APPENDIX I

TIME COMPLEXITY ANALYSES OF ALL CONSIDERED METHODS

For easy exposition, we analyze the complexity of the methods in the following order: HM, MHM, IMIHM, AGL, GP, JBI, AJBI, CMCC, our algorithm, and our modified algorithm.

A. For HM, MHM and IMIHM

the HM and MHM methods, once the HE mapping function has been constructed in $O(|\Omega_t|)$ time using all target pixels in Ω_t and all source pixels in Ω_s, it can be reused for all target pixels in I_t. For the HM method, it takes $O(|I_t|)$ time to complete the overall color correction task because the color of each target pixel can be corrected in $O(1)$ time using the precomputed HE mapping function.

As mentioned before, the main difference between HM and MHM is that for the MHM method, when the color of each target pixel p_t corresponds to a nearly vertical or horizontal segment in the HM mapping function, the color of p_t is corrected by randomly sampling from a segment defined by a uniform distribution. Therefore, for MHM, it also takes $O(|I_t|)$ time to complete the overall color correction task.

In the IMIHM method, the number of iterations and the overlapping rate between two consecutive sliding windows are set to 9 and 0.8, respectively. At the first iteration, using the sliding window approach and the HM mapping function, it takes $O(|\Omega_t|)$ time to perform color correction on the target pixels within the window. At the second iteration, using the sliding window approach with overlapping rate 0.8 to build up a new HE mapping function, it takes $O(|\Omega_t|)$ time to correct color for the target pixels in the newly moved window. In terms of the worst case consideration, for IMIHM, it also takes $O(|I_t|)$ time to complete the overall color correction task.

B. For AGL

AGL method consists of a global least squares optimization stage and a local compensation stage. In the global stage, the mean and standard deviation of the overlapping area are computed in $O(|\Omega_t|)$ time. A set of linear equations is then constructed and solved via a least squares approach, which requires $O(n^3)$ time, where n denotes the number of variables associated with all image pairs. For instance, for a set of 10 images, n equals 18 $(= 2 \times 9)$. In the local stage, each target image I_t is divided into a set of equal-sized grids, and the color statistics within each grid are computed in $O(|G|)$ time, where $|G|$ denotes the grid size. Each target pixel in the grid is then adjusted using bilinear interpolation and adaptive gamma correction, both of which operate in $O(1)$ time, resulting in an overall complexity of $O(|I_t|)$ to refine the color of each target image I_t in the local stage. Therefore, the total time complexity of AGL is bounded by $O(|I_t|)$ to complete the color correct task for I_t.

C. For GP

In the GP method, first, it takes $O(|\Omega_t|)$ time to construct the HE mapping function using the source and target pixels in the overlapping area. Next, six equidistant anchor points are selected from the HE mapping function. Then, it takes $O(|I_t|)$ time to calculate a few gradient statistics of the target image I_t. Based on the selected six anchor points and the constraints along with the calculated gradient statistics, a spline function is determined in $O(1)$ time using the convex quadratic programming technique. Finally, using the determined the spline function, the color correction of all target pixels in I_t can be completed in $O(|I_t|)$ time. Therefore, for GP, its time complexity is bounded by $O(|I_t|)$.

D. For JBI and AJBI

In the JBI method, the color differences along the stitching line L in the overlapping area are fully utilized to correct the color of each target pixel. Using the JBI technique, it takes $O(|L|)$ time to correct color for each target pixel in I_t. Overall, for JBI, it takes $O(|L||I_t|)$ time to complete the color correction task for I_t. In the AJBI method, considering the color differences of an appropriate stitching line interval, it also takes $O(|L|)$ time to correct color for each target pixel. Overall, for AJBI, its time complexity is bounded by $O(|L||I_t|)$ time to complete the color correction task.

E. For CMCC

In the CMCC method, using the 4:2:0(L) downsampling scheme, and noting that $|I_t| = |I_s|$ as stated in Table I, it takes $O(|I_t|)$ time to downsample the target image I_t and the source image I_s, obtaining the downsampled target and source images, I_t^d and I_s^d, respectively. Using the dehazing method [30], it takes $O(|I_t^d|)$ time to obtain the enhanced version of I_t^d, denoted by I'^d_t. Then, the low frequency component of I'^d_t is replaced by that of I_s^d, denoted by I''^d_t. Furthermore, it takes $O(|I_t|)$ time to upsample I''^d_t to obtain the upsampled image I''_t. Based on a grid approach, using the mean compensation technique between each grid in I_t and the corresponding grid in I''_t, the color of each target pixel in the grid of I_t can be corrected in $O(1)$ time. Overall, for CMCC, it takes$O(|I_t|)$ time to complete the color correction task.

F. For Our Algorithm and Our Modified Algorithm

According to Theorem 1, the time complexity of our algorithm is $O(|I_t||\Omega_t|)$. Using our modified algorithm, we show that the time complexity can be reduced to $O(|I_t|)$ time, while preserving the color correction quality of the original version.

As described in Subsection III-A, the inlier correspondence set in the overlapping area is defined as $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m\}$ with $m \leq |\Omega_t|$. Each inlier correspondence $\boldsymbol{c_i}$ is transformed into a single point p_i by aligning the source and target feature points. Next, based on the m points p_1, p_2, ..., and p_m, the Delaunay triangulation subroutine "getTriangleList" [39] is applied to produce a set of triangles in $O(|\Omega_t|)$ time. To correct each target pixel p_t in Ω_t, we first identify the closest triangle that contains p_t, and it can be done in $O(1)$ time. For p_t, let the three corner points of the searched triangle be denoted by p'_1, p'_2, and p'_3 corresponding to the three correspondences c'_1, c'_2, and c'_3, respectively. Based on the color differences of c'_1, c'_2, and c'_3, by Eqs. (2)–(3), the JBI-based color correction term for pt in Ω_t can be derived in $\mathrm{O}(1)$ time. Therefore, the EC-based fusion method can correct the color of p_t in $\mathrm{O}(1)$ time. As a result, correcting color for all target pixels in Ω_t can be done in $\mathrm{O}(|\Omega_t|)$ time.

After correcting color for all target pixels in Ω_t, we further present the modified BRI-based fusion method to correct color for target pixels in $I_t \backslash \Omega_t$. As an initial ripple point, each target pixel p_i on the boundary B propagates its color difference $D(p_i)$ (see Eq. (12)) and its position (x, y) forward to the neighboring target pixels in $I_t \backslash \Omega_t$ until all propagated ripples of $p_i(x, y)$ touch the fifth ripple instead of touching the boundary of the non-overlapping area. For all target points in the propagated five ripples, the BRI-based fusion method (see Eq. (17)) is applied to correct color for these target pixels, and it takes $\mathrm{O}(|\boldsymbol{B}|)$ time. Otherwise, for the remaining target points in the non-overlapping area, the HE method is applied to correct color for these target pixels directly, and it takes $\mathrm{O}(|I_t| - |\Omega_t| - 5|\boldsymbol{B}|)$ time. As a result, the color correction task for target pixels in $I_t \backslash \Omega_t$ can be done in $\mathrm{O}(|I_t|)$ time. Combining the time complexities spent in correcting color for target pixels in Ω_t and $I_t \backslash \Omega_t$, the total time complexity of our modified algorithm is $\mathrm{O}(|I_t|)$ time.