Gradient Domain Blending
Notation

- **Source image**: $g$
  - the image we're cutting out and pasting
- **Target image**: $f'$
  - the image we're changing
- **Blended image**: $f$
  - Final blending result
- **Replacement pixels**: $\Omega$
  - the pixels in target image that will be blended with source image
- **Neighbor pixels**: $N_p$
  - the pixel $p$'s neighbor pixels
Formula

• Apply source image gradient to target image

\[
\min_{f|\Omega} \sum_{p \in \Omega} \left( \left| N_p |f_p - \sum_{q \in N_p} f_q \right| \right)^2 - \left( \left| N_p |g_p - \sum_{q \in N_p} g_q \right| \right), \quad \text{with} \quad f|_{\partial \Omega} = f^*|_{\partial \Omega}
\]

• Convert to linear system

\[
| N_p |f_p - \sum_{q \in N_p} f_q = | N_p |g_p - \sum_{q \in N_p} g_q
\]
Adjust Image and Create Mask

• Manually adjust source image and offset

• Keep the resized source image and offset
Adjust Image and Create Mask

• Create source image mask using resized source image

• Use matlab function imfreehand and createMask
Index Pixels

• Index the replacement pixels
Compute equation

• Solve $Ax=b$

$$|N_p|f_p - \sum_{q \in N_p} f_q = |N_p|g_p - \sum_{q \in N_p} g_q$$

• $x$: $f$
• $A$: coefficient matrix
• $b$: solution vector

• Typically we only consider the $N_p=4$
  • The up, down, left and right neighbors
• $f_p=f'_p$, when $p$ is not replacement pixels
Compute equation

• For example

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 \\
1 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 1 & 2 & 0 \\
3 & 0 & 3 & 4 & 0 & 0 \\
4 & 0 & 0 & 5 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{bmatrix}
4 & -1 & 0 & -1 & 0 \\
-1 & 4 & 0 & 0 & 0 \\
0 & 0 & 4 & -1 & 0 \\
-1 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & -1 & 4 \\
\end{bmatrix}
\begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
\end{bmatrix}
=
\begin{bmatrix}
\Delta g_1 + f'_{3,1} + f'_{2,2} \\
\Delta g_2 + f'_{4,1} + f'_{5,2} + f'_{4,3} \\
\Delta g_3 + f'_{2,2} + f'_{1,3} + f'_{2,4} \\
\Delta g_4 + f'_{4,3} \\
\Delta g_5 + f'_{2,4} + f'_{4,4} + f'_{3,5} \\
\end{bmatrix}
\]

• \( \Delta g_p = 4g_p - \sum_{q \in N_p} g_q \)
Solve equation

• Use matlab function mldivide
• Use sparse matrix to represent coefficient matrix A