Geometric Representations and Deep Gaussian Conditional Random Field Networks for Computer Vision

Raviteja Vemulapalli
Department of Electrical and Computer Engineering
University of Maryland, College Park

Advisory Committee:

Professor Rama Chellappa (Advisor)

Professor Larry S. Davis

Professor Min Wu

Professor Amitabh Varshney

Professor Ramani Duraiswami

Acknowledgements

Professor Rama Chellappa
University of Maryland
College Park

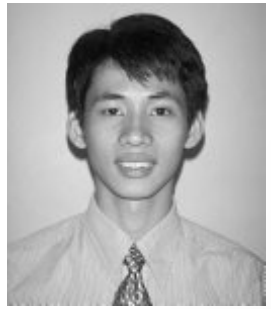
Dr. Oncel Tuzel
Apple

Dr. Felipe Arrate
Federico Santa Maria Technical
University, Chile

Dr. Jaishanker Pillai
Google

Dr. Kevin Zhou
Siemens Healthcare
Technology Center

Dr. Ming-Yu Liu
Mitsubishi Electric Research
Laboratories



Dr. Hien Van Nguyen
Uber ATC

Representation and Context Modeling

- *Representation* and *context modeling* are two of the most important factors that affect the performance of computer vision algorithms.

- Representations such as Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG), and more recently, deep network-based features have played a crucial role in various applications such as depth estimation, image retrieval, 3D reconstruction, object detection, object recognition, etc.

- Spatial context modeling tools such as conditional random field models have played a crucial role in applications like image enhancement, image segmentation, semantic scene understanding, etc.

Overview

- This thesis focuses on both the *representation* and *context modeling* aspects of computer vision.

Representation

Geometric representations for skeleton-based human action recognition

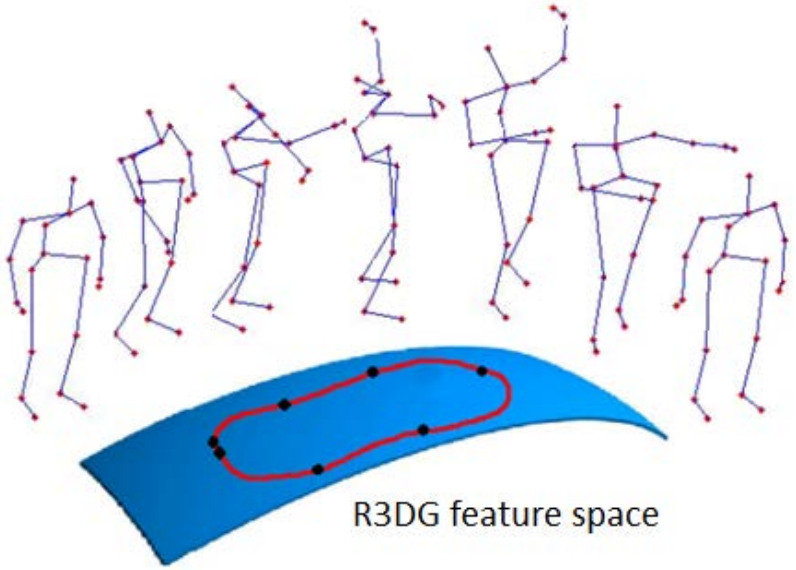
Context Modeling

Gaussian conditional random field-based deep networks for modeling spatial context

Overview of the talk

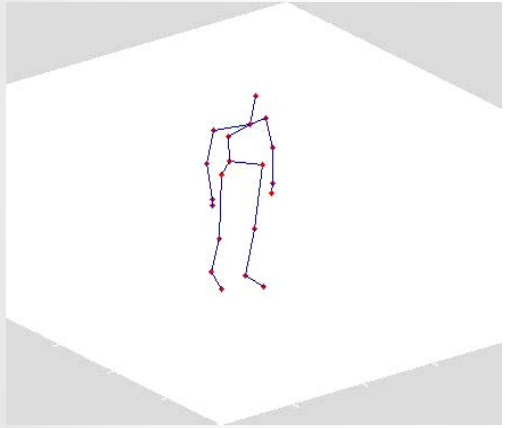
- Relative 3D geometry-based skeletal representations for human action recognition
- Rolling the special orthogonal group for recognizing human actions from 3D skeletal data
- Semantic image segmentation using Gaussian Conditional Random Field (CRF) network
- Image denoising using Gaussian CRF network
- Classification of manifold features using multiple kernel learning
- Unsupervised cross-modal medical image synthesis using mutual information maximization
- Future work

R3DG Features: Relative 3D Geometry based Skeletal Representations for Human Action Recognition

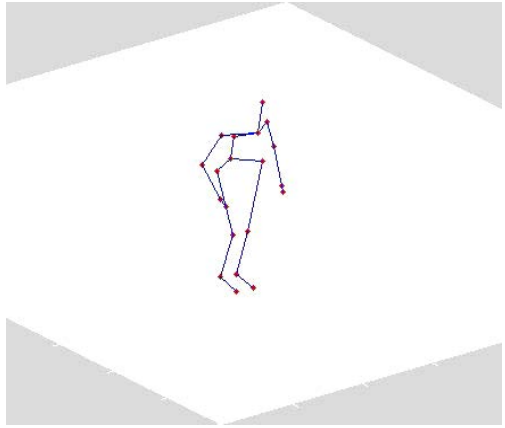


Motivation

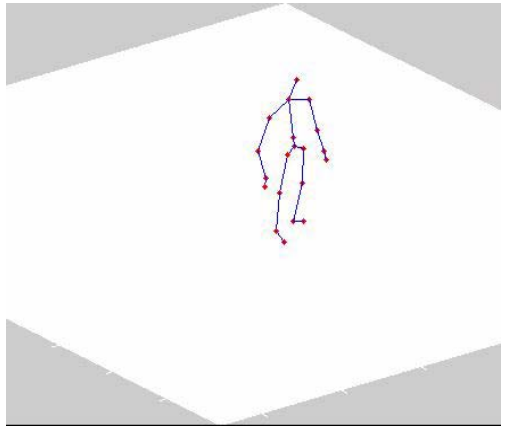
- Humans can recognize many actions directly from skeletal sequences.



