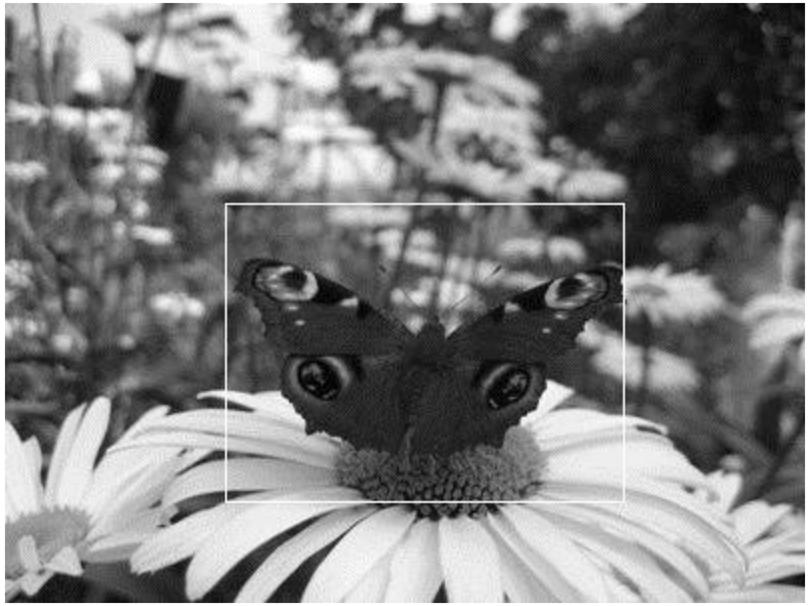
Image Alignment Algorithm

Ke Xu
Northeastern University

Goal $\min \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]^2$



$I(\mathbf{x})$

$\mathbf{W}(\mathbf{x};\mathbf{p})$

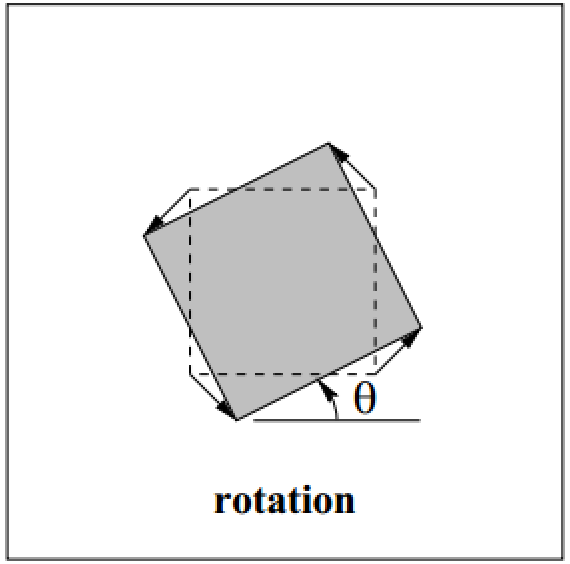
$T(\mathbf{x})$

- **Forward Additive Algorithm**
- **Forward Compositional Algorithm**
- **Inverse Additive Algorithm**
- **Inverse Compositional Algorithm**

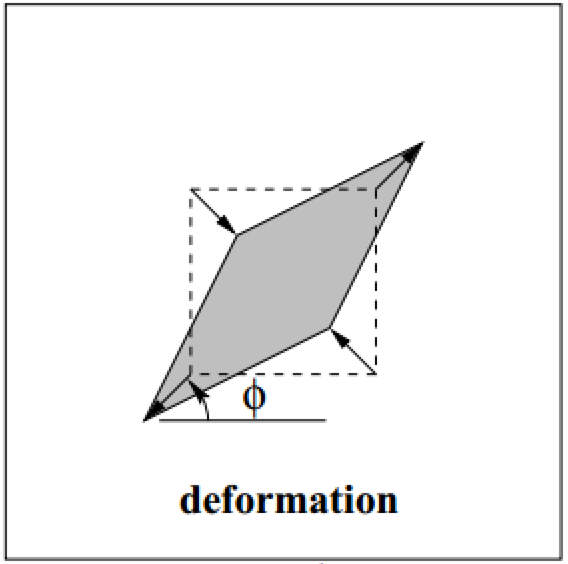
Model of Warps

- **2D Affine Translation (6-Dof)**

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



a



b

We donate the warp as:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) = \begin{pmatrix} (1+p_1)\cdot x & + & p_2 \cdot y & + & p_5 \\ p_3 \cdot x & + & (1+p_4)\cdot y & + & p_6 \end{pmatrix}$$

$$= \begin{pmatrix} 1+p_1 & p_2 & p_5 \\ p_3 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

