APPENDIX A
IDENTIFYING OCCLUSION FORCES

As depicted in Fig. 1, we consider the residual along a segment $\bar{V}_n V_m$ joining two adjacent vertices along the occlusion boundary, $\bar{V}_m \equiv (x_m, y_m)$ and $\bar{V}_n \equiv (x_n, y_n)$, with normal $\hat{n} \equiv (\hat{n}_x, \hat{n}_y)$. Without loss of generality, we assume that $\hat{n}_y > 0$, and $\hat{n}_y > |\hat{n}_x|$ i.e., the edge is within $45^\circ$ of the horizontal image axis, and the occluding side lies below the boundary segment. Accordingly, the anti-aliasing weights in (eqn.15 [1]) become

\[ w(x) = \frac{(x - q(x)) \cdot \hat{n}}{\hat{n}_y}. \] (1)

To compute $\nabla_\theta \hat{E}$ we need the partial derivatives of $\hat{R}$ with respect to $\theta$ (eqn.17 [1]). Some terms are due to changes in $\hat{R}$ while others are due to changes in the weights $w$. (For notational brevity we omit the parameters $\theta, L$ and $T$ when writing the residual function $R$).

Our goal here is to associate the terms of the derivative due to changes in the weights with the second line in (eqn.13 [1]). We denote this term $C$:

\[ C = \sum_{x \in \mathbb{A} \cap \mathbb{N}^2} \frac{\partial w(x)}{\partial \theta_j} [R(x) - R(q(x))]. \] (2)

Let $q_y(x)$ be the vertical projection of the point $x$ onto the line that contains the segment $\bar{V}_n \bar{V}_m$. Also let $(\bar{x}, \bar{y})$ denote the coordinate axes of the image plane. For any point $x$ in $\mathbb{A}$ one can show that

\[ w(x) = (x - q_y(x)) \cdot \bar{y}. \] (3)

Differentiating $w$ with respect to $\theta_i$ gives

\[ \frac{\partial w(x)}{\partial \theta_j} = -\frac{\partial q_y(x)}{\partial \theta_j} \cdot \bar{y}. \] (4)

The vertically projected point $q_y(x)$ lies in the segment $\bar{V}_m \bar{V}_n$, and therefore there exists $t \in [0, 1]$ such that

\[ q_y(x) = (1 - t)\bar{V}_m + t\bar{V}_n. \] (5)

We can then differentiate $q_y(x)$ with respect to $\theta_j$, i.e.,

\[ \frac{\partial q_y(x)}{\partial \theta_j} = \frac{\partial t}{\partial \theta_j} (\bar{V}_n - \bar{V}_m) + (1 - t) \frac{\partial \bar{V}_m}{\partial \theta_j} + t \frac{\partial \bar{V}_n}{\partial \theta_j}. \] (6)

Now, let $v_j$ denote the curve speed at $x$ when $\theta_j$ varies. It is the partial derivative of the curve $\Gamma_y$ with respect to a given pose parameter $\theta_i$, i.e.,

\[ v_j(x) = (1 - t_x) \frac{\partial \bar{V}_m}{\partial \theta_j} + t_x \frac{\partial \bar{V}_n}{\partial \theta_j}. \] (7)

Because $(\bar{V}_n - \bar{V}_m) \cdot \hat{n} = 0$, with the curve speed $v_j$ above, we obtain the following from (6)

\[ \hat{n} \cdot \frac{\partial q_y(x)}{\partial \theta_j} = \hat{n} \cdot v_j(q_y(x)). \] (8)

Because $\bar{x} \cdot \frac{\partial q_y(x)}{\partial \theta_j} = 0$ we obtain

\[ \hat{n} \cdot \frac{\partial q_y(x)}{\partial \theta_j} = (\hat{n}_x \hat{x} + \hat{n}_y \hat{y}) \cdot \frac{\partial q_y(x)}{\partial \theta_j} = \hat{n}_y (\frac{\partial q_y(x)}{\partial \theta_j} \cdot \hat{y}). \] (9)

