

DIP: Project 1

2017011620 计 73 李家昊

2020 年 4 月 20 日

1 Image Fusion

1.1 方法

根据 Poisson Image Processing [2] 中的 Guided Interpolation 方法, 记插值区域为 Ω , 目标图像的颜色函数为 f^* , 插值后 Ω 区域内颜色函数为 f , 由于可以对 RGB 三通道分别处理, 因此颜色可视作标量, 现给定一个参考梯度场 \mathbf{v} , 希望插值后 Ω 区域内的梯度与参考梯度尽可能一致, 且满足 Ω 的边界值与目标图像相同, 这样就得到一个带有 Dirichlet 边界条件的扩展 Laplace 方程,

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (1)$$

方程 (1) 可离散化如下,

$$\forall p \in \Omega, \quad |N_p|f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq} \quad (2)$$

其中, 对于 Ω 区域内的一个像素位置 p , N_p 为 p 的邻居集合, 包含上下左右四个方向, 对于 p 的某个邻居 $q \in N_p$, v_{pq} 为从 p 到 q 的参考梯度。

对于 Image Fusion 来说, 可以采用 Seamless Cloning 的方法, 将原图的梯度场作为参考梯度场。记原图的颜色函数为 g , 则有 $\mathbf{v} = \nabla g$, 因此 $v_{pq} = g_p - g_q$, 代入方程 (2) 求解即可。

1.2 实现细节

在具体求解过程中, 注意到方程 (2) 是一个 $|\Omega|$ 元线性方程组, 可以将它记为 $\mathbf{Ax} = \mathbf{b}$ 。其中, 线性方程组的系数矩阵为 $\mathbf{A} \in \mathbb{R}^{|\Omega| \times |\Omega|}$, 对于 $\forall p \in \Omega$, 有 $\mathbf{A}(p, p) = 4$, 且 $\mathbf{A}(p, q) = -I_{q \in N_p \cap \Omega}$, 其中 I 为示性函数。事实上, 可以进一步观察到, 矩阵 \mathbf{A} 的每一行最多只有 5 个元素, 因此可采用稀疏矩阵的方式来存储。对于线性方程组的右端项 \mathbf{b} , 它的每一个分量 $\mathbf{b}(p)$ 即为方程 (2) 的右端。

得到系数矩阵 \mathbf{A} 和右端项 \mathbf{b} 后, 可通过迭代方法求解线性方程组, 得到的解 \mathbf{x} 即为融合后在 Ω 区域内的颜色函数 f 。

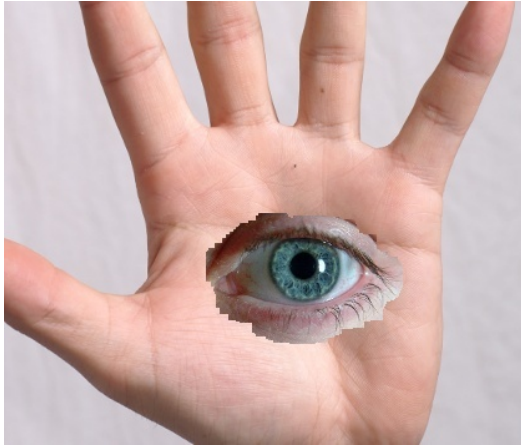


(a) Naive

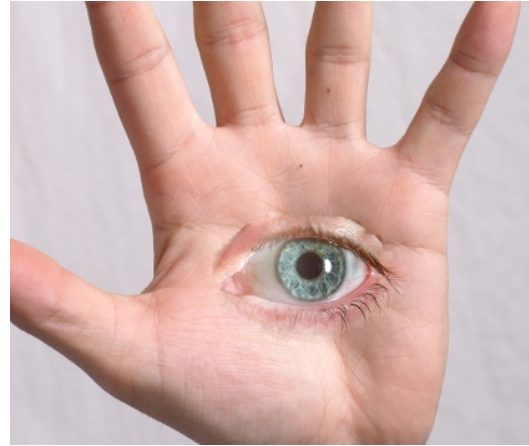


(b) Poisson

图 1: 第一组图片的融合效果



(a) Naive



(b) Poisson

图 2: 第二组图片的融合效果

1.3 实验结果

在给定的两组图片上，每组分别用无缝克隆的 Poisson 方法和直接裁剪拼接的 Naive 方法进行处理，生成的结果如图 1和图 2所示。可以看出，对于 Poisson 方法，图像边缘的过渡更加平滑和自然，同时保留了重要的原图信息。