An affinity = A Rotation + A Scaling in orthogonal directions[(1)]

$$\mathbf{p} = (p_1, p_2, p_3, p_4, p_5, p_6)^T$$

(1) MVG 2nd P39

Forward Additive Algorithm

- **Goal: Minimize the following expression**

$$\boxed{\sum_{\mathbf{x}}\left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}+\Delta\mathbf{p}\right)\right)-T\left(\mathbf{x}\right)\right]^2} \qquad \textbf{(1.1)}$$

$$\textbf{Do} \quad \mathbf{p}\leftarrow\mathbf{p}+\Delta\mathbf{p} \quad \textbf{Until} \quad \left\|\Delta\mathbf{p}\right\|\le\xi$$

- **Perform a first order Taylor Expansion on Eq. (1.1) at (x; p)**

Steepest Descent Images

$$\Rightarrow \sum_{\mathbf{x}}\left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)+\boxed{\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}}\Delta\mathbf{p}-T\left(\mathbf{x}\right)\right]^2 \qquad \textbf{(1.2)}$$

$$\frac{\partial I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)}{\partial\mathbf{p}}=\left.\frac{\partial I\left(\mathbf{u}\right)}{\partial\mathbf{u}}\right|_{\mathbf{u}=\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)}\cdot\left.\frac{\partial\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}}$$

Jacobian

NOTE: $\nabla I(\mathrm{x}) = \left(\frac{\partial I}{\partial \mathrm{x}}, \frac{\partial I}{\partial \mathrm{y}}\right)$, is evaluated at $\mathrm{W}(\mathrm{x};\mathrm{p})$. For $\frac{\partial I(\mathrm{u})}{\partial \mathrm{u}}$ is the gradient of $I(\mathrm{u})$, $\mathrm{u} = \mathrm{W}(\mathrm{x};\mathrm{p})$. As we perform the algorithm over the pixels x in the template image T , so compute the coordinates of the correspondence pixels u in image I, then get the gradient of u. And the gradient that we get is $\nabla I(\mathrm{x})$.

Considering the ***Jacobian*** of the warp:

$$\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)=\left(\mathbf{W}_x\left(\mathbf{x};\mathbf{p}\right),\mathbf{W}_y\left(\mathbf{x};\mathbf{p}\right)\right)^T$$

$$\frac{\partial\mathbf{W}}{\partial\mathbf{p}}=\begin{pmatrix}\frac{\partial\mathbf{W}_x}{\partial p_1} & \frac{\partial\mathbf{W}_x}{\partial p_2} & \cdots & \frac{\partial\mathbf{W}_x}{\partial p_n}\\ \frac{\partial\mathbf{W}_y}{\partial p_1} & \frac{\partial\mathbf{W}_y}{\partial p_2} & \cdots & \frac{\partial\mathbf{W}_y}{\partial p_n}\end{pmatrix}$$

The ***Jacobian*** of our model shows as follow, the x and y are the coordinates of warped image I. However the Jacobian is not the function of parameter p in our model.

$$\frac{\partial\mathbf{W}}{\partial\mathbf{p}}=\begin{pmatrix}x & y & 0 & 0 & 1 & 0\\ 0 & 0 & x & y & 0 & 1\end{pmatrix}$$

Forward Additive Algorithm

- **The partial derivative of Eq. (1.2) with respect to *Δp***

$$\sum_{\mathbf{x}}\left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)+\underbrace{\nabla I}_{1\times 2}\underbrace{\frac{\partial \mathbf{W}}{\partial \mathbf{p}}}_{2\times 6}\underbrace{\Delta\mathbf{p}}_{6\times 1}-T\left(\mathbf{x}\right)\right]^2 \quad \textbf{(1.2)}$$

$$2\sum_{\mathbf{x}}\left[\nabla I\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T\left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)+\nabla I\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\Delta\mathbf{p}-T\left(\mathbf{x}\right)\right] \quad \textbf{(1.3)}$$

Note: $f\left(\mathbf{x}\right)=\boldsymbol{\alpha}^T\mathbf{x}=\mathbf{x}^T\boldsymbol{\alpha} \quad \frac{df}{d\mathbf{x}}=\boldsymbol{\alpha}$

