函数值 MSE_{new}^t 的计算公式如下:

$$[0086] \quad MSE_{new}^t = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (y_{new_{i,j}}^t - y_{i,j})^2$$

[0087] 第 t 个新的解向量 L_{new}^t 的能耗值 PC_{new}^t 的计算公式如下:

$$[0088] \quad PC_{new}^t = \frac{1}{N} \sum_{k=1}^N b_{new_k}^t \times 100\%$$

[0089] 式中, $b_{new_k}^t$ 是第 t 个新的解向量 L_{new}^t 中的第 k 个区域的背光亮度, $y_{i,j}$ 是像素点 (i, j) 区域调光前的像素亮度, $y_{new_{i,j}}^t$ 是根据第 t 个新的解向量 L_{new}^t 对像素点 (i, j) 区域调光后的像素亮度, BL_{full} 是背光全亮时的背光亮度, N 是图像区域个数, m 是输入图像的高, n 输入图像的宽;

[0090] 6) 判断 $2 \times N$ 个新的解向量是否满足设定的能耗约束条件,若都不满足设定的能耗约束条件,则继续步骤7),否则将满足能耗约束条件的解按照对应的收益值从大到小进行排序,若排序后的最小值 $G_{min} < 0$,则用最小值 G_{min} 所对应的解向量 $L_{G_{min}}$ 来替换初始解向量

L_{init} ,即 $L_{init}=L_{G_min}$ 后返回步骤3),若排序后的最小值 $G_{min} \geq 0$ 且最大值 $G_{max} > 0$,则用最大值 G_{max} 所对应的解向量 L_{G_max} 来替换初始解向量 L_{init} ,即 $L_{init}=L_{G_max}$ 后返回步骤3),否则,继续步骤7);

[0091] 7) 令搜索步长 $R=R-1$,判断 R 是否大于或等于1,若是,则返回步骤3),否则,跳转值步骤8);

[0092] 8) 方法终止,输出新的解向量 L_{new} 作为最终解,即为最终的区域背光亮度分配方案。

[0093] 下面给出具体实例:

[0094] 当背光分为35个区域,每个区域的背光值为0到255之间的整数时,本发明的最佳实施如下:

[0095] (1) 初始化搜索步长 $R=20$,能耗约束条件为以最大值法进行区域背光亮度调节时所需的能耗值。

[0096] (2) 将图像按照背光的区域划分方式分成相同的35个区域,分别求每个区域所有像素点的亮度最大值,其中第 k 个区域的亮度最大值表示为 l_{init}^k 。然后通过所有35个区域的亮度最大值生成一个初始解向量 L_{init} , L_{init} 是一个35维向量, $L_{init} = [l_{init}^1, l_{init}^2, \dots, l_{init}^k, \dots, l_{init}^{35}]$,其中 l_{init}^k 是向量第 k 维的值。

[0097] (3) 分别计算出初始解向量 L_{init} 对应的目标函数值 MSE_{init} 以及初始能耗值 PC_{init} ,其中,

[0098] 所述的目标函数值 MSE_{init} 是采用如下公式计算:

$$[0099] \quad MSE_{init} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (y'_{init,i,j} - y_{i,j})^2$$

[0100] 所述的初始能耗值 PC_{init} 是采用如下公式计算:

$$[0101] \quad PC_{init} = \frac{1}{N} \sum_{k=1}^N \frac{b_{init_k}}{BL_{full}} \times 100\%$$

[0102] 式中, b_{init_k} 是第 k 个区域的初始背光亮度, $y_{i,j}$ 是像素点 (i,j) 区域调光前的像素亮度, $y'_{init,i,j}$ 是根据初始解向量 L_{init} 对像素点 (i,j) 区域调光后的像素亮度, BL_{full} 是背光全亮时的背光亮度,一般取 $BL_{full}=255$, N 是图像区域个数, m 是输入图像的高, n 输入图像的宽。

[0103] (4) 分别将初始解向量 L_{init} 中的每个维度值减去搜索步长 $R=20$,获得35个新的解向量: $L_{new1}^1, L_{new1}^2, \dots, L_{new1}^k, \dots, L_{new1}^{35}$ 。其中 $L_{new1}^k = [l_{new1}^1, l_{new1}^2, \dots, l_{new1}^k, \dots, l_{new1}^{35}]$,为第 k 个新的解向量, l_{new1}^k 是第 k 个新的解向量中第 k 维向量的值,计算公式如下:

$$[0104] \quad l_{new1}^k = \begin{cases} l_{init}^k - R, & l_{init}^k - R > 0 \\ 0, & l_{init}^k - R \leq 0 \end{cases}$$

[0105] 然后再将初始解向量 L_{init} 中每个维度值 l_{init}^k 加上 R ,继续获得另外35个新的解向量:

