Adaptive and Constrained Algorithms for Inverse Compositional Active Appearance Model Fitting

— CVPR 2008 Paper Supplemental Material —

George Papandreou and Petros Maragos
School of Electrical and Computer Engineering
National Technical University of Athens, Greece
http://cvsp.cs.ntua.gr

1. Including Priors into Flexible Warp-based Inverse Compositional Algorithms

In this note we expand the discussion of Section 4.1 of the main paper on computing the inverse-compositional to additive parameter update \((4+n) \times (4+n)\) Jacobian matrix \(J_\hat{p}\). Full details are given for the particularly interesting case of the thin-plate spline warp [2].

Our starting point is the relationship \(W(x; \hat{p} + J_\hat{p}d\hat{p}) \approx W(W(x; -d\hat{p}); \hat{p})\), which holds for all points \(x\) in the image plane to first order in \(d\hat{p}\) [1, 4]. Differentiation w.r.t. \(d\hat{p}\) yields

\[
\frac{\partial W}{\partial \hat{p}} \bigg|_{(x; \hat{p})} \approx - \frac{\partial W}{\partial x} \bigg|_{(x; \hat{p})} \frac{\partial W}{\partial \hat{p}} \bigg|_{(x; \hat{p}=0)}.
\]

Equation (1) gives \(2 \times (4+n)\) constraints per image point \(x\).

Since the warp \(W\) is uniquely determined by positions of the shape landmarks, it suffices to apply Eq. (1) \(L\) times, once for the spatial position \(x_l, l = 1, \ldots, L\) of each landmark on the mean shape \(s_0\). Putting together the \(L\) resulting terms in a single block matrix equation yields

\[
\begin{bmatrix}
\frac{\partial W}{\partial \hat{p}} \bigg|_{(x_1; \hat{p})} \\
\vdots \\
\frac{\partial W}{\partial \hat{p}} \bigg|_{(x_L; \hat{p})}
\end{bmatrix}
\begin{bmatrix}
J_\hat{p} \\
\vdots \\
J_\hat{p}
\end{bmatrix}
\approx
\begin{bmatrix}
\frac{\partial W}{\partial x} \bigg|_{(x_1; \hat{p}=0)} \\
\vdots \\
\frac{\partial W}{\partial x} \bigg|_{(x_L; \hat{p}=0)}
\end{bmatrix}.
\]

Denoting as \(\partial W / \partial \hat{p} \bigg|_{(x_1, \ldots, x_L; \hat{p})}\) the \((2L) \times (4+n)\) stacked matrix of derivatives on the left-hand-side and as \(\partial W / \partial x \bigg|_{(x_1, \ldots, x_L; \hat{p})} \odot \partial W / \partial \hat{p} \bigg|_{(x_1, \ldots, x_L; 0)}\) the stacked block-by-block matrix product on the right-hand-side of the previous equation, we can write it more compactly as

\[
\frac{\partial W}{\partial \hat{p}} \bigg|_{(x_1, \ldots, x_L; \hat{p})} \odot J_\hat{p} \approx - \frac{\partial W}{\partial x} \bigg|_{(x_1, \ldots, x_L; \hat{p}=0)} \odot \frac{\partial W}{\partial \hat{p}} \bigg|_{(x_1, \ldots, x_L; 0)}.
\]

Solving this with the method of least squares yields the Jacobian estimate

\[
J_\hat{p} = - \left( \frac{\partial W}{\partial \hat{p}} \bigg|_{(x_1, \ldots, x_L; \hat{p})} \odot \frac{\partial W}{\partial \hat{p}} \bigg|_{(x_1, \ldots, x_L; 0)} \right)^{-1} \left( \frac{\partial W}{\partial x} \bigg|_{(x_1, \ldots, x_L; \hat{p})} \odot \frac{\partial W}{\partial \hat{p}} \bigg|_{(x_1, \ldots, x_L; 0)} \right).
\]
which is Eq. (22) of our main paper.

We move forward and show how the matrices involved in Eq. (4) can be computed. Regarding the \((2L) \times (4 + n)\) matrix \(\frac{\partial W}{\partial p}\) , we need compute the \(\frac{\partial W}{\partial p}\) Jacobian. Applying the chain rule on \(W(x, \hat{p}) = S_t(W(x, p))\) and considering separately the similarity \(\mathcal{S}\) and deformation \(p\) parameters gives

\[
\frac{\partial W}{\partial p}(x, \hat{p}) = \begin{bmatrix}
\frac{\partial \mathcal{S}}{\partial \mathcal{T}}(W(x, \hat{p})); \frac{\partial \mathcal{S}}{\partial \mathcal{T}}(W(x, \hat{p}); t) \frac{\partial W}{\partial \mathcal{T}}(x, \hat{p})
\end{bmatrix}
\]

