DISCUSSION LOG

Sikang Yan

University of Kaiserslautern

yan@rhrk.uni-kl.de

April 7, 2019

THEORY

In our cloth simulation, we follow the precedure written by Rohmer et al.. We consider to begin with the *Defomation gradient* **F**, which is defined by

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}, \tag{1}$$

where the **x** denotes the deformed vector and the **X** denotes the reference vector.
Since we have no further information about the mapping from **X** to **x**, we use the vector $\left(\mathbf{u_1}, \mathbf{u_2}\right)$ and $\left(\overline{\mathbf{u_1}}, \overline{\mathbf{u_2}}\right)$ to approximate the Defomation gradient, which is defined as

$$\mathbf{F} = \left[\mathbf{u_1}, \mathbf{u_2}\right]\left[\overline{\mathbf{u_1}}, \overline{\mathbf{u_2}}\right]^{-1}, \tag{2}$$

Attention should be paid especially:

- eq.(2) characrize only the 2D deformation of each triangle.
- in Rohmer et al. **F** is symbolised as **T**.

KW13

We provide here our code preceed with concrete data:

- $faces(i,j)$ is the jth vertex of the ith triangle, here we choose the $face(0,0), face(0,1), face(0,2)$ as examples and calculate the $VecT$ and $VecR$ for $face(0,0)$.

cloth_vec

$$VecT = el_1 VertT_1 - el_1 VertT_2; el_1 VertT_1 - el_1 VertT_3 \tag{3}$$

$$VecR = el_1 VertR_1 - el_1 VertR_2; el_1 VertR_1 - el_1 VertR_3 \tag{4}$$

cloth_vec

$$VecT = [0.771842 \; -0.0144887 \; 6.39045] - [0.780121 \; -0.0186188 \; 6.37318] \tag{5}$$

$$[0.771842 \; -0.0144887 \; 6.39045] - [0.737177 \; -0.00912791 \; 6.39292 \tag{6}$$

$$VecR = [0.759919 \; -0.015194 \; 6.38401] - [0.767822 \; -0.0212492 \; 6.3669]; \tag{7}$$

$$[0.7599190.015194 \; 6.38401] - [0.726977 \; -0.00749985 \; 6.38692] \tag{8}$$

$$\tag{9}$$

such that

cloth_vec

$$
\begin{aligned}
VecT =& [-0.0079\ 0.0061\ 0.0171\ 0.0329\ -0.0077\ -0.0029]; && (10)\\
VecR =& [-0.0083\ 0.0041\ 0.0173\ 0.0347\ -0.0054\ -0.0025]; && (11)\\
& && (12)
\end{aligned}
$$

where the first 3 entries of $VecR$ is the vector $\mathbf{u_1}$ and the last 3 entries of $VecR$ is the vector $\mathbf{u_2}$. $VecT$ analogiously.

cloth_eig_2D

we use here $Eigen :: Map$ to transform the vector $VecT$ and $VecR$ to $2 * 2$ 2D deformation gradient $\mathbf{F}$.

$$\mathbf{F} = \begin{bmatrix} -0.0083 & 0.0347 \\ 0.0041 & -0.0054 \end{bmatrix} \begin{bmatrix} -0.0079 & 0.0329 \\ 0.0061 & -0.0077 \end{bmatrix}^{-1} \tag{13}$$

hence the $\mathbf{F}$ has a rotation information $\mathbf{R}$ and a stretch information $\mathbf{U}$,

$$\mathbf{F} = \mathbf{R}\mathbf{U}. \tag{14}$$

we use $\mathbf{F}^T\mathbf{F}$ to eliminate the ratotion information to obtain $\det \mathbf{R} = 1$

$$\begin{aligned}
\mathbf{F}^T\mathbf{F} =& \left(\mathbf{R}\mathbf{U}\right)^T\mathbf{R}\mathbf{U} && (15)\\
=& \mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U} && (16)\\
=& \mathbf{U}^2 && (17)\\
=& \mathbf{C} && (18)
\end{aligned}$$

and using decomposition, if we have the form

$$\mathbf{C} = \lambda_1^2\mathbf{v}_1\mathbf{v}_1^T + \lambda_2^2\mathbf{v}_2\mathbf{v}_2^T \tag{19}$$

then we could obtain the $\mathbf{U}$ by applying

$$\mathbf{U} = \lambda_1\mathbf{v}_1\mathbf{v}_1^T + \lambda_2\mathbf{v}_2\mathbf{v}_2^T \tag{20}$$