2 Face Morphing

2.1 方法

在人脸融合任务中，需要根据原图和目标图求出中间状态。解决方法可大致分为形状融合和颜色融合两个步骤，即首先将原图和目标图分别对齐到中间状态，再将它们的颜色融合，这样就得到了最终融合的图像。

首先定义融合率 $\alpha \in (0, 1)$ ，表示将目标图的权重置为 α ，原图的权重置为 $1 - \alpha$ ，将原图和目标图加权平均后就得到中间状态。

对于形状融合，在齐次坐标意义下，首先标记出原图和目标图中对应的特征点 $\{\mathbf{p}_i\}_{i=1}^n$ 和 $\{\mathbf{q}_i\}_{i=1}^n$ ，则中间状态的特征点 $\{\mathbf{r}_i\}_{i=1}^n$ 可由原图和目标图的对应特征点加权平均得到，

$$\mathbf{r}_i = (1 - \alpha)\mathbf{p}_i + \alpha\mathbf{q}_i, \quad i = 1, 2, \dots, n \quad (3)$$

为了将原图对齐到中间状态，首先对中间状态的特征点 $\{\mathbf{r}_i\}_{i=1}^n$ 进行 Delaunay 三角剖分，然后将原图对应的三角形区域仿射变换到中间状态的三角形区域。具体来说，假设 $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ 为原图的一个 Delaunay 三角，它对应的中间状态为 $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ ，则仿射变换矩阵 \mathbf{A} 满足，

$$\mathbf{A}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \quad (4)$$

由此可求出 \mathbf{A}^{-1} 。再根据仿射变换，对于中间状态三角内的每个像素坐标 \mathbf{p} ，可得到它在原图的像素坐标为 $\mathbf{A}^{-1}\mathbf{p}$ ，若不为整数，进行插值即可。

遍历每个 Delaunay 三角，在每个三角中遍历每个像素，即可将原图变换到中间状态。用相同的方法将目标图变换到中间状态，这样就将原图和目标图分别对齐到了中间状态。

对于颜色融合，记对齐到中间状态的原图和目标图分别为 \mathbf{I} 和 \mathbf{J} ，用同样的加权平均方法对整张图片进行 Cross Dissolve 即可，

$$\mathbf{M} = (1 - \alpha)\mathbf{I} + \alpha\mathbf{J} \quad (5)$$

这样就得到了最终融合的图片 \mathbf{M} 。

2.2 实现细节

本实验最大的工作量在于特征点的标注，这里采用机器和人工相结合的方式。其中第一组图片中采用了 Face++ 的人脸检测 API 来标注人脸图像，然后进行了手工修正，并增加了对头顶，发际线，耳朵，衣领等特征点的标记，第二组图片中，由于 Face++ API 无法识别狮子的脸，因此只能全部手工标注。