- **Set Eq. (1.3) to zero then get *Δp***

$$\Delta\mathbf{p}=H^{-1}\sum_{\mathbf{x}}\left[\nabla I\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T\left[T\left(\mathbf{x}\right)-I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)\right] \quad \textbf{(1.4)}$$

Where the H is the n×n Hessian matrix:

$$H=\sum_{\mathbf{x}}\left[\nabla I\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T\left[\nabla I\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right] \quad \textbf{(1.5)}$$

- **In each iteration we update the warp by *Δp***

$$\mathbf{p}\leftarrow\mathbf{p}+\Delta\mathbf{p}$$

We can just update the **p** by *Δ***p** additively.

Forward Additive Algorithm

The Lucas-Kanade Algorithm(1981)

Iterate:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
2. Compute the error image $T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
3. Warp the gradient of image I to compute ∇I
4. Evaluate the Jacobian $\frac{\partial \mathrm{W}}{\partial \mathrm{p}}$ at $\mathbf{W}(\mathbf{x};\mathbf{p})$
5. Compute the steepest descent images $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
6. Compute the Hessian matrix $H = \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]$
7. Compute $\sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
8. Compute $\Delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T \left[T(\mathbf{x}) - I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right)\right]$
9. Update the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$

until $\|\Delta \mathrm{p}\| \leq \xi$

- The warped gradient ∇I, Jacobian $\frac{\partial W}{\partial p}$, Hessian Matrix H is depended on **p**. So all this steps must be performed in each iteration.
- However we can per-compute the gradient of image I. And it is accessible when we compute ∇I.

Forward Compositional Algorithm

- Goal: Minimize the following expression

$$\sum_{\mathbf{x}} \left[I\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p}\right)\right) - T(\mathbf{x}) \right]^2 \qquad \textbf{(2.1)}$$

Do $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ **Until** $\|\Delta\mathbf{p}\| \le \xi$

- **Perform a first order Taylor Expansion on Eq. (2.1) at $(\mathbf{x};\mathbf{0})$**

$$\sum_{\mathbf{x}} \left[I\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x};0);\mathbf{p}\right)\right) + \nabla I(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 \qquad \textbf{(2.2)}$$

Steepest Descent Images

Considering the chain-rule, we can get:

Jacobian

$$\frac{\partial I\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x};\mathbf{q});\mathbf{p}\right)\right)}{\partial \mathbf{q}} = \left.\frac{\partial I\left(\mathbf{W}(\mathbf{u};\mathbf{p})\right)}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{W}(\mathbf{x};0)} \cdot \left.\frac{\partial \mathbf{W}(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}}\right|_{\mathbf{p}=0}$$

To get the gradient $\nabla I(\mathbf{W}(\mathbf{x};\mathbf{p}))$ in this expression, we can first wrap image I to get $I(\mathbf{W})$, then compute the gradient of $I(\mathbf{W})$.

- **We assume $\mathbf{W}(\mathbf{x};0) = \mathbf{x}$, then simplify Eq. (2.2) as:**

$$\sum_{\mathbf{x}} \left[I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right) + \nabla I(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 \qquad \textbf{(2.4)}$$

There are two differences between Eq. (2.4) and Eq.(1.2):

- **The gradient image**. In Eq. (2.4), the gradient of ∇I is replaced with the gradient of $\nabla I(\mathbf{W})$. And both gradient images are evaluated at $\mathrm{W}(\mathrm{x};\mathrm{p})$ of the warped image $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$.
- **The *Jacobian* matrix**. In Eq. (1.2), the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is evaluated at $(\mathbf{x};\mathbf{p})$, while in Eq. (2.4) it is evaluated at $(\mathbf{x};\mathbf{0})$.