Therefore

\[ \frac{\partial w(x)}{\partial \theta_j} = -\frac{\partial q_y(x)}{\partial \theta_j} \cdot \bar{y} = -\hat{n}_y \frac{\partial q_y(x)}{\partial \theta_j} = -\hat{n} \cdot v_j(q_y(x)) \cdot \frac{\partial w(x)}{\partial \theta_j}. \] (10)

We can rewrite $C$ as follow:

\[ C = \frac{1}{\hat{n}_y} \sum_{x \in \mathbb{A} \cap \mathbb{N}^2} -[R(x) - R(p(x))] \hat{n} \cdot v_j(q_y(x)). \] (11)

We now show that with some approximations we can associate this term with the second term in (eqn.13 [1]). We assume $R$ and $R^+$ to be smooth. Therefore

\[ R(q(x)) \approx R(q_y(x)) \] (12)

\[ R(x) \approx R^+(q_y(x)). \] (13)

Therefore, given the definition of the occlusion forces (eqn.13 [1]), $C$ can be approximated by $\hat{C}$ defined as:

\[ \hat{C} = \frac{1}{\hat{n}_y} \sum_{x \in \mathbb{A} \cap \mathbb{N}^2} -f_{oc}(q_y(x)) \cdot v_j(q_y(x)). \] (14)

The division by $\max(|\hat{n}_x|, |\hat{n}_y|)$ in our definition of the weight (eqn.15 [1]) ensures that within each vertical line there is a single point $x$ with integer coordinates and its weight in $[0, 1]$. This appears more clearly in (3).

Given a vertical line with constant $x$, this point has the coordinates $x = (x, \lceil \bar{y}(x) \rceil)$ where

\[ \bar{y}(x) = y_m + (x - x_m) \Delta_y, \] (15)

with

\[ \Delta_y = (y_n - y_m)/(x_n - x_m) \] (16)

being the slope of the segment. From (eqn.16 [1]) we obtain

\[ \mathbb{A} \cap \mathbb{N}^2 = \{ (x, \lceil \bar{y}(x) \rceil) | x \in \mathbb{N}, p((x, \lceil \bar{y}(x) \rceil)) \in \bar{V}_m \bar{V}_n \}. \] (17)

We assume now that the condition that the point $x$ should orthogonally project into the segment (i.e., $q(x, \lceil \bar{y}(x) \rceil) \in \bar{V}_m \bar{V}_n$) can be approximated by the condition $x \in [x_m, x_m]$ as the line is within $45^\circ$ of vertical. The resulting approximate anti-aliased region is no longer rectangular but a parallelogram with two vertical sides.
After discretization on the grid this might result, in some cases, of a point being neglecting near the extremities of the segment.

\[ A \cap \mathbb{N}^2 \approx \{(x, \lbrack \bar{y}(x) \rbrack) | x \in \{\lbrack x_n \rbrack, \ldots, \lbrack x_m \rbrack\}\}. \quad (18) \]

This approximation of the anti-aliased region and the fact that \( q_y(x, \lbrack \bar{y}(x) \rbrack) = (x, \bar{y}(x)) \) allows us to approximate \( \tilde{C} \) by \( \hat{C}_2 \) as follow:

\[ \hat{C}_2 = \frac{1}{n_y} \sum_{x=\lbrack x_n \rbrack}^{\lbrack x_m \rbrack} -f_{oc}(x, \bar{y}(x)) \cdot v_j(x, \bar{y}(x)) \cdot \epsilon. \quad (19) \]

Now let \( \Delta V = (\Delta_V, \Delta_y) \) denote the 2D displacement along the segment \( \bar{V}_m \bar{V}_n \) when \( x \) is incremented by one. Then

\[ |\Delta V| = \sqrt{1 + \Delta_y^2} = \frac{1}{n_y}. \quad (20) \]