$L_{new2}^1, L_{new2}^2, \dots, L_{new2}^k, \dots, L_{new2}^{35}$; 其中 $L_{new2}^k = [l_{new2}^1, l_{new2}^2, \dots, l_{new2}^k, \dots, l_{new2}^{35}]$, 为第k个新的解向量, l_{new2}^k 是第k个新的解向量中第k维向量的值, 计算公式如下:

$$[0106] \quad l_{new2}^k = \begin{cases} l_{init}^k + R, & l_{init}^k - R > 0 \\ 0, & l_{init}^k - R \leq 0 \end{cases}$$

[0107] 至此, 共生成了70个新的解。

[0108] (5) 将这70个新的解计为: $L_{new}^1, L_{new}^2, \dots, L_{new}^t, \dots, L_{new}^{69}, L_{new}^{70}$ 。分被计算这70个解的收益值, 其中第t个解 L_{new}^t 的收益值 G^t 计算公式如下

$$[0109] \quad G^t = \begin{cases} \frac{|MSE_{new}^t - MSE_{init}^t|}{PC_{new}^t - PC_{init}^t}, & PC_{new}^t - PC_{init}^t > 0, MSE_{new}^t - MSE_{init}^t < 0 \\ \frac{|MSE_{new}^t - MSE_{init}^t|}{PC_{new}^t - PC_{init}^t}, & PC_{new}^t - PC_{init}^t < 0, MSE_{new}^t - MSE_{init}^t < 0 \\ 0, & MSE_{new}^t - MSE_{init}^t \leq 0 \end{cases}$$

[0110] 其中, 第t个新的解向量 L_{new}^t 的目标函数值 MSE_{new}^t 的计算公式如下:

$$[0111] \quad MSE_{new}^t = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (y_{new_{i,j}}^t - y_{i,j})^2$$

[0112] 第t个新的解向量 L_{new}^t 的能耗值 PC_{new}^t 的计算公式如下:

$$[0113] \quad PC_{new}^t = \frac{1}{N} \sum_{k=1}^N b_{new_k}^t \times 100\%$$

[0114] 式中, $b_{new_k}^t$ 是第t个新的解向量 L_{new}^t 中的第k个区域的背光亮度, $y_{i,j}$ 是像素点 (i, j) 区域调光前的像素亮度, $y_{new_{i,j}}^t$ 是根据第t个新的解向量 L_{new}^t 对像素点 (i, j) 区域调光后的像素亮度, BL_{full} 是背光全亮时的背光亮度, N 是图像区域个数, m 是输入图像的高, n 输入图像的宽。

[0115] (6) 判断70个新生成的解向量是否满足功耗约束条件, 若都不满足设定的能耗约束条件, 则继续步骤(7), 否则将满足能耗约束条件的解按照对应的收益值从大到小进行排序, 若排序后的最小值 $G_{min} < 0$, 则用最小值 G_{min} 所对应的解向量 $L_{G_{min}}$ 来替换初始解向量 L_{init} , 即 $L_{init} = L_{G_{min}}$ 后返回步骤(3), 若排序后的最小值 $G_{min} \geq 0$ 且最大值 $G_{max} > 0$, 则用最大值 G_{max} 所对应的解向量 $L_{G_{max}}$ 来替换初始解向量 L_{init} , 即 $L_{init} = L_{G_{max}}$ 后返回步骤(3), 否则, 继续步骤(7)。

[0116] (7) 令搜索步长 $R = R - 1$, 然后判断 R 是否大于或等于1, 若是, 则返回步骤(3), 否则, 跳转值步骤(8)。

[0117] (8) 算法终止, 输出 L_{init} 作为最终解, 即为最终的区域背光亮度分配方案。

[0118] 为了测试本发明的一种基于贪婪算法的区域背光动态调光方法的性能, 选取了亮

度覆盖范围较广的4种不同场景的图像,与最大值法(一种基于图像亮度特征参数最大值的区域背光动态调光算法)进行性能对比仿真测试。仿真实验是在MATLAB R2010b环境中进行的,所有测试图像的分辨率为 1920×1080 。针对4幅测试图像,本发明的方法与最大值法在功耗相同的情况下所获得的调光后图像的显示质量用峰值信噪比PSNR来表示,PSNR越高则图像质量越好,其对比结果如表1所示。实验结果表明,与基于图像特征参数的区域背光动态调光算法相比,本发明的方法在相同或者更低的能耗下可以使区域调光后的图像获得更高的显示质量。

[0119] 表1所发明算法与最大值法性能对比

[0120]

PSNR(dB) 图像	方法	最大值法	本发明的方法	PSNR 提升百分比 (%)
高亮度图像		33.08	35.42	7.07
低亮度图像		30.90	33.83	9.48
高对比度图像		29.29	36.62	25.03
低对比度图像		28.46	31.90	12.09