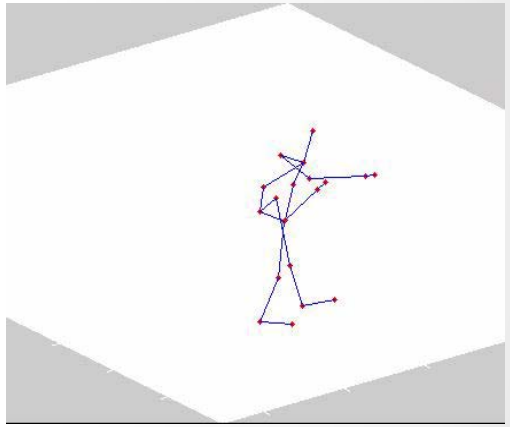
Tennis serve



Jogging



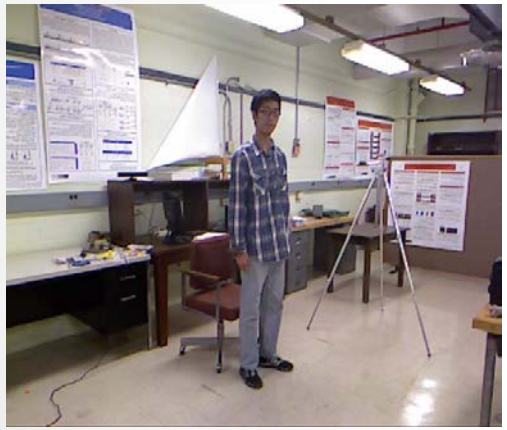
Sit down



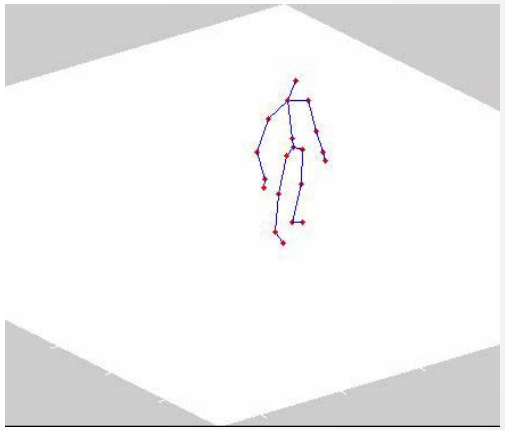
Boxing

Motivation

Human performing an action → Depth sensors + Depth-based skeleton estimation algorithm [Shotton 2011] → Real-time skeletal sequence



UTKinect dataset [Xia2012]



J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, "Real-time Human Pose Recognition in Parts From a Single Depth Image", In CVPR, 2011.

L. Xia, C. C. Chen and J. K. Aggarwal, "View Invariant Human Action Recognition using Histograms of 3D Joints ", In CVPRW, 2012.

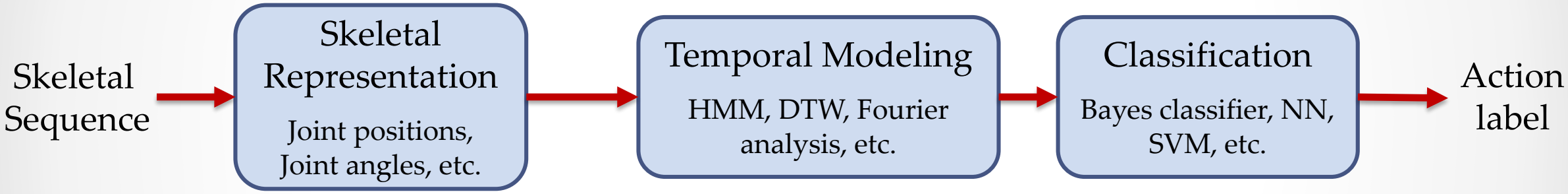
Motivation

Gesture-based Control

Elderly Care

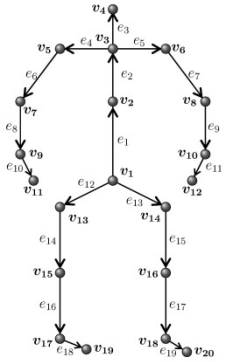
Teaching Robots

Skeleton-based Action Recognition

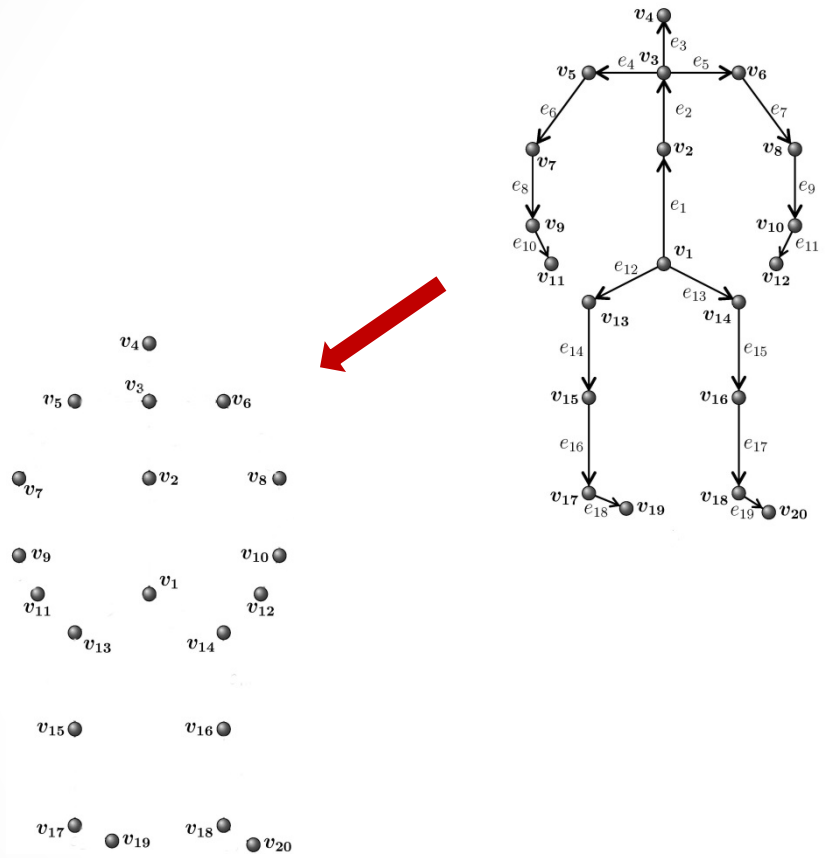


Overview of a typical skeleton-based action recognition system

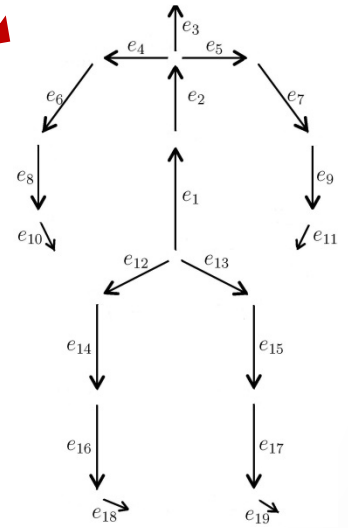
How should we represent a 3D human skeleton for action recognition ?



Human Skeleton: Points or Rigid Rods?



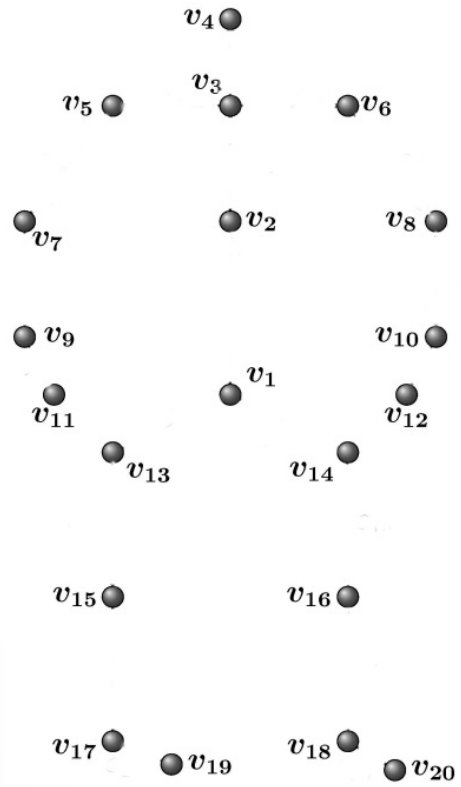
Set of points
(joints)



Set of rigid rods
(body parts)

Human Skeleton as a Set of Points

- One of the most popularly-used skeletal representation.