Forward Compositional Algorithm

The Shum-Szeliski Algorithm(2000)

Per-compute:

4. Evaluate the Jacobian $\frac{\partial \mathrm{W}}{\partial \mathrm{p}}$ at $(\mathbf{x}; 0)$

Iterate:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
2. Compute the error image $T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
3. Compute the gradient $\nabla I(\mathbf{W})$ of image $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
5. Compute the steepest descent images $\nabla I(\mathrm{W})\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
6. Compute the Hessian matrix $H = \sum_{\mathbf{x}}\left[\nabla I(\mathrm{W})\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T\left[\nabla I(\mathrm{W})\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]$
7. Compute $\sum_{\mathbf{x}}\left[\nabla I(\mathrm{W})\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T[T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
8. Compute $\Delta\mathbf{p} = H^{-1}\sum_{\mathbf{x}}\left[\nabla I(\mathrm{W})\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T[T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
9. Update the parameters $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})$

until $||\Delta \mathrm{p}|| \leq \xi$

- **According to the Eq. (2.4), change the ∇I with $\nabla I(\mathbf{W})$ in following steps (6~8).**
- **In each iteration we update the warp by Δp. In forward compositional algorithm the update step can be perform as:**

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}) = \mathbf{W}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p})$$

$$\mathbf{W}(\mathbf{x};\mathbf{p}) = \begin{pmatrix} 1+p_1 & p_2 & p_5 \\ p_3 & 1+p_4 & p_6 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1+\Delta p_1 & \Delta p_2 & \Delta p_5 \\ \Delta p_3 & 1+\Delta p_4 & \Delta p_6 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Affine Matrix **Updating Affine Matrix**

Inverse Compositional Algorithm

- **Goal: Minimize the following expression**

$$\boxed{\sum_{\mathbf{x}}\left[T\left(\mathbf{W}\left(\mathbf{x};\Delta\mathbf{p}\right)\right)-I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)\right]^2} \qquad \textbf{(3.1)}$$

Do $\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\leftarrow \mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\circ \mathbf{W}\left(\mathbf{x};\Delta\mathbf{p}\right)^{-1}$ **Until** $\left\|\Delta\mathbf{p}\right\|\leq\xi$

- **Perform a first order Taylor Expansion on Eq. (3.1) at (x; 0)**

$$\Rightarrow \sum_{\mathbf{x}}\left[T\left(\mathbf{W}\left(\mathbf{x};0\right)\right)+\nabla T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p}-I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)\right]^2 \qquad \textbf{(3.2)}$$

Considering the chain-rule, we can get:

$$\frac{\partial T\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)}{\partial\mathbf{p}}=\left.\frac{\partial T\left(\mathbf{u}\right)}{\partial\mathbf{u}}\right|_{\mathbf{u}=\mathbf{W}\left(\mathbf{x};0\right)}\cdot\underbrace{\left.\frac{\partial\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)}{\partial\mathbf{p}}\right|_{\mathbf{p}=0}}_{\textit{Jacobian}}$$

- **We assume $\mathbf{W}(\mathbf{x};0)=\mathbf{x}$, then simplify Eq. (3.2) as:**

$$\sum_{\mathbf{x}}\left[T\left(\mathbf{x}\right)+\nabla T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p}-I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)\right]^2 \qquad \textbf{(3.4)}$$

- **The solve of the expression is:**

$$\Delta\mathbf{p}=H^{-1}\sum_{\mathbf{x}}\left[\nabla T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^T\left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)-T\left(\mathbf{x}\right)\right] \qquad \textbf{(3.5)}$$

- **The Hessian matrix is changed:**

$$H=\sum_{\mathbf{x}}\left[\nabla T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^T\left[\nabla T\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right] \qquad \textbf{(3.6)}$$

The ∇T is the gradient of the template image T, the ***Jacobian*** is evaluated at $(\mathbf{x};\mathbf{0})$, and H is not depends on parameter $\mathbf{p}$.

Inverse Compositional Algorithm

The Baker-Matthews Algorithm(2001)

Per-compute:

3. Evaluate the gradient ∇T of image $T(\mathbf{x})$
4. Evaluate the Jacobian $\frac{\partial \mathrm{W}}{\partial \mathrm{p}}$ at $(\mathbf{x};0)$
5. Compute the steepest descent images $\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
6. Compute the Hessian matrix $H = \sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]$

Iterate:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
2. Compute the error image $I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})$
7. Compute $\sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]$
8. Compute $\Delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T \left[I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right) - T(\mathbf{x})\right]$
9. Update the parameters $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$

until $\|\Delta \mathrm{p}\| \leq \xi$

- **In each iteration we update the warp by Δp.**

$$\mathbf{W}(\mathbf{x};\mathbf{p})$$

$$= \begin{pmatrix} 1+p_1 & p_2 & p_5 \\ p_3 & 1+p_4 & p_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1+\Delta p_1 & \Delta p_2 & \Delta p_5 \\ \Delta p_3 & 1+\Delta p_4 & \Delta p_6 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Forward Additive Algorithm

