基于TGV和灰度通道相似性的深度补全

lwj

2021 年 11 月 13 日

目录

1 利用优化实现颜色修复

1.1 局部颜色相似性假设

深度恢复和颜色恢复从某种程度上是一样的，Levin[1]假设在邻域内，图像颜色存在线性相关性，他利用这种相关性对彩色图像进行了颜色恢复(colorization)的工作，Levin[1]提到相似性的两种度量方式如下：

$$w(i,j) \propto 1+\frac{1}{\sigma_j^2}(I(i)-\mu_i)(I(j)-\mu_i) \tag{1}$$

$$w(i,j) \propto \exp\frac{-\left(I(i)-I(j)\right)^2}{2\sigma_i^2} \tag{2}$$

式(1) 和 (2)中 $w(i,j)$表示在j和i处颜色的线性因子，一个朴素的假设是不同通道局部相似性因子$w(i,j)$是相近的，因此在已知其中一个通道数据的情况下，其他通道的w，可以通过已知通道（例如灰度I)近似求解。

1.2 基于局部颜色相似性的颜色修复模型

对于m行n列的图像，假设已知某一通道的数据I，待恢复的颜色通道的数据为f,其中未知颜色数据的区域为Z，所有图像区域为$\Omega=[1,mn)$，$u,f\in R^{mn}$,$f(i)=0,i\in Z$，通过求解损失函数(3)最小值得问题 ,即可恢复缺失的颜色数据

$$J(u)=\sum_{i\in Z}\left\|u(i)-\sum_{j\in N(i,r),j\neq i}w(i,j)u(j)\right\|^2 \tag{3}$$

(3)可以写成矩阵的形式：

$$\begin{aligned} J(u) &= \left\|\left(I-W\right)u-f\right\|^2 \\ &= \sum_{i\in Z}\left(u(i)-\sum_{j\in N(i,r)}w(i,j)u(j)\right)^2+\sum_{i\in\Omega\setminus Z}\left(u(i)-f(i)\right)^2 \end{aligned} \tag{4}$$

其中稀疏矩阵$W\in R^{mn\times mn}$如下

$$W(i,j)=\begin{cases} \frac{w(i,j)}{\sum\limits_{j\in N(i,r),j\neq i}w(i,j)} & ,j\in N(i,j),j\neq i,i\in Z \\ 0 & ,others \end{cases} \tag{5}$$

最小化(4)可以通过最小二乘法进行求解

$$u = \left((I-W)^T(I-M)\right)^{-1}(I-W)^T f \tag{6}$$

(6)可以通过$eigen$求解，但是实际发现求解稀疏矩阵线性问题的效率上，$matlab$会更高一些。

1.3 模型改进

此外flyfj[2]对(4)的一种改进为

$$\begin{aligned} J(u) &= \|((\alpha_c G + I - W')\,u - \alpha_c f\|^2, \alpha_c \in [0,1] \\ &= \overbrace{\sum_{i\in\Omega}\left(u(i) - \sum_{j\in N(i,r)} w(i,j)u(j)\right)^2}^{\text{约束项}} + \underbrace{\alpha_c \sum_{i\in\Omega\setminus Z}(u(i)-f(i))^2}_{\text{保真项}} \end{aligned} \tag{7}$$

$$W'(i,j) = \begin{cases} \dfrac{w(i,j)}{\sum\limits_{j\in N(i,r), j\neq i} w(i,j)} & , j\in N(i,j), j\neq i, i\in\Omega \\ 0 & , others \end{cases} \tag{8}$$

$$G(i,j) = \begin{cases} 1, i=j \ and \ f(i)\neq 0 \\ 0, others. \end{cases} \tag{9}$$