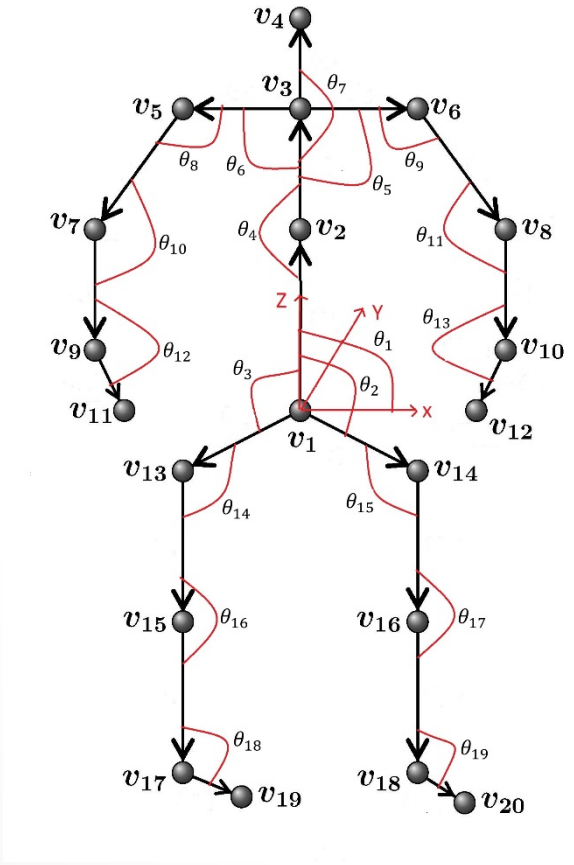
Representation

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ \vdots \\ v_{19} \\ v_{20} \end{bmatrix}$$

Concatenation of the 3D coordinates of the joints.

Human Skeleton as a Set of Rigid Rods

- Spatial configuration of a set of rods can be represented using joint angles.



Representation

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{18} \\ \theta_{19} \end{bmatrix}$$

Concatenation of the 3D joint angles.

Proposed Representation: Motivation

- Human actions are characterized by how different body parts move relative to each other.
- For action recognition, we need a skeletal representation whose temporal evolution directly describes the relative motion between various body parts.
- We represent a human skeleton using the relative 3D geometry between different body parts.

Relative 3D Geometry between Body Parts

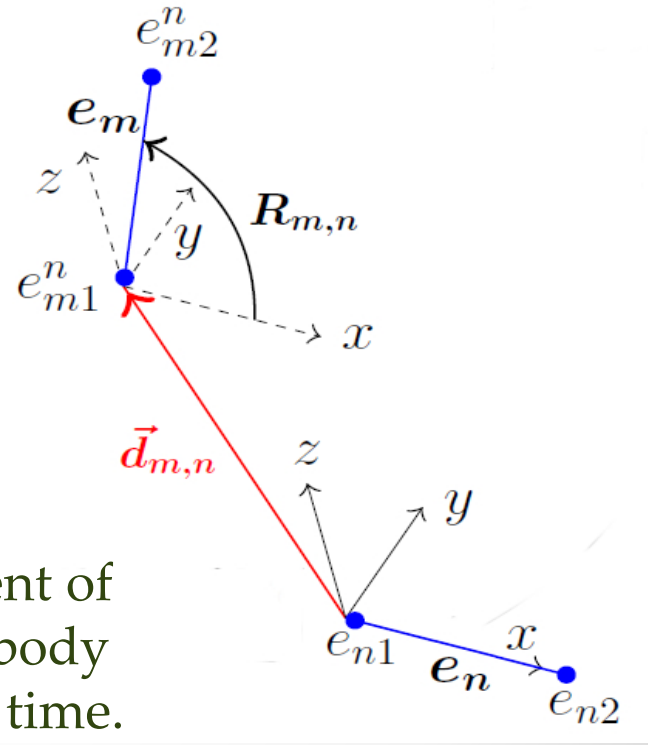
- We describe the relative geometry between two body parts (e_m, e_n) using the rigid body transformation (measured in the local coordinate system attached to e_n) required to take e_n to the position of e_m.

$$\begin{bmatrix} e^n_{m1}(t) & e^n_{m2}(t) \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} R_{m,n}(t) & \vec{d}_{m,n}(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e_{n1}(t) & s_{m,n} * e_{n2}(t) \\ 1 & 1 \end{bmatrix}$$

Element of the special Euclidean group SE(3)

Rigid body transformation varies with time.

Scaling factor: Independent of time since lengths of the body parts do not change with time.



Special Euclidean Group *SE(3)*

- $SE(3)$ is the set of all 4×4 matrices of the form $P = \begin{bmatrix} R & \vec{d} \\ 0 & 1 \end{bmatrix}$, where $R \in SO(3)$ and $\vec{d} \in \mathcal{R}^3$.
- The group $SE(3)$ is a smooth 6-dimensional curved manifold.
- The tangent plane to $SE(3)$ at the identity matrix I_4 is known as the Lie algebra of $SE(3)$.
- Lie algebra $\mathfrak{se}(3)$ is a 6-dimensional vector space.
- The exponential map $exp_{SE(3)}: \mathfrak{se}(3) \to SE(3)$ and the logarithm map $log_{SE(3)}: SE(3) \to \mathfrak{se}(3)$ are given by

$$exp_{SE(3)}(B) = \boldsymbol{e}^B, \quad log_{SE(3)}(P) = \boldsymbol{log}(P),$$

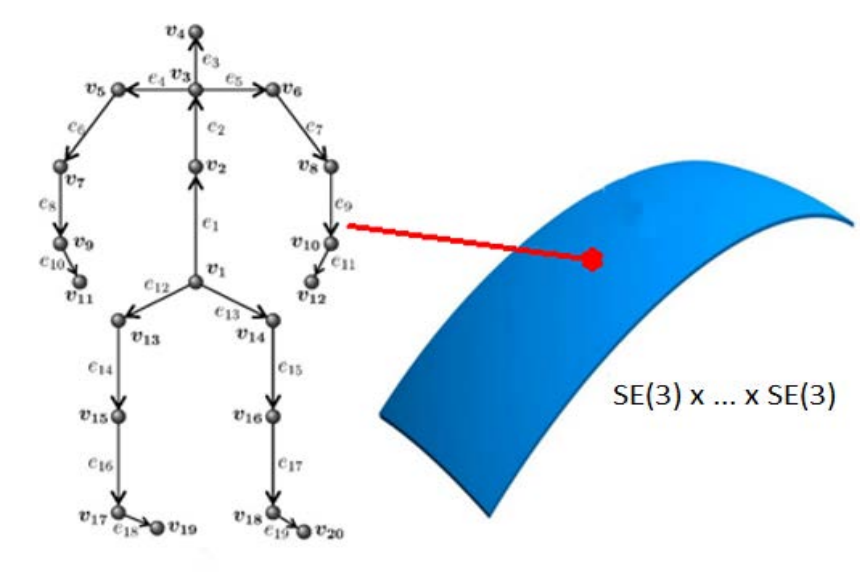
where $\boldsymbol{e}$ and $\boldsymbol{log}$ denote the usual matrix exponential and logarithm.