- **Goal: the same as additive algorithm initially**

$$\boxed{\sum_{\mathbf{x}}\left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}+\Delta\mathbf{p}\right)\right)-T\left(\mathbf{x}\right)\right]^2} \quad \mathbf{(4.1)}$$

Do $\mathbf{p}\leftarrow\mathbf{p}+\Delta\mathbf{p}$ **Until** $\left\|\Delta\mathbf{p}\right\|\le\xi$

- **Perform a first order Taylor Expansion on Eq. (4.1) at (x; p)**

$$\sum_{\mathbf{x}}\left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)+\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p}-T\left(\mathbf{x}\right)\right]^2 \quad \mathbf{(4.2)}$$

In Hager and Belhumeur (1998) it is assumed that the current estimates of the parameters are approximately correct: i.e.

$$I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)\approx T\left(\mathbf{x}\right) \quad \mathbf{(4.3)}$$

Taking partial derivatives with respect to **x** and using the chain rule gives:

$$\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{x}}\approx\nabla T \quad \mathbf{(4.4)}$$

- **Inverting $\frac{\partial\mathbf{W}}{\partial\mathbf{x}}$ and substituting Eq. (4.4) into Eq. (4.2) gives:**

$$\sum_{\mathbf{x}}\left[I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)+\nabla T\left(\frac{\partial\mathbf{W}}{\partial\mathbf{x}}\right)^{-1}\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p}-T\left(\mathbf{x}\right)\right]^2 \quad \mathbf{(4.5)}$$

- **To completely change the role of the template and the image *I* , we replace *Δ*p with *Δ*p. The final goal is then:**

$$\boxed{\sum_{\mathbf{x}}\left[T\left(\mathbf{x}\right)+\nabla T\left(\frac{\partial\mathbf{W}}{\partial\mathbf{x}}\right)^{-1}\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p}-I\left(\mathbf{W}\left(\mathbf{x};\mathbf{p}\right)\right)\right]^2} \quad \mathbf{(4.6)}$$

update the parameters by:

$$\mathbf{p}\leftarrow\mathbf{p}-\Delta\mathbf{p}$$

Forward Additive Algorithm

- **Considering there are two parts of Jacobians in Eq. (4.6), assumed that the product of the two Jacobians can be written as:**

$$\left(\frac{\partial \mathbf{W}}{\partial \mathbf{x}}\right)^{-1}\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \Gamma(\mathbf{x})\Sigma(\mathbf{p}) \qquad \textbf{(4.7)}$$

where $\Gamma(\mathbf{x})$ is a $2 \times k$ matrix that is only depends on $\mathbf{x}$, and the $\Sigma(\mathbf{p})$ is a $k \times$ n matrix that is only depends on $\mathbf{p}$. However, not all warps can be written in this way. As for the affine warp we discussed:

$$\left(\frac{\partial \mathbf{W}}{\partial \mathbf{x}}\right)^{-1} = \begin{pmatrix} 1+p_1 & p_2 \\ p_3 & 1+p_4 \end{pmatrix}^{-1} = \frac{1}{\det}\begin{pmatrix} 1+p_4 & -p_2 \\ -p_3 & 1+p_1 \end{pmatrix}$$

then:

$$\left(\frac{\partial \mathbf{W}}{\partial \mathbf{x}}\right)^{-1}\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \frac{1}{(1+p_1)\cdot(1+p_4)-p_2\cdot p_3} \times \begin{pmatrix} 1+p_4 & -p_2 \\ -p_3 & 1+p_1 \end{pmatrix}\begin{pmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{pmatrix}$$

It can be written as:

$$\Gamma(\mathbf{x})\Sigma(\mathbf{p}) = \frac{1}{\det}\begin{pmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1+p_4 & 0 & -p_2 & 0 & 0 & 0 \\ 0 & 1+p_4 & 0 & -p_2 & 0 & 0 \\ -p_3 & 0 & 1+p_4 & 0 & 0 & 0 \\ 0 & -p_3 & 0 & 1+p_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+p_4 & -p_2 \\ 0 & 0 & 0 & 0 & -p_3 & 1+p_4 \end{pmatrix}$$

- **Two Jacobians has therefore been written in the form of Eq. (4.7). Then Eq. (4.6) can be re-written as:**

$$\sum_{\mathbf{x}}\left[T(\mathbf{x}) + \nabla T\Gamma(\mathbf{x})\Sigma(\mathbf{p})\Delta\mathbf{p} - I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right)\right]^2 \qquad \textbf{(4.8)}$$