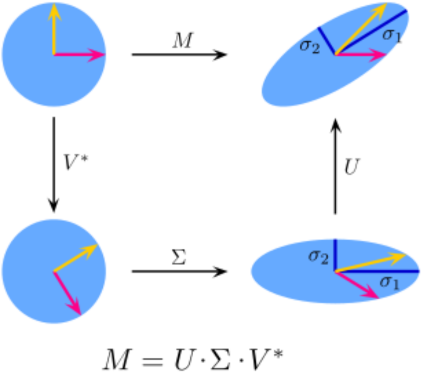

Figure: Singular Value Decomposition SVD

the main backdraw of Rohmer et al. is that this is only applied for 2D problem, thus we need a new algorithm, which can also take the consideration for the vertical deformation. Therefore, the *Kabisch Algorithm* is introduced.

Kabisch Algorithm

Let $\mathbf{P}$ denotes the points set of template frame, and $\mathbf{Q}$ denotes the points set of reference frame. The *optimal rotation matrix* $\mathbf{R}$ can be calculted as

$$\mathbf{R} = \mathbf{V}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{bmatrix}\mathbf{U}^T \tag{21}$$

where $\mathbf{V}$ and $\mathbf{U}$ and the *SVD* of *cross-covariance matrix* $\mathbf{H}$, which is determined by

Theorem (cross-covariance matrix)

$$\mathbf{H} = \mathbf{P}^T\mathbf{Q} \tag{22}$$

and $d = \det \mathbf{V}\mathbf{U}^T$

The matrix $\mathbf{H}$ is derived from the *orthogonal Procrustes problem*[**?**]. It is defined as the least-squares problem of transforming a given matrix $\mathbf{P}$ into a given matrix $\mathbf{Q}$ by an orthogonal transformation matrix $\mathbf{R}$, such that the sums of squares of the residual matrix $\mathbf{E} = \mathbf{\Omega P} - \mathbf{Q}$ is a minimum

$$\mathbf{R} = \arg\min_{\mathbf{\Omega}} \|\mathbf{\Omega P} - \mathbf{Q}\|_F \quad \text{subject to} \quad \mathbf{\Omega}^T\mathbf{\Omega} = \mathbf{I}, \tag{23}$$

where $\|\cdot\|_F$ is the Frobenius norm. This problem is equivalent to find the nearest orthogonal matrix $\mathbf{R}$ to a given matrix $\mathbf{H} = \mathbf{P}^T\mathbf{Q}$ [**?**], which most closely maps $\mathbf{P}$ to $\mathbf{Q}$. Thus, we use $\mathbf{H}$ to approximate the deformation gradient $\mathbf{F}$ in Eq. (**??**)

$$\widetilde{\mathbf{F}} = \mathbf{H}. \tag{24}$$

The cross-covariance matrix discribe the covariance of one process with the other at pairs of time points and measure of similarity. We use therefore the cross-covariance matrix $\mathbf{H}$ to approximate our deformation gradient $\mathbf{F}$.

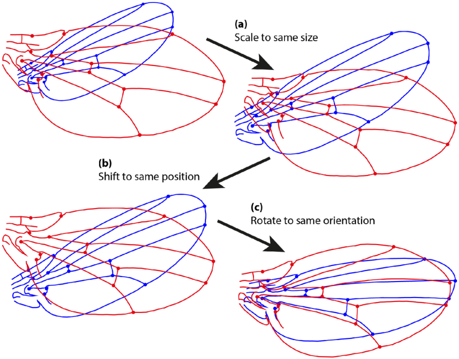

Figure: Procrustes superimposition

Let i be the all index set of the nearst vertice of vertex l in the template frame and $\bar{i}$ be the all index set of the nearst vertice of vertex l in the template frame. Process define

$$\mathbf{P} = \begin{bmatrix} \vec{x}_1 & \vec{y}_1 & \vec{z}_1 \\ \vec{x}_2 & \vec{y}_2 & \vec{z}_2 \\ & \vdots & \\ \vec{x}_N & \vec{y}_N & \vec{z}_N \end{bmatrix} \quad 1, 2, \ldots, N \in i \tag{25}$$