Proposed Skeletal Representation

- Human skeleton is described using the relative 3D geometry between all pairs of body parts.

$$\left\{P_{m,n}(t) = \begin{bmatrix} R_{m,n}(t) & \vec{d}_{m,n}(t) \\ 0 & 1 \end{bmatrix} : m \neq n, 1 \leq m, n \leq 19\right\} \in SE(3) \times \cdots \times SE(3).$$

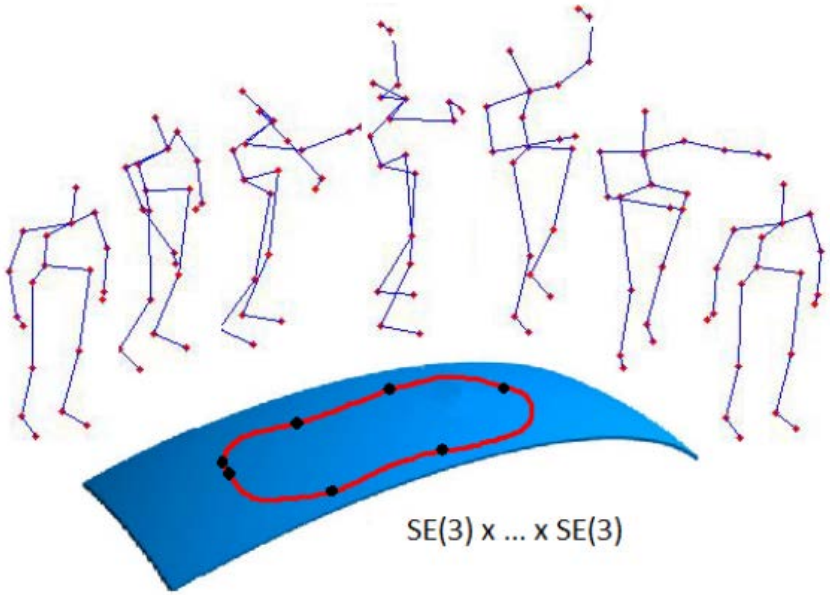
Lie group obtained by combining multiple *SE(3)* using the standard direct product $\times$.



Proposed Action Representation

- Using the proposed skeletal representation, a skeletal sequence can be represented as a curve in the Lie group $SE(3) \times \cdots \times SE(3)$:

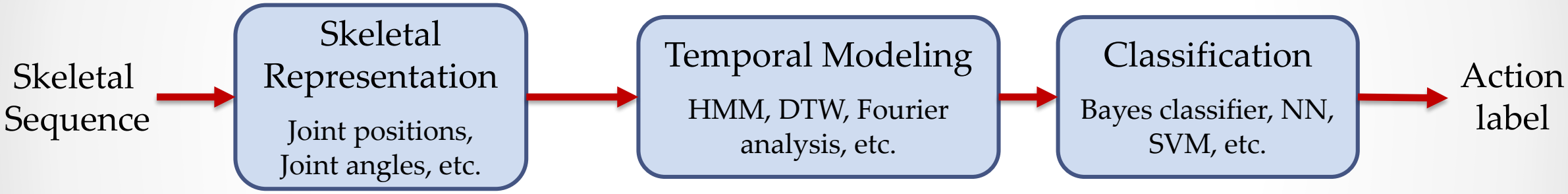
$$\left\{ \left\{ P_{m,n}(t) \mid m \neq n, 1 \leq m, n \leq 19 \right\}, t \in [0, T] \right\}.$$



Proposed Action Representation

- Classification of the curves in $SE(3) \times \cdots \times SE(3)$ into different action categories is a difficult task due to the non-Euclidean nature of the space.
- Standard classification approaches like support vector machines (SVM) and temporal modeling approaches like Fourier analysis are not directly applicable to this space.
- To overcome these difficulties, we map the curves from the Lie group $SE(3) \times \cdots \times SE(3)$ to its Lie algebra $\mathfrak{se}(3) \times \cdots \times \mathfrak{se}(3)$, which is a vector space, using the logarithm map.
- We perform action recognition by classifying the vector space curves.

Skeleton-based Action Recognition



Overview of a typical skeleton-based action recognition system

Temporal Modeling and Classification

- Action classification is a difficult task due to various issues like rate variations and noise.
- To handle rate variations, we use Dynamic Time Warping (DTW).
- To handle noise, we use the Fourier temporal pyramid (FTP) representation proposed by [Wang 2012].
- We use linear SVM with Fourier temporal pyramid representation for final classification.

J. Wang, Z. Liu, Y. Wu and J. Yuan, "Mining Actionlet Ensemble for Action Recognition with Depth Cameras", In CVPR, 2012.

Computation of Nominal Curves using DTW

Input: Curves $\mathfrak{C}_1(t), \ldots, \mathfrak{C}_J(t)$ at $t = 0, 1, \ldots, T$.
Maximum number of iterations max and threshold δ.

Output: Nominal curve $\mathfrak{C}(t)$ at $t = 0, 1, \ldots, T$.

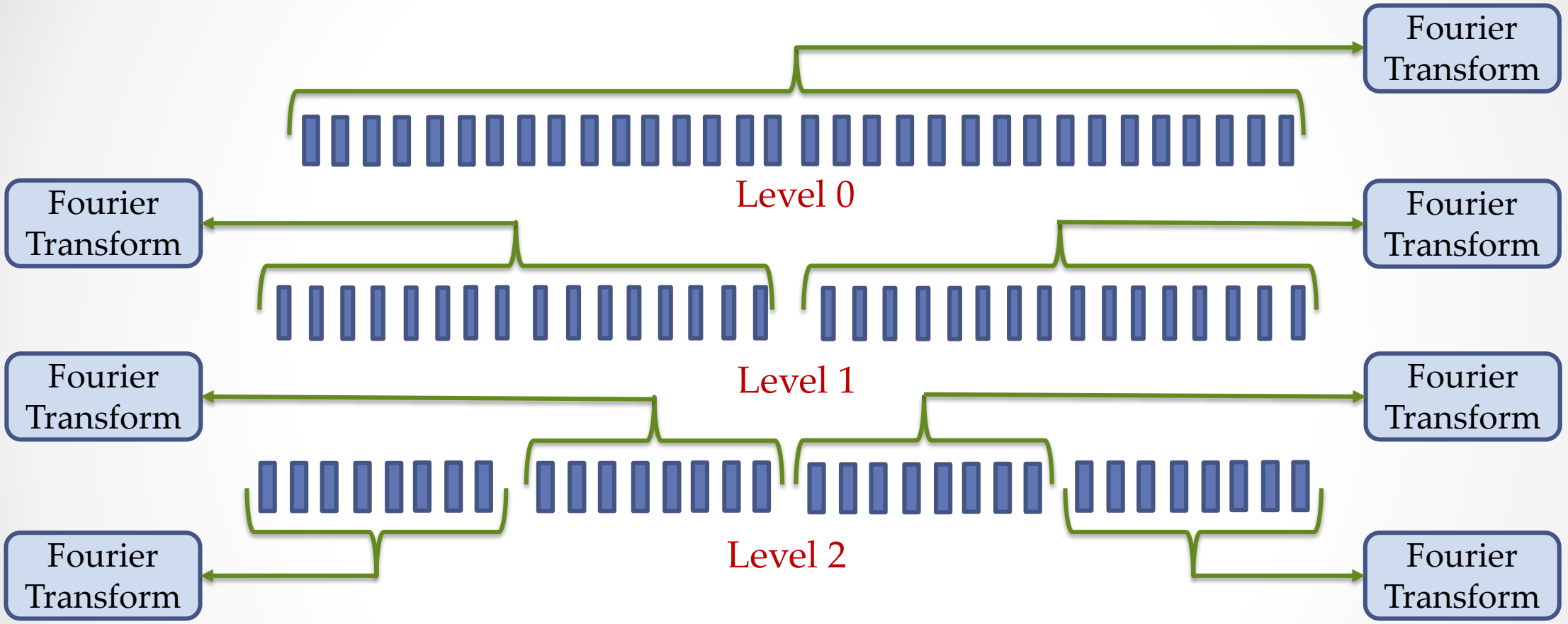
Initialization: $\mathfrak{C}(t) = \mathfrak{C}_1(t)$, iter = 0.
while iter $< max$
  Warp each curve $\mathfrak{C}_j(t)$ to the nominal curve $\mathfrak{C}(t)$ using DTW with squared Euclidean distance to get a warped curve $\mathfrak{C}_j^w(t)$.