此外，标注前首先需要统一原图和目标图的大小，仿射变换的插值采用双线性插值，Delaunay 三角剖分使用 `scipy` 库的实现。

2.3 实验结果

在给定的两组图片上，标定的特征点及 Delaunay 三角剖分如图 3和图 5所示，每组图片生成 8 个中间状态，得到的序列如图 4和图 6所示。

两组图片中，除了第一组图片的领结处有少量 artifact 之外，其他位置的过渡均十分自然。

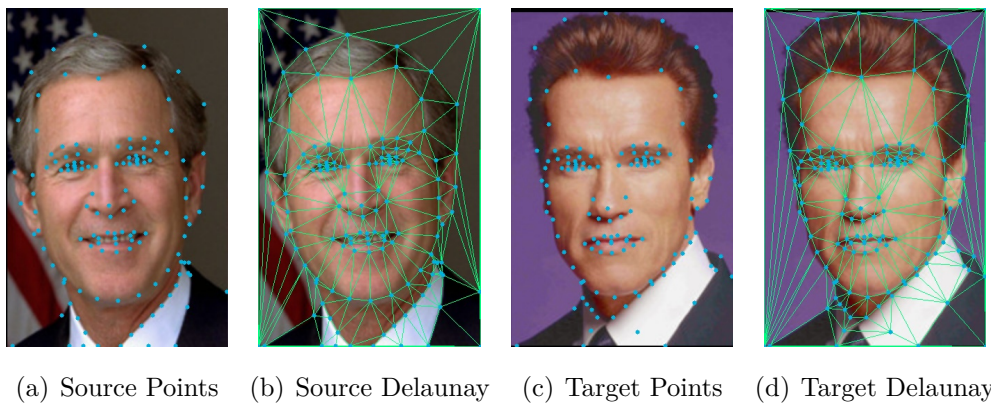


图 3: 第一组图片的特征点及 Delaunay 三角剖分

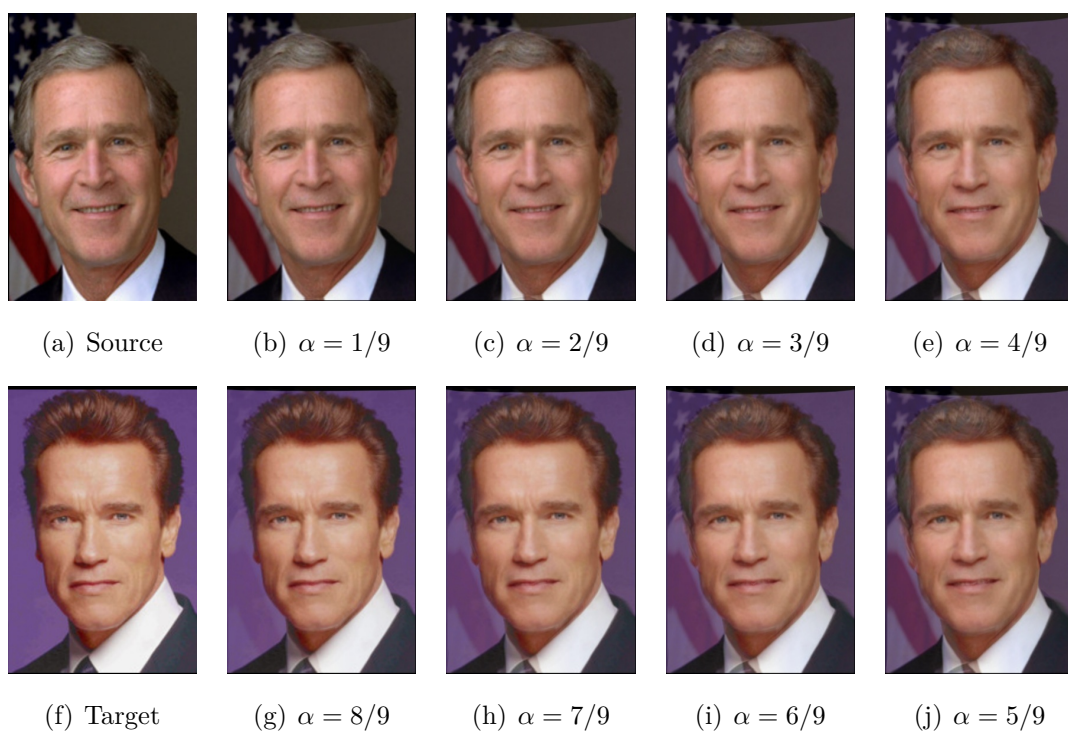


图 4: 在不同的融合率 α 下, 第一组图片的融合效果

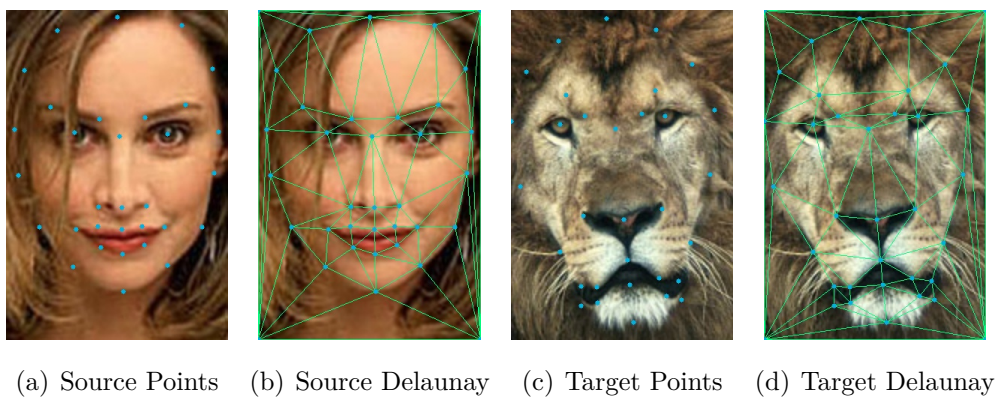


图 5: 第二组图片的特征点及 Delaunay 三角剖分

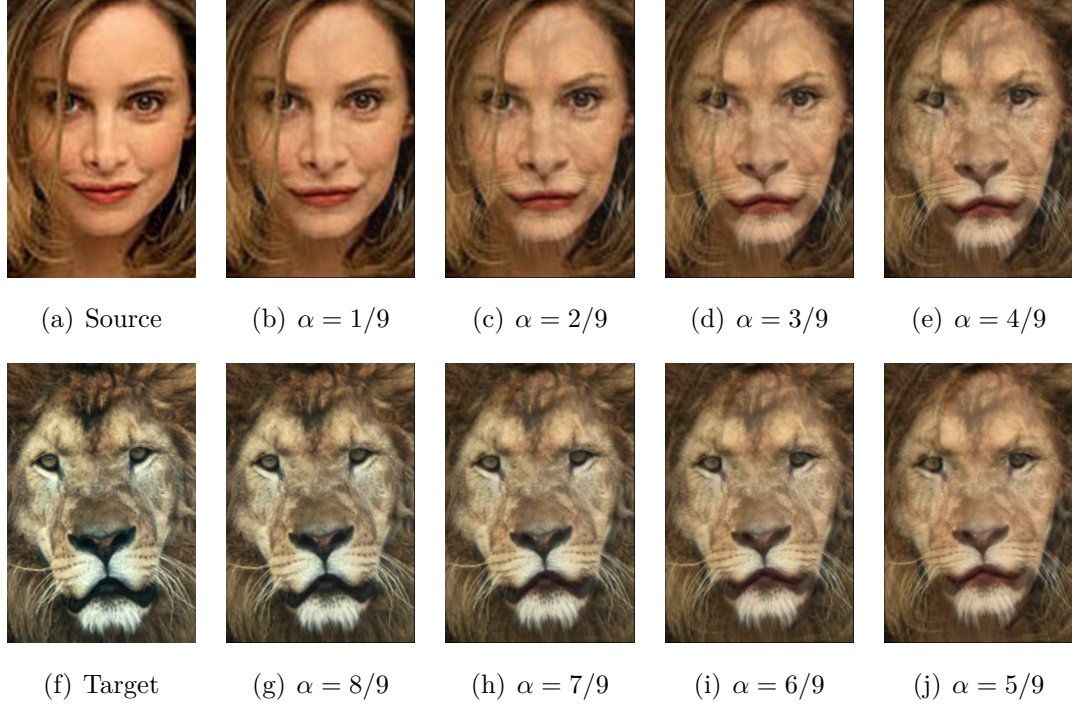


图 6: 在不同的融合率 α 下，第二组图片的融合效果

3 View Morphing

3.1 方法

实现思路主要参考了 View Morphing 的论文 [3]，考虑在不同视角下拍摄的原图和目标图，如果直接进行常规的 morph，会使刚体发生扭曲和变形，不符合常理。为了维持刚体在变换中的刚性，论文提出了 View Morphing 框架，它主要分为三个步骤：首先将原图和目标图 prewarp 到同一平面上，然后进行常规的 morph，最后将融合后的图像 postwarp 到所需视角。

常规的 morph 在 Face Morphing 任务中已经实现了，这里需要解决的问题主要是 prewarp 和 postwarp 的选取。对于 prewarp，这里参考了论文的附录部分，通过求解基本矩阵 \mathbf{F} ，找到极点 \mathbf{e}_0 和 \mathbf{e}_1 ，从而计算出原图和目标图的 prewarp 矩阵 \mathbf{H}_0 和 \mathbf{H}_1 ，然后就可以将原图和目标图对齐到同一平面。