$$\mathbf{Q} = \begin{bmatrix} \overline{\vec{x}}_1 & \overline{\vec{y}}_1 & \overline{\vec{z}}_1 \\ \overline{\vec{x}}_2 & \overline{\vec{y}}_2 & \overline{\vec{z}}_2 \\ & \vdots & \\ \overline{\vec{x}}_N & \overline{\vec{y}}_N & \overline{\vec{z}}_N \end{bmatrix} \quad 1, 2, \ldots, N \in \bar{i} \tag{26}$$

and we apply to obtain the deformation gradient $\mathbf{F}$ of vertex l

$$\mathbf{F} = \mathbf{P}^T \mathbf{Q} \tag{27}$$

Whereas the deformation gradient measures the local deformation, the *spacial velocity gradient* $\mathbf{L}$ describes the rate of deformation, given by

$$\mathbf{L} = grad\mathbf{v} = \dot{\mathbf{F}}\mathbf{F}^{-1}, \tag{28}$$

where $\mathbf{v}$ denotes the *velocity* of the material point $\mathbf{X}$, and $\dot{\mathbf{F}}$ denotes the material time derivative of deformation gradient $\mathbf{F}$.
Analogically, we apply the polar decomposition on the spacial velocity gradient $\mathbf{L}$[**?**], and obtain

$$\mathbf{L} = \mathbf{D} + \mathbf{W}. \tag{29}$$

Thus, we could decompose the tensor $\mathbf{L}$ into its symmetric part $\mathbf{D}$ and skew-symmetric part $\mathbf{W}$

$$\mathbf{D} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^T), \tag{30}$$

$$\mathbf{W} = \frac{1}{2}(\mathbf{L} - \mathbf{L}^T). \tag{31}$$

where $\mathbf{D}$ is called the *rate of strain tensor* and $\mathbf{W}$ is called the *rate of rotation tensor*.

Assuming $\mathrm{d}\mathbf{x}$ is a material line element in the current configuration, the rate of change of its length $\dot{\epsilon}_{ii}$ and angle $\dot{\gamma}_{ij}$ is measured by means of **D**[?].

$$\dot{\epsilon}_{ii} = \mathbf{e}_i \mathbf{D} \mathbf{e}_i = D_{ii} \tag{32}$$

$$\dot{\gamma}_{ij} = 2\mathbf{e}_i \mathbf{D} \mathbf{e}_j = D_{ij} \tag{33}$$

where D_{ij} are the ij-entries of **D**. Additionally, if **D** has the same base $\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}$ as in eq.(20), than

$$D_{11} = \frac{\dot{\lambda}_1}{\lambda_1} \tag{34}$$

$$D_{22} = \frac{\dot{\lambda}_2}{\lambda_3} \tag{35}$$

$$D_{33} = \frac{\dot{\lambda}_2}{\lambda_3} \tag{36}$$

KW13

from 0-2 EigNorm1 8759 NaN, Problrm:: 26278 has -0 eigenvalues, can't be squre. So I use a if(isnan) and define idx=0.
add lambda1,2,3 for all, which is in debug

KW14

save the backsite of clothes using 1-2, 1-3, 1-4. NOt much differences from colormap, need a new color map???
save lambda1 for 1,74, they are different, but slightly.
lambda2 has more differences
do also for lambda3

KW15

check color run application in loop consider the kr to ring 1-2 1-3 ... 1-4 2-5 ... 72-75

tensor flow

add neighbor2x

Neo Hook matrial

KW15

please refer

`http://web.mit.edu/abeyaratne/Volumes/RCA_Vol_II.pdf` p68-70

λ and $\mathbf{D}$ relationship

remedy Neighbor2x

matrix_example.cpp

add func to calculate D and L ,think timestep is to small 2-3

`http://www.continuummechanics.org/velocitygradient.html`

0.006 $1/200$

think should use less 2% points for KDtree

KW15

it is because the date set has isolated points: but I thought it doesn't matter, it influence only the colormap.
thoese points can be deleted during the optimazation.
calculate D using kd-tree.
ask Ali the offical video about clothing deforamtion.
need to be emphasised that we focus on the CR(next time step)

KW15

LIbreoffice Draw A1

add results