  Compute a new nominal $\mathfrak{C}'(t)$ using

  $\mathfrak{C}'(t) = \frac{1}{J} \sum_{j=1}^{J} \mathfrak{C}_j^w(t)$.

  if $\sum_{t=0}^{T} \|\mathfrak{C}'(t) - \mathfrak{C}(t)\|_2^2 \leq \delta$ ($\|\cdot\|_2$ denotes ℓ_2 norm)
    break
  end
  $\mathfrak{C}(t) = \mathfrak{C}'(t)$; iter = iter + 1;
end

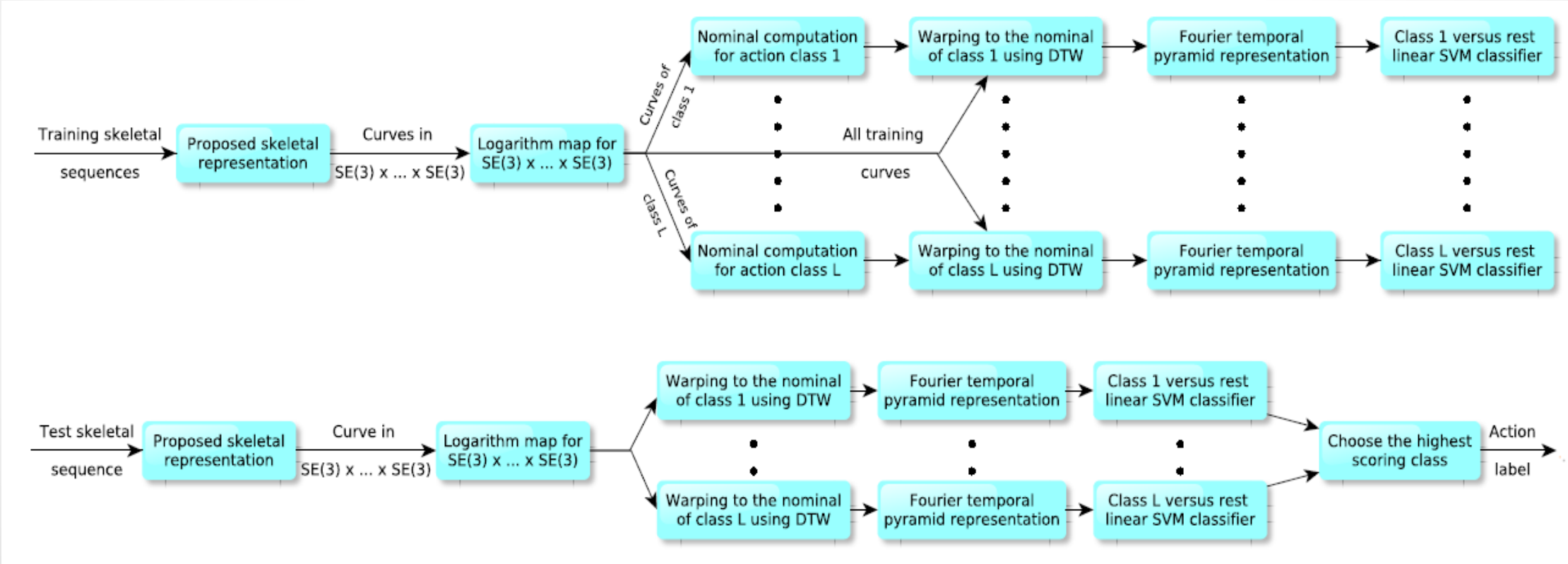
Fourier Temporal Pyramid Representation



Magnitude of the low frequency Fourier coefficients from each level are used to represent a time sequence.

J. Wang, Z. Liu, Y. Wu and J. Yuan, "Mining Actionlet Ensemble for Action Recognition with Depth Cameras", In CVPR, 2012.

Overview of the Proposed Approach



Experiments: Datasets

G3D-Gaming Action dataset

- Total 663 action sequences
- 10 subjects
- 20 actions

Punch right, Punch left, Kick right, Kick left, Defend, Golf swing, Tennis serve, Tennis swing forehand, Tennis swing backhand, Walk, Run, Jump, Climb, Throw bowling ball, Aim and fire gun, Crouch, Steer, Wave, Flap, Clap

UTKinect-Action dataset

- Total 199 action sequences
- 10 subjects
- 10 actions

Walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap

Florence3D-Action dataset

- Total 215 action sequences
- 9 actions
- 10 subjects

Wave, drink, answer phone, clap, sit down, stand up, read watch, bow, tie shoe lace

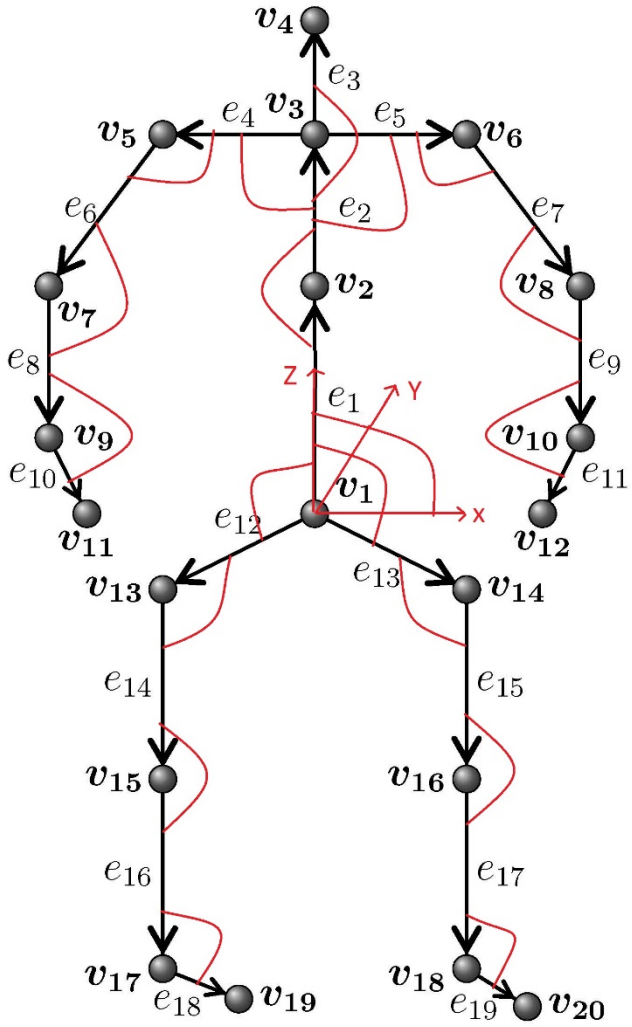
Alternative Representations for Comparison

Joint positions (JP):

Concatenation of the 3D coordinates of the joints.