对于 postwarp 矩阵 \mathbf{H}_s ，按照论文对 \mathbf{H}_0 和 \mathbf{H}_1 加权平均的方法，得到的结果并不是太理想，因此这里选取一个 Perspective Warp 矩阵 \mathbf{H}_s ，使得融合后图像的四个角映射到最终图像的四个角，从而得到较为理想的效果。

理论上这三个步骤可以分步执行，但是实际上在 prewarp 或者 morph 的中间状态时，图像内容可能被边界遮挡，导致 postwarp 后图像残缺。因此，这里实现了完全端到端的变换，具体实现方法如下。

首先确定原图和目标图的 prewarp 矩阵 \mathbf{H}_0 和 \mathbf{H}_1 ，将原图和目标图的对应点分别进行 prewarp，沿用 Face Morphing 小节的符号，即得到 $\{\mathbf{H}_0 \mathbf{p}_i\}_{i=1}^n$ 和

$\{\mathbf{H}_1 \mathbf{q}_i\}_{i=1}^n$ ，同样求出中间状态，

$$\mathbf{r}_i = (1 - \alpha)\mathbf{H}_0 \mathbf{p}_i + \alpha \mathbf{H}_1 \mathbf{q}_i, \quad i = 1, 2, \dots, n \quad (6)$$

对中间状态的特征点 $\{\mathbf{r}_i\}_{i=1}^n$ 进行 Delaunay 三角剖分，对于某个 Delaunay 三角，记其从 prewarp 后的原图映射到中间状态的仿射变换为 \mathbf{A} ，然后用 \mathbf{H}_s 将中间状态的三角区域 postwarp 到最终状态，对于最终状态三角内一点 \mathbf{x} ，可以计算出其对应的原图坐标，即为 $\mathbf{H}_0^{-1} \mathbf{A}^{-1} \mathbf{H}_s^{-1} \mathbf{x}$ 。这样就避免了引入中间状态，实现了端到端的变换。

3.2 实现细节

对于特征点的标注，这里采用 dlib 和人工结合的方式，其中 dlib 生成的面部特征点用于计算基本矩阵 \mathbf{F} ，并由此得到 prewarp 矩阵，其余的人工标注点连同 dlib 生成的点用于 morph 过程中的 Delaunay 三角剖分。

对于基本矩阵的求解，这里采用八点法，具体实现方法可参考维基百科 [1]。对于 prewarp 的选取，这里省略了论文中 \mathbf{H}_1 的平移和放缩矩阵 \mathbf{T} ，原因是当 \mathbf{F} 的计算不准确时， \mathbf{T} 矩阵可能是奇异的。

3.3 实验结果

首先给出分步执行的结果，以第一组图片为例，当融合率 $\alpha = 0.5$ 时，prewarp 后的原图和目标图，融合图像，以及 postwarp 后的融合图像如图 7，可以看出，prewarp 导致原图的大量信息丢失，目标图被边界截断，导致 morph 后的图像出现裂缝，postwarp 后出现残缺。



图 7: prewarp 后的原图和目标图，postwarp 前后的融合图像

为了弥补这一缺陷，这里实现了端到端的 View Morphing，在给定的两组图片上，实验结果如图 8 和图 9，可以看到，生成的中间序列实现了视角的转换，且内容十分完整，没有出现残缺现象，仅有少量的 artifact。