We also introduce \( t = x - \lbrack x_n \rbrack, N = \lbrack x_m \rbrack - \lbrack x_n \rbrack \) and \( \epsilon = \lbrack x_n \rbrack - x_n \). We obtain, after some derivation,

\[ \hat{C}_2 = |\Delta V| \sum_{t=0}^{N-1} -f_{oc}(\bar{V}_n + (t+\epsilon)\Delta V) \cdot v_j(\bar{V}_n + (t+\epsilon)\Delta V). \quad (21) \]

This last approximation of \( C \) can be identified as a discrete approximation of the integral, along \( \bar{V}_n \bar{V}_n \), of

\[ f(x) = -f_{oc}(x) \cdot v_j(x). \quad (22) \]

This term corresponds to the contribution of the segment in the second term of (eqn.13 [1]). We demonstrated that the implementation based on a discrete image domain with anti-aliasing yields, after differentiation, terms that are consistent (up to an approximation) with the occlusion forces that were obtained by differentiating the objective function defined on the continuous image domain.

**APPENDIX B**

**BLOCKWISE BFGS UPDATES**

Quasi-Newton methods rely on a good approximation to the Hessian of the objective function. Better approximations often lead to faster convergence. Here we approximate the objective function \( H(q) \) using a variant of the BFGS update method. It was adapted to exploit the partial independence of separated fingers. The resulting sparseness of the Hessian leads to the definition of a Blockwise BFGS update. This blockwise update increased the convergence rate by a factor ranging from 3 to 6.

To obtain a good convergence rate even during the first iterations, it is important to initialize the approximate Hessian well. Rather than using the identity matrix, as is often done, we used a scaled version of the matrix \( J_q^Y \) where \( J_q^Y \) is the Jacobian of projected vertices with respect to \( \theta \). This favors the displacements in the depth direction for which the gradient is small due to weak support from the image data.

Because the contributions to the overall cost of two well-separated fingers are independent, the true Hessian will not be fully populated but will exhibit blocks of zeros (Fig. 2.a). This sparsity is accentuated if, instead of using joints angles, we individually parameterize the pose of each bone using a 7D vector, comprising a quaternion and a translation vector such that \( \theta \in \mathbb{R}^{126} \) (The bones of the wrist and the arm are rigidly fixed and therefore we need not represent one of the two).

As shown in Fig. 2.b), non-zero entries of the 126 × 126 Hessian appear on 7 × 7 blocks. Each off-diagonal block corresponds to a pair of hand parts that either occlude one another or share some facets in their influence area for the pose space deformation.

To exploit the Hessian sparsity, with quaternions used to parameterize bones, without the need for further non-linear constraints, we first decompose the function \( E(\theta) \) into \( E(\theta) = E_q(Q(\theta)) \), where \( Q \) maps the joint-angle representation to quaternions. The Hessian \( \frac{\partial^2 E^2}{\partial \theta^2} \) is then approximated by \( H_q(\frac{\partial^2 E_q}{\partial \theta^2} \) where \( H_q = Q(\frac{\partial^2 E}{\partial \theta^2} \).

At each step, we refine the Hessian approximation \( H_q \) with an adapted BFGS update. We approximate the structure of \( H_q \) by assuming complete independence between parts of the hand. This produces block-diagonal structure (see Fig. 2.c) where non-zero entries occur in 7 × 7 blocks along the diagonal. The standard BFGS update does not exploit this structure, and would otherwise populate the entire matrix. Using the BFGS formula, rather than update \( H_q \), we only update the non-zero 7 × 7 blocks on the diagonal independently. About 7 gradient evaluations are then necessary to obtain a reasonable local approximation of the Hessian, while the standard BFGS method would require about 28 evaluations. This has a direct impact on the convergence rate of the optimization. The method induces more zeros than in the true Hessian but still leads to significant improvement over the standard BFGS update. As we keep performing
increments on $\theta$ during the optimization, we do not need to add nonlinear constraints that enforce validity of relative poses of connected bones that would be necessary if the increments were done in the quaternion representation space. The improvement in the minimization process, in terms of the number of objective function evaluations, is shown in Fig. 2, where we estimated the pose for a single frame.

**REFERENCES**