Pairwise relative positions of the joints (RJP):

Concatenation of the 3D vectors $\overrightarrow{v_i\, v_j},\ 1 \leq i < j \leq 20$.



Joint angles (JA):

Concatenation of the quaternions corresponding to the joint angles (shown using red arcs in the figure).

Individual body part locations(BPL):

Each body part e_m is represented as a point in $SE(3)$ using its relative 3D geometry with respect to the global x-axis.

Comparison between Skeletal Representations

- Half of the subjects were used for training and the other half were used for testing.

Average classification accuracy

Dataset	JP	RJP	JA	BPL	Proposed
G3D Gaming Action	87.28	90.03	86.25	87.40	**91.60**
UTKinect Action	95.08	95.48	94.07	94.58	**97.20**
Florence3D Action	85.26	85.17	80.45	81.38	**90.71**

Comparison with State-of-the-art

UTKinect Action	
Hanklets	86.76
Random forests	87.90
Histograms of 3D joints	90.92
Motion trajectories	91.50
Elastic functional coding	94.87
Proposed approach	**97.20**

Florence3D Action	
Multi-part bag-of-poses	82.00
Motion trajectories	87.04
Elastic functional coding	89.67
Proposed approach	**90.71**

G3D-Gaming	
RBM+HMM	86.40
Proposed approach	**91.60**

Average classification accuracy

Scale-invariant Skeletal Representation

- The size/scale of the skeleton varies from subject to subject.
- We handled these scale variations by normalizing all the skeletons to a reference size.
- If we use only the 3D rotations between body parts instead of full rigid body transformations, we get a scale-invariant skeletal representation.
- Since 3D rotations are members of the special orthogonal group $SO(3)$, this scale-invariant representation becomes a point in the Lie group $SO(3) \times \cdots \times SO(3)$.

Special Orthogonal Group $SO(3)$

- $SO(3) = \{R \in \mathcal{R}^{3\times 3} : R'R = I, \det(R) = 1\}$.
- The group $SO(3)$ is a smooth 3-dimensional curved manifold.
- The tangent plane to $SO(3)$ at the identity matrix I_3 is known as the Lie algebra of $SO(3)$.
- Lie algebra $\mathfrak{so}(3)$ is a 3-dimensional vector space.
- The exponential map $exp_{SO(3)}: \mathfrak{so}(3) \to SO(3)$ and the logarithm map $log_{SO(3)}: SO(3) \to \mathfrak{so}(3)$ are given by

$$exp_{SO(3)}(B) = \boldsymbol{e}^B, \; log_{SO(3)}(R) = \boldsymbol{log}(R),$$

where $\boldsymbol{e}$ and $\boldsymbol{log}$ denote the usual matrix exponential and logarithm.

Action Recognition with $SO(3)$ Representation

- Similar to the case of $SE(3)$, we mapped the action curves from $SO(3) \times \cdots \times SO(3)$ to its Lie algebra using the logarithm map

- Performed action recognition using DTW, FTP and SVM.

Average classification accuracy

	G3D-Gaming	UTKinect Action	Florence3D Action
$SE(3) \times \cdots \times SE(3)$	91.60	97.20	90.71
$SO(3) \times \cdots \times SO(3)$	91.48	96.78	90.52

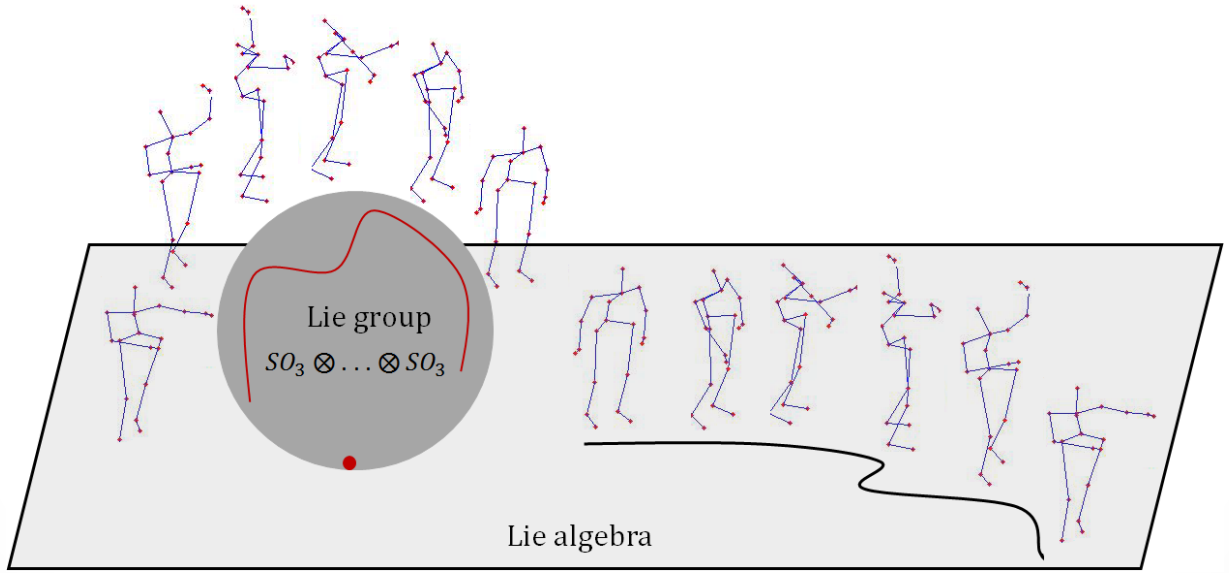
Quaternions and Dual Quaternions

- Rotations can be represented using quaternions instead of $SO(3)$.
- Rigid body transformations can be represented using dual quaternions instead of $SE(3)$.
- We performed experiments with quaternions and dual quaternions and got slightly better results on some of the datasets.

Summary

- We introduced relative 3D geometry-based skeletal representations for action recognition.
- $SE(3)$ was used to represent 3D rigid body transformations and $SO(3)$ was used to represent 3D rotations.
- We mapped the action curves from the Lie groups to the corresponding Lie algebras using the logarithm map.
- Action recognition was performed using a combination of DTW, FTP and linear SVM.
- The proposed representations outperform many existing skeletal representations.

Rolling the Special Orthogonal Group for Recognizing Human Actions from 3D Skeletal Data



$SO(3)$–based Scale Invariant Representation

- In the first part, we obtained a scale-invariant skeletal representation by using only the 3D rotations to represent the relative geometry between various body parts.