此外，这里还补充了两组图片，实验结果如图 10 和图 11，其中原图和目标图是通过镜面对称生成的，因此 prewarp 的计算结果更加准确，过渡也更自然。

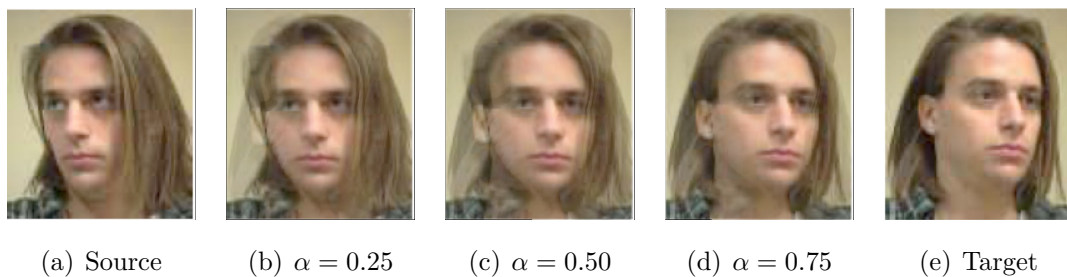


图 8: 在不同的融合率 α 下, 第一组图片的融合效果

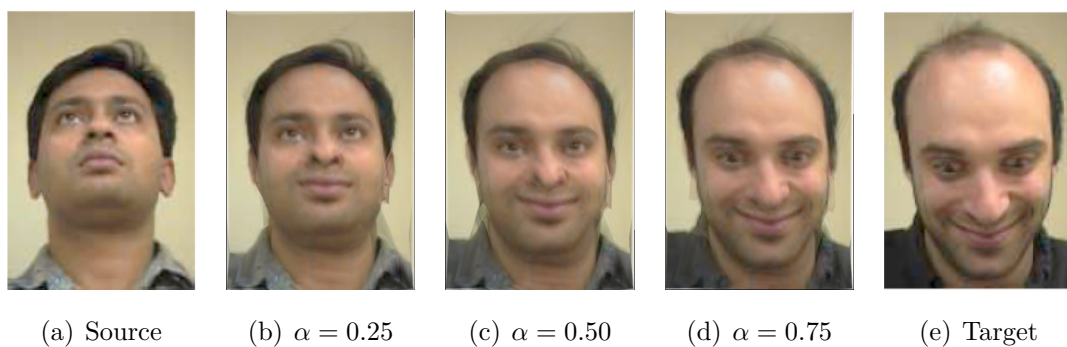


图 9: 在不同的融合率 α 下, 第二组图片的融合效果

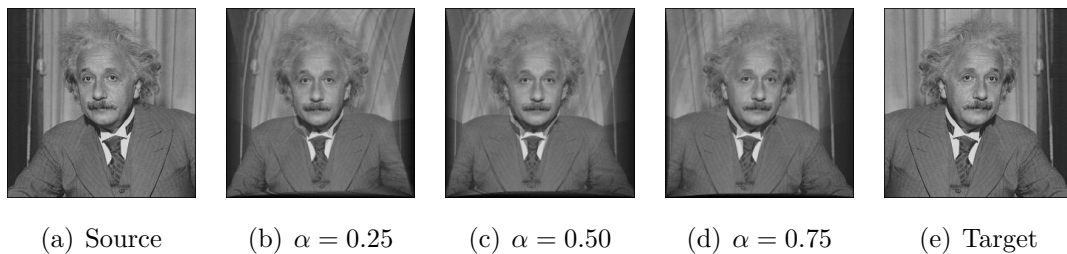


图 10: 在不同的融合率 α 下, 第三组图片的融合效果

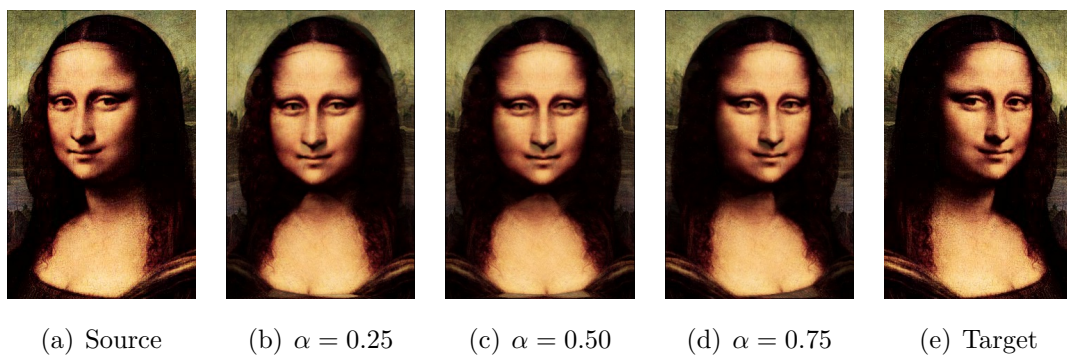


图 11: 在不同的融合率 α 下, 第四组图片的融合效果

参考文献

- [1] Eight-point algorithm. https://en.wikipedia.org/wiki/Eight-point_algorithm. Accessed: 2020-04-20.
- [2] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003.
- [3] S. M. Seitz and C. R. Dyer. View morphing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 21–30, 1996.