相似性约束的范围也从未知图像区域Z扩展到整个图像区域Ω，改进之后的模型能够更好的约束修复数据，不仅包含已知$\Omega\setminus Z$对Z的约束，同时Z对其他区域$\Omega\setminus Z$也被考虑进去了。引入了惩罚因子α_c，表征保真项的权值。改进之后，其深度恢复效果得到很大的提升。图1是仅使用未知区域约束项的效果，图2是使用全局约束项的效果，对应的惩罚因子$\alpha_c = 1.0$。可以看出深度修复的效果得到明显提升，一个原因是用来恢复深度信息的数据更多了。

另一种改进是针对线性系数$w(i,j)$的一种改进。Levin[1]的模型在图像边缘处的效果比较差，特别是未知区域Z和已知区域$Y=\Omega\setminus Z$的边缘正好和rgb图像边缘正好重叠的区域(见图3),由于该区域其他方向的深度信息不

图 1: 只使用未知区域局部约束

图 2: 使用了全局约束项

可用，而已知区域的深度信息对该区域又通过一个比较小的权值$w(i,j)$进行影响。这种区域在进行深度数据坐标系变换之后是非常常见的区域。因此可以对图像进行语义分割，划分成不同的区块，对属于不同区块的权

图 3: 优化效果比较差的区域,图中红框

值$w(i,j)$直接置零。

$$W''(i,j)=\begin{cases}\frac{w(i,j)}{\sum\limits_{j\in N(i,r),j\neq i} w(i,j)} & ,j\neq i, seg(i)=seg(j), j\in N(i,j), i\in\Omega \\ 0 & ,others\end{cases} \tag{10}$$

其中$seg(i)$返回i处的所属的语义分割集合的id。如图由于减少了不同区域的错误影响，其优化效果得到一定的提升。

由于将属于不同区域的权值置零了。矩阵W'的稀疏程度增加，容易产生奇异性。需要进一步考虑算法处理。

图 4: 使用全局约束但是没有利用语义分割

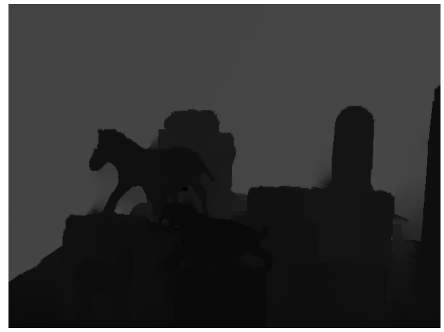

图 5: 利用语义分割优化全局约束

2 基于tgv的深度数据恢复

2.1 ITGV

Ranftl[3]提出使用Nagel-Enkelmann算子[4]将结构信息融入到TGV模型中的正则化手段ITGV。其公式如(11)

$$ITGV_\alpha^2(u) = \min_{w \in \mathbb{R}^2} \left\{ \alpha_u \int_\Omega \left| D^{\frac{1}{2}} \nabla u - w \right| dx + \alpha_w \int_\Omega |\nabla w| \, dx \right\} \tag{11}$$

其中的$D^{\frac{1}{2}}$为Nagel-Enkelmann算子

$$D^{\frac{1}{2}} = \exp\left(-\gamma \left| \nabla I_L \right|^\beta \right) n n^T + n^\perp {n^\perp}^T \tag{12}$$

n是图像梯度的方向，$n = \frac{\nabla I_L}{|\nabla I_L|}$，$n^\perp$垂直于$n$，$\gamma$和$\beta$是自定义参数。Ferstl[5]进一步的提出，用ITGV进行深度数据上采样的方法，并取得了较好的效果。其模型见下式(13)

$$J(u,w) = ITGV_\alpha^2(u,w) + \int_\Omega \lambda \left| u - f \right|^2 dx \tag{13}$$