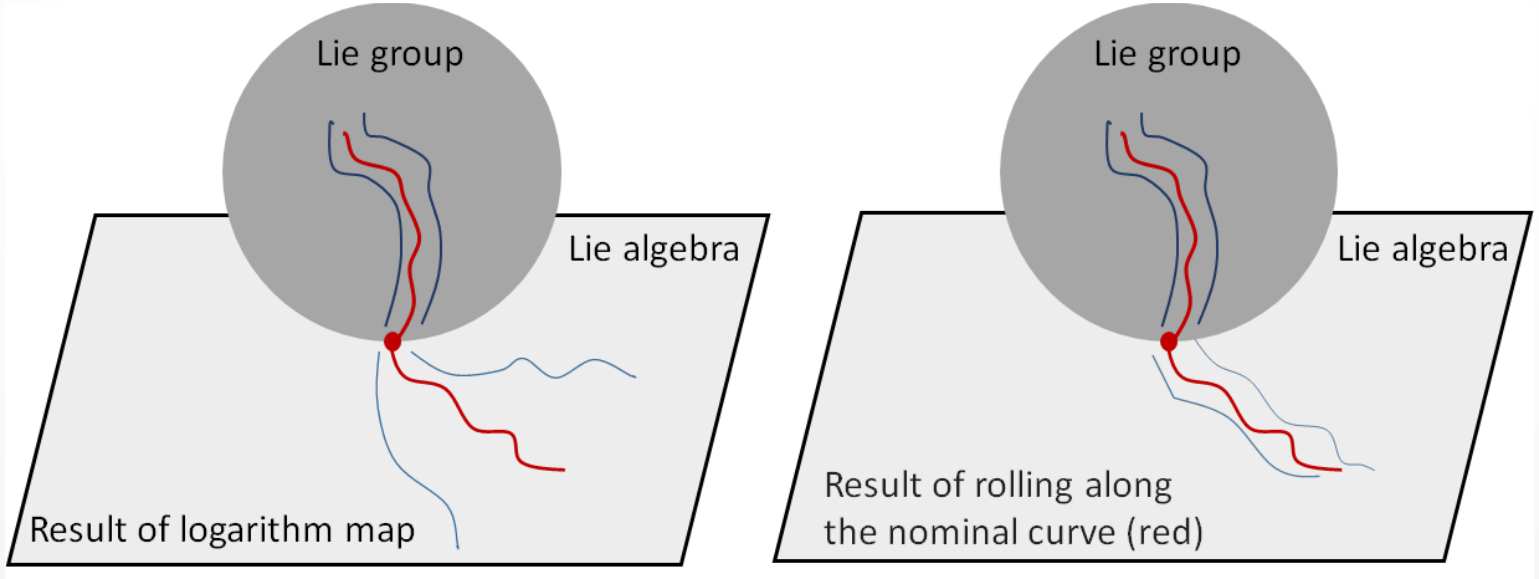
- We represented skeletons as points and actions as curves in the Lie group $SO(3) \times \cdots \times SO(3)$.

- We mapped the action curves from the Lie group to the Lie algebra using the logarithm map and performed action recognition by classifying the Lie algebra curves.

(Rolling + Logarithm Map) vs Logarithm map

- Mapping the action curves to the Lie algebra using the Logarithm map introduces distortions in the action curves.

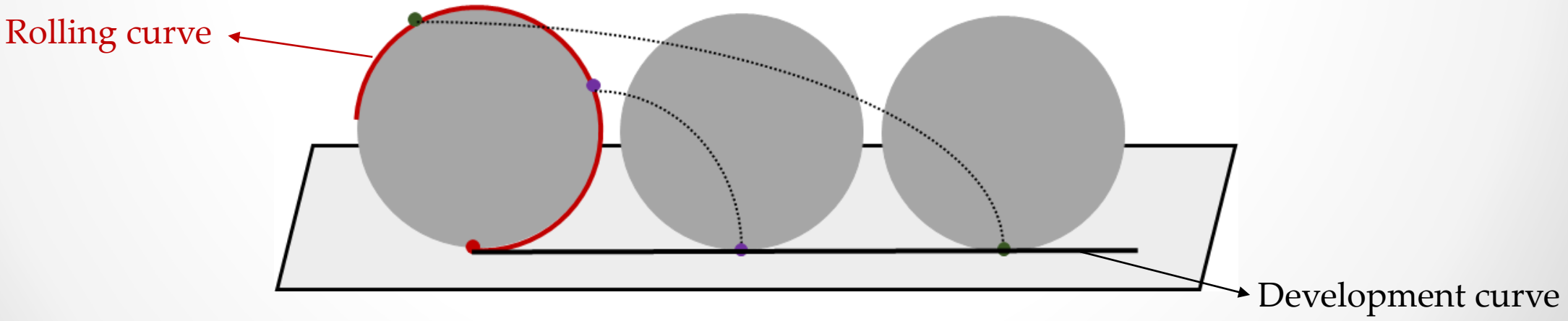
- We combine the logarithm map with rolling to reduce the distortions.



Rolling Motion

- For two m-dimensional Riemannian manifolds M and M', both embedded in the same ambient Euclidean space $R^n (n \geq m)$, the rolling motion describes how M rolls over M' as a rigid body without slip and twist.

- Classical example: Rolling of 2-dimensional sphere over the tangent plane at a point.



Rolling $SO(3)$ on a Tangent Plane

- Theorem: Let $\{\Omega(t) \in \mathfrak{so}(3) \mid t \in [0,T]\}$ be any continuous curve in the Lie algebra of $SO(3)$. For some $R_0 \in SO(3)$, let $\mathrm{C(t)} = (U(t),\ V(t),\ X(t))$ be the solution of

$$\frac{d}{dt}X(t) = \Omega(t)R_0, \qquad \frac{d}{dt}U(t) = -\frac{1}{2}\Omega(t)U(t), \qquad \frac{d}{dt}V(t) = \frac{1}{2}R_0^T\Omega(t)R_0V(t),$$

satisfying $\mathrm{C}(0) = (I,\ I,\ 0)$. Then the action of $C(t)$ on $SO(3)$ defined as

$$(U,V,X) \circ Z = UZV^T + X;\ Z \in SO_3$$

results in rolling of $SO(3)$ over the tangent plane at R_0 with the rolling curve given by

$$\alpha(t) = U(t)^T R_0 V(t) \in SO(3).$$

Knut Hüper and F. Silva Leite. On the geometry of rolling and interpolation curves on Sn, SOn, Grassmann manifolds. Journal of dynamical and Control Systems, 13(4):467-502, 2007.

Rolling $SO(3)$ on a Tangent Plane

- The above theorem states that every continuous curve in the Lie algebra of $SO(3)$ defines a rolling motion over every tangent plane.
- It also describes how to compute the rolling motion and the corresponding rolling curve starting from the Lie algebra curve.
- However, it does not say anything about how to compute the rolling motion starting from the rolling curve.
- In this work, we are interested in rolling $SO(3)$ along specific rolling curves.

Rolling $SO(3)$ along a given Rolling Curve

- Theorem: Let $\{B_0, B_1, \dots, B_T\}$ be a (discrete) curve in $SO(3)$. Let $\Omega_1, \dots, \Omega_T$ be T skew-symmetric matrices defined recursively using

$$\Omega_n = \log\left(e^{-\frac{\Omega_{n-1}}{2}} \dots e^{-\frac{\Omega_1}{2}} B_n B_1^T e^{-\frac{\Omega_1}{2}} \dots e^{-\frac{\Omega_{n-1}}{2}}\right).$$

Let $C(t) = (U(t), V(t), X(t))$ be a curve defined as

$$U(t) = e^{-\frac{(t-n+1)\Omega_n}{2}} e^{-\frac{\Omega_{n-1}}{2}} \dots e^{-\frac{\Omega_1}{2}}, \ \mathrm{V}(t) = B_0^T e^{\frac{(t-n+1)\Omega_n}{2}} e^{\frac{\Omega_{n-1}}{2}} \dots e^{\frac{\Omega_1}{2}} B_0,$$

