Surgical Scissor Training

Felix Jonathan, Michael Norris

March 25, 2016

1 Setup

- Scissor with EM sensor attached rigidly
- Cardboard with EM sensor attached rigidly
- EM pointer tool, pre-calibrated with Aurora software

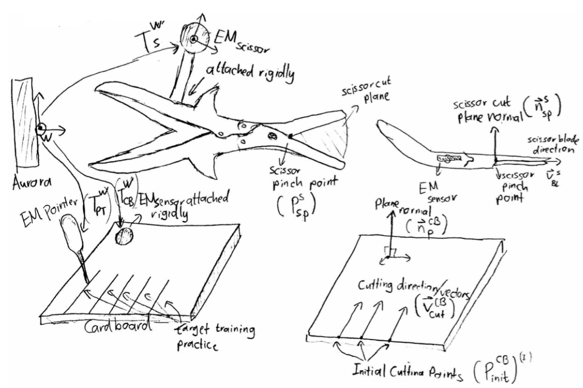

Figure 1: Schematic Diagram Of Scissor Training Setup

2 Input Data

- T^W_S: EM sensor on scissor provides transformation between world coordinate frame (Aurora) with scissor local coordinate frame.
- $T^W_C B$: EM sensor on scissor provides transformation between world coordinate frame (Aurora) with cardboard local coordinate frame.

- $P_P^W T$: EM sensor provides the end point position in world coordinate frame (Aurora). The sensor actually providing the transformation to EM pointer, however the orientation of pointer is not used in this training.

3 Data Collection Procedure

1. Trace the cardboard surface with pointer tip to get cardboard surface point cloud data. This will be used to calculate cardboard normal vector.
2. Use a ruler and trace along several line with pointer tip. Record each line data individually.
3. Move pointer tip to every initial cutting point position and record each point individually.
4. Record data of scissor cutting the traced line individually

4 PseudoCode list

This function is used to the reference frame of a point, input, or vector:

Function *ChangeReference (Data input, Transform reference)*
  Data: point(P_{data}^W), vector($\vec{v}_{data}^W$), or transformation(T_{data}^W)) and target reference frame in world coordinate system (T_{ref}^W)
  Result: Data in reference coordinate system: P_{data}^W or $\vec{v}_{data}^W$ or T_{data}^{ref}
  switch *Data type* **do**
    case *Data type == transform*
      $T_{data}^{ref} = (T_{ref}^W)^{-1} T_{data}^W$ **return** T_{data}^{ref}
    end
    case *Data type == point*
      $P_{data}^{ref} = (T_{ref}^W)^{-1} P_{data}^W$ **return** P_{data}^{ref}
    end
    case *Data type == vector*
      $\vec{d}_{data}^{ref} = (R_{ref}^W)^{-1} \vec{d}_{data}^W$ **return** $\vec{d}_{data}^{ref}$
    end
  endsw

Algorithm 1: Change Reference Frame

This function is used to generate a estimated 3D best line fit from a point cloud:

Function *CalculateVectorFromPointCloud (pointArray Input)*

Data: Point cloud $P_{data}^{W}[1..n]$ (n: number of points)

Result: vector that represents line direction $\vec{v}_{line}^{ref}$

Initialize $P_{data}^{ref}[1..n]$

forall the $P_{data}^{W}[i]$ *in* $P_{data}^{W}[1..n]$ **do**

$P_{data}^{ref}[i] = \text{ChangeReferenceFrame}(P_{data}^{W}[i], T_{ref}^{W})$

end

Calculate the mean of point cloud $P_{data}^{ref}[n]$: $\overline{P}_{data}^{ref}$

Initialize $\widetilde{P}_{data}^{ref}[1..n]$

forall the $P_{data}^{ref}[i]$ *in* $P_{data}^{ref}[1..n]$ **do**

$\widetilde{P}_{data}^{ref}[i] = P_{data}^{ref}[i] - \overline{P}_{data}^{ref}$

end

Calculate Covariance Matrix of C:

$$C = \frac{(\widetilde{P}_{data}^{ref}[n])^{T}(\widetilde{P}_{data}^{ref}[n])}{n-1}$$

Calculate SVD of C to get the first eigenvector of point cloud, which represents $\vec{v}_{line}^{ref}$ **return** $\vec{v}_{line}^{ref}$

Algorithm 2: Estimate Vector From Point Cloud

This function is used to estimate initial cutting point from pointcloud:

Function *estimateInitialCuttingPoint (Array of pointArray Input, TransformArray scissorToCardboard)*

Data: Collection of point cloud $P_{data}^{C}B[1..m][1..n](m : number of lines on training, n : points in one initial cutting point), collection of transformations when getting the point cloud T_{S}^{CB}[$

Result: pinch point of scissor in scissor coordinate system P_{sp}^{s}

Initialize Array of Points $\overline{P}_{data}^{s}[1..m]$

forall the $P_{data}^{C}B[i]$ *in* $P_{data}^{C}B[1..m]$ **do**

$\overline{P}_{data}^{S}[i]$ = ChangeReference(Mean of point cloud $P_{data}^{C}B[i][1..n]$, $(T_{S}^{CB}[i])^{-1}$)

end

Algorithm 3: Estimate Initial Cutting Point From Point Clouds of Initial Cutting Coordinate

This function is used to the estimate scissor cutting plane normal direction:

Function *getScissorNormal (vectorArray LineOfCuts,normal PlaneRef, TransformArray WorldToScissor, TransformArray WorldToPlaneRef)*

Data: Collection of line of cut data $\vec{v}^{ref}_{cut}[1..n], planenormaldata \vec{n}^{ref}_{P}, collection of transforms T^{W}_{S}[1..n], T^{W}_{ref}[1..n]$ when getting doing scissor cut

Result: $\vec{n}^{s}_{sp}$

Initialize $\vec{v}^{s}_{sp}[1..n]$

forall the $\vec{v}^{ref}_{cut}[i]$ *in* $\vec{v}^{ref}_{cut}[1..n]$ **do**

T^{S}_{ref} = ChangeReference($T^{W}_{ref}, T^{W^{-1}}_{S}$)

$\vec{v}^{s}_{sp}[i]$= ChangeReference(($\vec{v}^{ref}_{cut}[i] \times \vec{n}^{ref}_{P}$), T^{S}_{ref})

end

Calculate the mean of collection of vector $\vec{v}^{s}_{sp}[1..n]$: $\overline{\vec{v}}^{s}_{sp}$

return $\overline{\vec{v}}^{s}_{sp}$

Algorithm 4: Estimate Scissor plane of cut normal direction

5 Processing Data

1. Use function ChangeReference to transform all collected data (scissor transformation and pointer tip point cloud result from tracing cardboard surface and line) into cardboard coordinate system.

2. Estimate cardboard plane normal from point cloud data collected when tracing cardboard surface by using function SeptumSurfaceGeneration described in septum tracing procedure.

3. Estimate Line of cut vectors from point cloud data when doing line tracing with function CalculateVectorFromPointCloud

4. Estimate initial cutting point from initial cutting position data by using function estimateInitialCuttingPoint.

5. Calculate the scissor normal data with function getScissorNormal from estimated line of cut vectors, cardboard plane, and transformation from cardboard sensor to scissor.