Taking advantage of the fact that we only need to evaluate the quantities above on the landmark positions \(x_l\), it is easy to show (c.f. [4, Sec. 4.1.2]) that

\[
\frac{\partial W}{\partial \mathcal{T}}(x, \hat{p}) = \begin{bmatrix}
s_p \quad s_p^+ \quad 1_x \quad 1_x^+
\end{bmatrix} \begin{bmatrix}
1 + t_1 \\ t_2 \\ 1 + t_2
\end{bmatrix} [s_1 \ldots s_n] + t_2 [s_1^+ \ldots s_n^+],
\]

where \(s_p = s_0 + \sum_{i=1}^n p_i s_t\) is the deformed shape. given the parameters \(p, s^+\) denotes the shape \(s\) rotated counter-clockwise by \(90^\circ\) and \(1_x = [1 \ 0 \ldots \ 1 \ 0]^T\) is the shape with 1's in the \(x\)-coordinate and 0's in the \(y\)-coordinate.

Regarding the \(2L \times 2\) matrix \(\frac{\partial W}{\partial \mathcal{T}}(x, \hat{p})\), we need compute the Jacobian \(\frac{\partial W}{\partial \mathcal{T}}(x, \hat{p})\). Application of the chain rule on \(W(x, \hat{p}) = S_t(W(x, p))\) gives

\[
\frac{\partial W}{\partial \mathcal{T}}(x, \hat{p}) = \frac{\partial \mathcal{S}}{\partial \mathcal{T}}(W(x, \hat{p}) ; t) \frac{\partial W}{\partial \mathcal{T}}(x, \hat{p}) = \begin{bmatrix}
1 + t_1 \\ t_2 \\ 1 + t_2
\end{bmatrix} \frac{\partial W}{\partial \mathcal{T}}(x, \hat{p}).
\]

Computation of the deformation field Jacobian \(\frac{\partial W}{\partial \mathcal{T}}(x, \hat{p})\) depends on the warp family under consideration. For the often used thin-plate spline warp [2], we can write the warp function \(W(x, p)\) in the form of a generalized linear model (c.f. [3, App. F])

\[
W(x, p) = \underbrace{W(p)}_{2 \times (L+3) (L+3) \times 1} \underbrace{k(x)}_{(L+3) \times 1},
\]

where the vector \(k(x)\) is given by

\[
k(x) = \begin{bmatrix}
U(|x - x_1|) \\ \vdots \\ U(|x - x_L|) \\
x \quad y
\end{bmatrix}^T,
\]

\(U(r) = r^2 \ln r^2\) is the spline kernel, and \(W(p)\) is determined by requiring that the warp maps \(s_0\) to \(s_p\). The final result is

\[
\frac{\partial W}{\partial \mathcal{T}}(x, \hat{p}) = W(p) \frac{dk(x)}{dx}\bigg|_x = W(p) \begin{bmatrix}
2(1 + \ln r_{x, 1}^2)(x - x_1)^T \\ \vdots \\ 2(1 + \ln r_{x, L}^2)(x - x_L)^T
\end{bmatrix},
\]

where \(r_{x, l} = \|x - x_l\|_2\). We need to evaluate \(\frac{dk(x)}{dx}\bigg|_x\) for each landmark point \(x_l\). Since the term \(k(x)\) does not depend on the shape parameter \(p\), \(\frac{dk(x)}{dx}\bigg|_x\) can be pre-computed and be subsequently used in every AAM iteration.

The cost of computing the Jacobian matrix \(J_p\) can be analyzed as follows: (a) Computing the \((2L) \times (4 + n)\) matrix \(\frac{\partial W}{\partial p}(x, \hat{p})\) is \(O(nL)\). (b) Computing the \(2L \times 2\) matrix \(\frac{\partial W}{\partial \mathcal{T}}(x, \hat{p})\) is \(O(L)\). (c) Forming the stacked block-by-block matrix product \(\frac{\partial W}{\partial \mathcal{T}}(x, \hat{p}) \odot \frac{\partial W}{\partial \mathcal{T}}(x, 0)\) is \(O(nL)\). (d) Forming the \((4 + n) \times (4 + n)\) least-squares system matrix \(\frac{\partial W}{\partial \mathcal{T}}^T \frac{\partial W}{\partial \mathcal{T}}(x, \hat{p})\) is \(O(n^2L)\). (e) Inverting the same system matrix is \(O(n^3)\). Overall, the last two operations dominate the cost of the procedure, whose overall complexity is thus \(O(n^2L + n^3)\).
References