其中系数λ在深度数据缺失的区域为零。即$\lambda(i) = 0, i \in Z$。TGV正则化，能够在保留边缘的同时，去除噪声，在已有深度数据的区域$D = \Omega \setminus Z$ 由于其具有各项异性的传播特性，能够衰减梯度方向的传播,保留梯度切线方向的传播，因此能够比较好的保留原始图像的边缘信息，但是在深度为零的区域，这种正则化作用由于没有原始数据的边缘梯度做为参考，这种正则化作用就被减弱了。因此通过引入结构化参数$D^{\frac{1}{2}}$将灰度通道的数据的边缘信息作为参照，保证了ITGV在深度缺失区域的各项异性。

2.2 STGV

Drozdov进一步的提出改进的边缘算子[6]用于深度恢复。并在已有的数据集上取得了不错的效果。STGV基于超像素技术，将rgb图像进行超像素处理，计算每个超像素内的深度分布，提出异常点，利用$DBSCAN$聚类算法实现超像素的合并。同时利用贪婪算法，保证每个超像素块内的已知深度区域具有足够的比例。利用超像素信息，进一步加强边缘算子的各向异性，见式(14)。

$$\mathbb{D} = \Gamma n n^T + n^{\perp} n^{\perp T} \tag{14}$$

其中Γ为边缘指示函数，见式(15)，

$$\Gamma = \begin{cases} 1 \ inside\ segment, \\ 0 \ segment\ border \end{cases} \tag{15}$$

将式(14)代入式(13)中得到

$$J(u) = \min_{w \in \mathbb{R}^2} \left\{ \alpha_u \int_{\Omega} |(D)\nabla u - w| \, dx + \alpha_w \int_{\Omega} |\nabla w| \, dx \right\} + \int_{\Omega \setminus Z} \lambda \, |u - f|^2 \, dx \tag{16}$$

超像素边缘在一定程度是贴合物体的轮廓，在超像素边缘(一定程度上也是图像边缘)区域，通过(15)完全抑制了法线方向的扩散，深度修复具有各向异性。而在超像素内部，具有各向同性传播。因此STGV能够取得由于ITGV的效果。见图6和7，迭代了14730次（耗时1009秒），参数为：$\alpha_w = 5, \alpha_u = 1.2, \lambda = 50, \gamma = 0.85, \beta = 9$。

图 6: ITGV深度修复效果

图 7: STGV效果(图像边缘经过手动调整)

STGV存在一下几个问题：

1) 效果严重依赖超像素算法对边缘识别的准确度，如果边缘识别错误，则深度修复会带来很严重的错误，修复效果还不如ITGV正则化的效果，见图8和图9

2) 优化算法需要迭代很久才能够收敛。对于648 × 486分辨率的数据，在$I7\ GTX$2080配置的电脑上跑了数个小时都优化不出来。

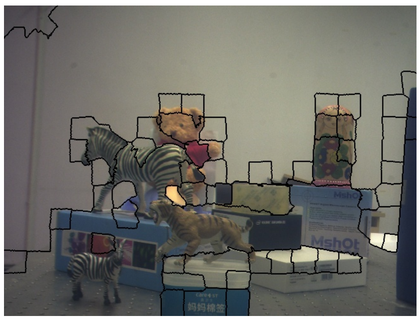

图 8: 错误的超像素结果

图 9: 错误超像素产生的深度修复效果

注意，为了增加图像对比度，图*6,7,9*做了最大值归一化处理。后续深度图如无特殊说明，均做了同样的处理

2.3 TGV收敛问题

TGV问题一般使用交替方向乘子法（$Alternating\ Direction\ Method\ of\ Multipliers, ADMM$求解。Chambolle[7]提出了基于prime-dual的算法实现,见算法1，收敛速度优于交替方向乘子法。对于求解鞍点问题(17)

$$\min_{x \in X} \max_{y \in Y} \langle Kx, y \rangle + G(x) - F^*(y) \tag{17}$$

其中$\langle \cdot, \cdot \rangle$ 表示对偶积，离散情况下可以看成是向量的点乘。可以使用算法1进行求解。 其中$L = \|K\|$表示算子K的范数。