Forward Additive Algorithm

- **The Eq. (4.8) has the closed from solution:**

$$\Delta \mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla T \Gamma(\mathbf{x}) \Sigma(\mathbf{p}) \right]^T \left[I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right) - T(\mathbf{x}) \right] \tag{4.9}$$

- **The Hessian matrix is:**

$$H = \sum_{\mathbf{x}} \left[\nabla T \Gamma(\mathbf{x}) \Sigma(\mathbf{p}) \right]^T \left[\nabla T \Gamma(\mathbf{x}) \Sigma(\mathbf{p}) \right] \tag{4.10}$$

The $\Sigma(\mathbf{p})$ does not depend on $\mathbf{x}$, the Hessian matrix can re-written as:

$$H = \Sigma(\mathbf{p})^T \sum_{\mathbf{x}} \left[\nabla T \Gamma(\mathbf{x}) \right]^T \left[\nabla T \Gamma(\mathbf{x}) \right] \Sigma(\mathbf{p})$$

- **Denoting:**

$$H_* = \sum_{\mathbf{x}} \left[\nabla T \Gamma(\mathbf{x}) \right]^T \left[\boxed{\nabla T \Gamma(\mathbf{x})} \right] \tag{4.11}$$

Steepest Descent Images

If $\Sigma(\mathbf{p})$ is invertible:

$$H^{-1} = \Sigma(\mathbf{p})^{-1} H_*^{-1} \Sigma(\mathbf{p})^{-T}$$

- **The Eq. (4.9) can be written as:**

$$\Delta \mathbf{p} = \Sigma(\mathbf{p})^{-1} H_*^{-1} \sum_{\mathbf{x}} \left[\nabla T \Gamma(\mathbf{x}) \right]^T \left[I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right) - T(\mathbf{x}) \right] \tag{4.12}$$

- **The Eq. (4.12) can be changed into two steps:**

$$\Delta \mathbf{p}_* = H_*^{-1} \sum_{\mathbf{x}} \left[\nabla T \Gamma(\mathbf{x}) \right]^T \left[I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right) - T(\mathbf{x}) \right] \tag{4.13}$$

$$\Delta \mathbf{p} = \Sigma(\mathbf{p})^{-1} \Delta \mathbf{p}_* \tag{4.14}$$

- **The warp is updated by:**

$$\mathbf{p} \leftarrow \mathbf{p} - \Sigma(\mathbf{p})^{-1} \Delta \mathbf{p}_* \tag{4.15}$$

Forward Additive Algorithm

The Hager-Belhumeur Algorithm(1998)

Per-compute:

3. Evaluate the gradient ∇T of image $T(\mathbf{x})$
4. Evaluate $\Gamma(\mathbf{x})$
5. Compute the modified steepest descent images $\nabla T\Gamma(\mathbf{x})$
6. Compute the modified Hessian matrix $H_* = \sum_{\mathbf{x}}[\nabla T\Gamma(\mathbf{x})]^T[\nabla T\Gamma(\mathbf{x})]$

Iterate:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
2. Compute the error image $I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})$
7. Compute $\sum_{\mathbf{x}}[\nabla T\Gamma(\mathbf{x})]^T\left[I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right) - T(\mathbf{x})\right]$
8. Compute $\Delta\mathbf{p}_* = {H_*}^{-1}\sum_{\mathbf{x}}[\nabla T\Gamma(\mathbf{x})]^T\left[I\left(\mathbf{W}(\mathbf{x};\mathbf{p})\right) - T(\mathbf{x})\right]$
9. Compute $\Sigma(\mathbf{p})^{-1}$ and update the warp $\mathbf{p} \leftarrow \mathbf{p} - \Sigma(\mathbf{p})^{-1}\Delta\mathbf{p}_*$

until $||\Delta \mathrm{p}|| \leq \xi$

Reference

[1] Baker S, Matthews I. Lucas-Kanade 20 Years On: A Unifying Framework[J]. International Journal of Computer Vision, 2004, 56(3):221-255.

[2] Baker S, Matthews I. Equivalence and efficiency of image alignment algorithms[C]// Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, 2001:I-1090-I-1097 vol.1.

[3] Hartley R, Zisserman A. Multiple view geometry in computer vision. With foreword by Olivier Faugeras. 2nd edition[J]. 2003.