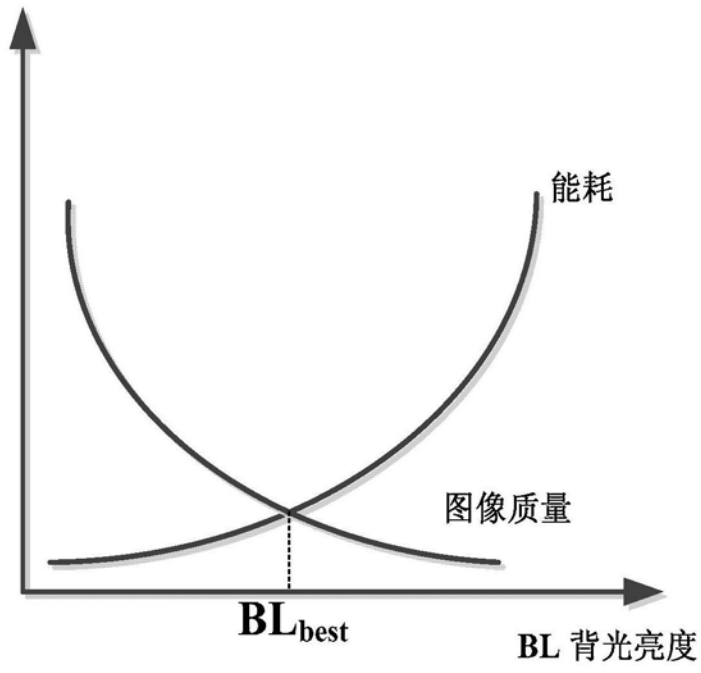


图1

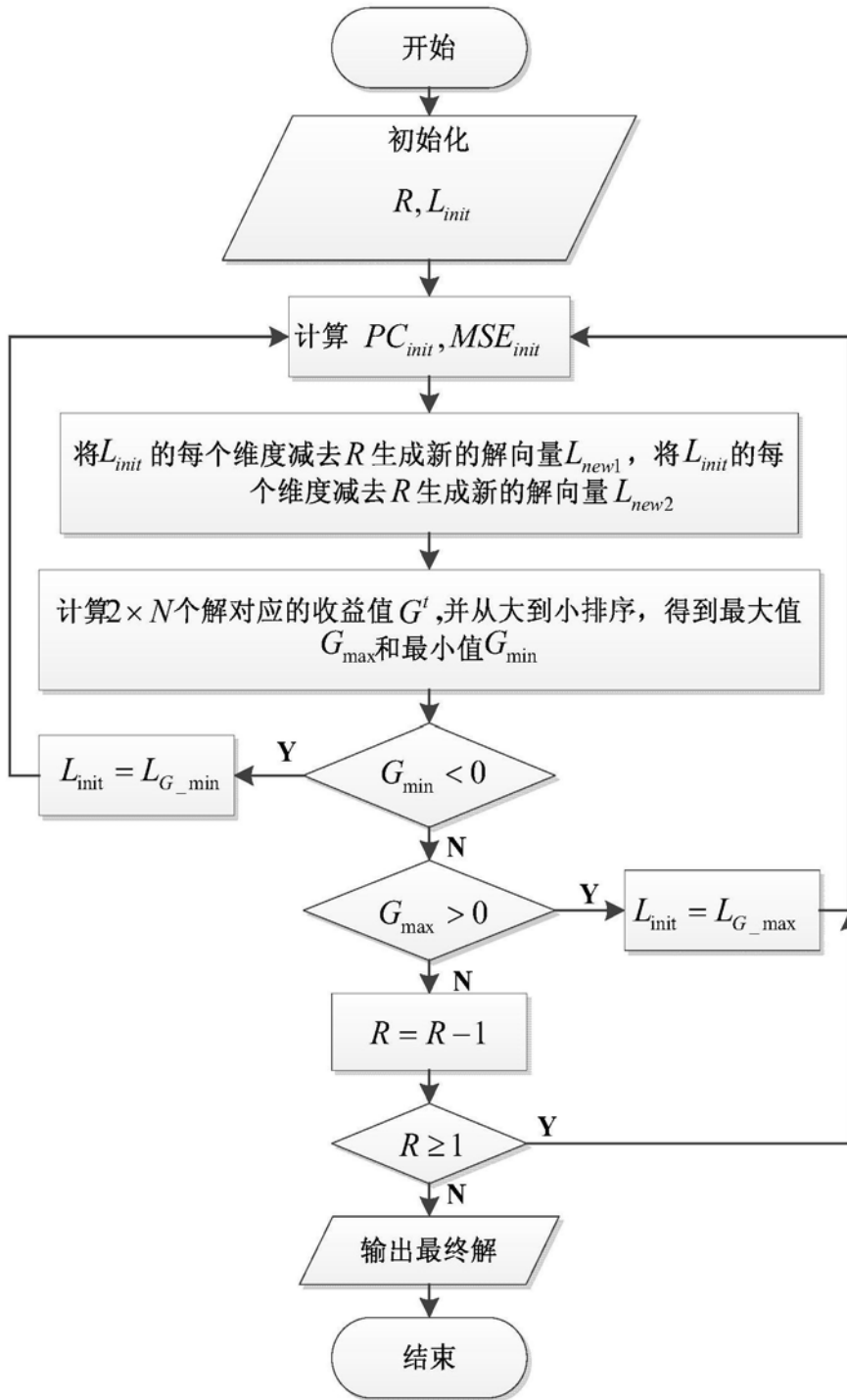


图2