针对K比较复杂的情况，其范数往往比较难计算，可以根据Pock[8]提到的做法，用算法2求解 算法2可以看成是Chambolle[7]提到的$Algorithm$2.和Pock[8]提到的算法的结合版本。

Algorithm 1 pdAlg1

Initialization: $Choose \tau, \sigma > 0, \theta \in [0,1], (x^0, y^0) \in X \times Y \ and \ set \ \overline{x}^0 = x^0 . s.t. \tau\sigma L^2 < 1$

Iterations($n \geq 0$): $Update \ x^n, y^n, \overline{x}^n \ as \ follows:$

$$\begin{cases} y^{n+1} = (I + \sigma\partial F^*)^{-1}(y^n + \sigma K\overline{x}^n) \\ x^{n+1} = (I + \tau\partial G)^{-1}(x^n - \tau K^* y^{n+1}) \\ \overline{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n) \end{cases} \tag{18}$$

Algorithm 2 pdPrecondition

Initialization: $Choose \tau_0, \sigma_0 > 0 \ with \ \tau_0\sigma_0 L^2 \leq 1, (x^0, y^0) \in X \times Y \ and \ set \ \overline{x}^0 = x^0.$

Caculate Tensors:$Choose \ \alpha \in [0,2] \ and \ Caculate \ as \ follow:$

$$\begin{aligned} T = diag(\boldsymbol{\tau}), where \ \boldsymbol{\tau} = (\tau_1, ..., \tau_n), \tau_j = \frac{1}{\sum_{i=1}^{m} |K_{i,j}|^{2-\alpha}}, \\ \Sigma = diag(\boldsymbol{\sigma}), where \ \boldsymbol{\sigma} = (\sigma_1, ..., \sigma_m). \sigma_i = \frac{1}{\sum_{j=1}^{n} |K_{i,j}|^{\alpha}} \end{aligned} \tag{19}$$

Iterations($k \geq 0$): $Update \ x^n, y^n, \overline{x}^n \ as \ follows:$

$$\begin{aligned} x^{k+1} &= (I + \tau_k T\partial G)^{-1}(x^k - \tau_k T K^T y^k), \\ y^{k+1} &= (I + \sigma_k \Sigma\partial F^*)(y^k + \sigma_k \Sigma K \overline{x}^k). \\ \theta_k &= \frac{1}{\sqrt{1 + 2\gamma\tau_k}}, \tau_{k+1} = \theta_k \tau_k, \sigma_{k+1} = \frac{\sigma_k}{\tau_n} \\ \overline{x}^{k+1} &= x^{k+1} + \theta_k(x^{k+1} - x^k) \end{aligned} \tag{20}$$

3 TGV和colorize结合

可以考虑将TGV和colorize结合，优化图像效果。具体做了两种尝试:

3.1 利用COLORIZE作为保真项

一种思路是将(7)(8)作为保真项，采用TGV作为正则化，进行问题的求解，如式(21)。下面简称为CSTGV(colorized superpixel total generize variation)。

$$J(u,w) = \alpha_u \sum |\mathbb{D}(\nabla u - w)| + \alpha_w \sum |\xi w| + \sum \frac{1}{2\lambda} \|Au - \alpha_c f\|^2 \quad (21)$$

其中$A = \alpha_c G + I - W'$，通过灰度数据可以计算出来。Au可以通过matlab实现。采用改进的prime-dual算法来实现优化,原始的问题(21)的$Fenchel-Legendre$对偶形式为：

$$\begin{aligned} &\min_{(u,w)\in U\times V} \max_{(p,q,r)\in V\times W\times U} \overbrace{\langle \mathbb{D}(\nabla u - w), p\rangle + \langle \xi w, q\rangle + \langle W'u, r\rangle}^{\langle Kx,y\rangle} \\ &\quad - \underbrace{\left[\langle \alpha_c f, r\rangle + \frac{\lambda}{2}\|r\|^2 + \mathbb{I}_{\{\|\cdot\|_\infty \le \alpha_u\}}(p) + \mathbb{I}_{\{\|\cdot\|_\infty \le \alpha_w\}}(q)\right]}_{F^*(y)} \end{aligned} \quad (22)$$

对应于式(17)式(22)中的$G(x) = 0$,K如下

$$K = \begin{bmatrix} \mathbb{D}\nabla & -\mathbb{D} \\ 0 & \xi \\ A & 0 \end{bmatrix} \quad (23)$$

其中的差分算子∇使用前向差分算子,对m行n列的图像

$$\begin{aligned} \nabla &= \begin{bmatrix} \nabla_x \\ \nabla_y \end{bmatrix} \\ \nabla_x &= I_m \otimes \nabla_f \\ \nabla_y &= \nabla_f \otimes I_m \end{aligned} \quad (24)$$

其中⊗表示Kronecker内积。I_m是$m \times m$维的单位矩阵，∇_f表示带有循环边界的一维差分算子，$\nabla_f \in R^{n\times n}$。

$$\nabla_f = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix} \tag{25}$$

图10和 图11分 别 是$STGV$和$CSTGV$的 优 化 效 果。 用 到 的 参 数 为：

图 10: STGV原始效果

图 11: colorize保真项CSTGV优化效果

$itertimes = 54740, \alpha_w = 5, \alpha_u = 1.2\lambda = 1e-2$(式(21)的$\lambda$和(13)的$\lambda$不同)。对于图像恢复，调节保真项权值$\lambda$能够提升效果。如图13和12，在$itertimes = 54740, \alpha_w = 5, \alpha_u = 1.2, \lambda = 1e-4$参数下，效果明显优于图10和图11

图 12: STGV $\lambda = 1e-4$,耗时4051.26s

图 13: CSTGV $\lambda = 1e-4$,耗时5253s

3.2 利用COLORIZE优化后的结果作为初始值进行STGV优化

STGV收敛速度慢的一个解决手段，可以通过选取合适的初始值来解决。而colorize得到的结果存在比较多的毛刺，可以简单的将两个算法流程串联。以提升效果。需要注意的是，colorize得到的结果(记为g)和原始输入f需要经过相同的归一化因子归一化，否则会导致尺度上的错误。效果见图14

(a) colorize no using seg

(b) colorize using seg

(c) colorize no seg then stgv

(d) colorize using seg then stgv

图 14: 对colorize优化后的图像进行tgv平滑操作

使用参数为$\alpha_u = 1.2, \alpha_w = 4.5, \lambda = 40, itertimes = 2000$

3.3 接下来的工作

研究其他的深度修复算法，同时考虑完善现有算法的一些细节，

- 使用分割信息之后colorize的奇异问题;
- tgv收敛和算法2中参数γ的关系

- c++实现和matlab实现的colorize算法存在局部数据差异
- matlab实现cstgv，并测试效果
- 完善原先代码前向循环边界差分部分

参考文献

[1] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, August 2004.

[2] flyfj. Visiontoolbox. `https://github.com/flyfj/VisionToolbox/blob/master/3D/DepthEnhancer/colorization/fill_depth_colorization.m/` Accessed November 12, 2021.

[3] René Ranftl, Stefan Gehrig, Thomas Pock, and Horst Bischof. Pushing the limits of stereo using variational stereo estimation. In *2012 IEEE Intelligent Vehicles Symposium*, pages 401–407, 2012.

[4] Hans-Hellmut Nagel and Wilfried Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(5):565–593, 1986.

[5] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Ruether, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *2013 IEEE International Conference on Computer Vision*, pages 993–1000, 2013.

[6] Gilad Drozdov, Yevgengy Shapiro, and Guy Gilboa. Robust recovery of heavily degraded depth measurements. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 56–65, 2016.

[7] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, May 2011.

[8] Thomas Pock and Antonin Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *2011 International Conference on Computer Vision*, pages 1762–1769, 2011.