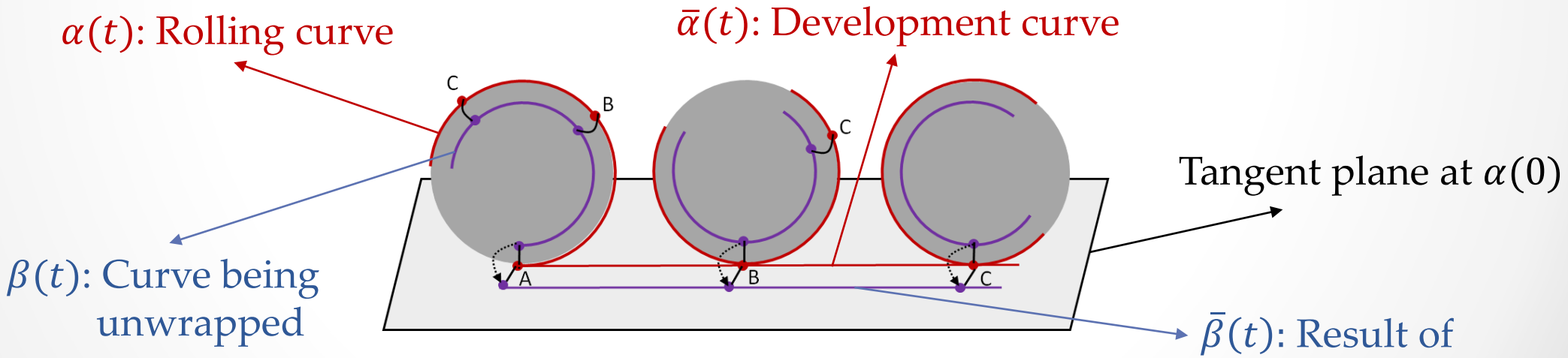
$$X(t) = B_0 \sum_{i=1}^{n-1} \Omega_i + (t-n+1)\Omega_n B_0, \ t \in [n-1, n], \ n = 1,2, \dots, T.$$

Then, the action of $C(t)$ on SO_3 defined as $(U, V, X) \circ Z = UZV^T + X;\ Z \in SO_3$ results in rolling of SO_3 over the tangent plane at B_0 along the curve $B(t)$.

Unwrapping while Rolling

- We can map curves from the Lie group $SO(3)$ to its Lie algebra by unwrapping them with the logarithm map while rolling.

$$\bar{\beta}(t) = \mathbf{log}\left[U(t)\left(\beta(t) - \alpha(t)\right)V(t)^T\alpha(0)^T + I\right]\alpha(0) + \left[U(t)\alpha(t)V(t)^T + X(t)\right]$$



Unwrapping the blue curve onto the Lie algebra while rolling the Lie group along the red curve.

Unwrapping while Rolling

- Unwrapping (using the logarithm map) while rolling preserves the distances between the curves that are being unwrapped and the rolling curve.

- Theorem: Let $\{\alpha(t), \beta(t) \in SO(3) \mid t \in [0, T]\}$ be two curves. Let $\bar{\alpha}(t)$ and $\bar{\beta}(t)$ respectively be the curves obtained by unwrapping (using the logarithm map) $\alpha(t)$ and $\beta(t)$ while rolling $SO(3)$ over the tangent space at $\alpha(0)$ along the curve $\alpha(t)$. Then, we have

$$d_{T_{\alpha(0)}SO(3)}\left(\bar{\alpha}(t), \bar{\beta}(t)\right) = d_{SO(3)}\left(\alpha(t), \beta(t)\right) \forall t.$$

  - $d_{T_{\alpha(0)}SO(3)}$ represents the standard Euclidean distance in the tangent space $T_{\alpha(0)}SO(3)$
  - $d_{SO(3)}$ represents the geodesic distance in $SO(3)$

Proposed Action Recognition Approach

- Representation: We represent skeletons as points and actions as curves in the Lie group $SO(3) \times \cdots \times SO(3)$.
- DTW: For each action category, we compute a nominal curve using DTW and warp all the action curves to the nominal curves.
- Rolling: After DTW, the action curves are mapped to the Lie algebra by rolling $SO(3) \times \cdots \times SO(3)$ along the nominal curves. Rolling of $SO(3) \times \cdots \times SO(3)$ is performed by rolling each $SO(3)$ individually.
- Classification: We first convert each Lie algebra curve into a feature vector by either concatenating all the temporal samples or by using the FTP representation, and then classify these vectors using a one-vs-all linear SVM classifier.

Experiments: Datasets

G3D-Gaming Action dataset

- Total 663 action sequences
- 10 subjects
- 20 actions

Punch right, Punch left, Kick right, Kick left, Defend, Golf swing, Tennis serve, Tennis swing forehand, Tennis swing backhand, Walk, Run, Jump, Climb, Throw bowling ball, Aim and fire gun, Crouch, Steer, Wave, Flap, Clap

MSR-Pairs dataset

- Total 353 action sequences
- 10 subjects
- 12 actions

Pick up a box, put down a box, push a chair, pull a chair, wear a hat, take off a hat, put on a backpack, take off a backpack, lift a box, place a box

Florence3D-Action dataset

- Total 215 action sequences
- 9 actions
- 10 subjects

Wave, drink, answer phone, clap, sit down, stand up, read watch, bow, tie shoe lace

Results (Cross Subject Test Setting)

Average classification accuracy (concatenated representation)

Dataset	Logarithm map	Unwrapping while rolling
Florence3D	86.83	89.82
MSR Pairs	92.96	94.09
G3D	87.89	87.77

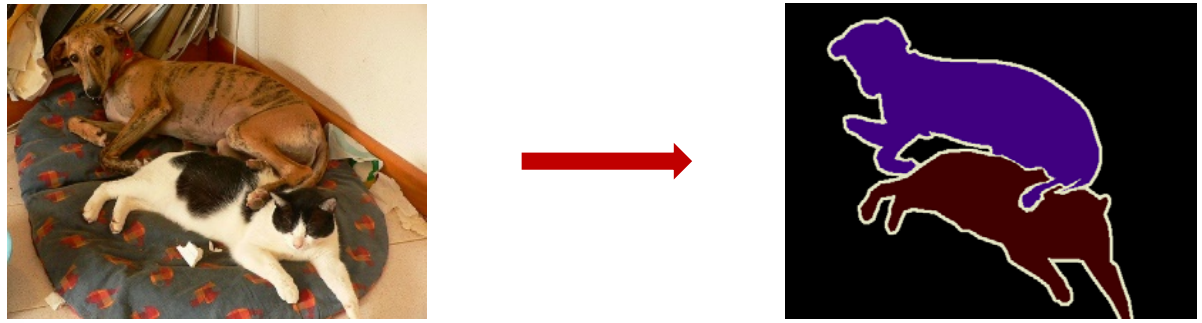
Average classification accuracy (FTP representation)

Dataset	Logarithm map	Unwrapping while rolling
Florence3D	90.89	91.40
MSR Pairs	94.10	94.67
G3D	91.48	91.42

Summary

- We represented skeletons as points and actions as curves in the Lie group $SO(3) \times \cdots \times SO(3)$.
- Instead of directly using the logarithm map, we combined it with rolling to map the action curves from $SO(3) \times \cdots \times SO(3)$ to its Lie algebra.
- We showed how to compute the rolling motion corresponding to a given rolling curve on $SO(3)$.
- We showed that rolling reduces the distortions in the action curves while mapping them to the Lie algebra, and helps in improving the classification accuracy.

Gaussian Conditional Random Field Network for Semantic Segmentation



Deep Neural Networks

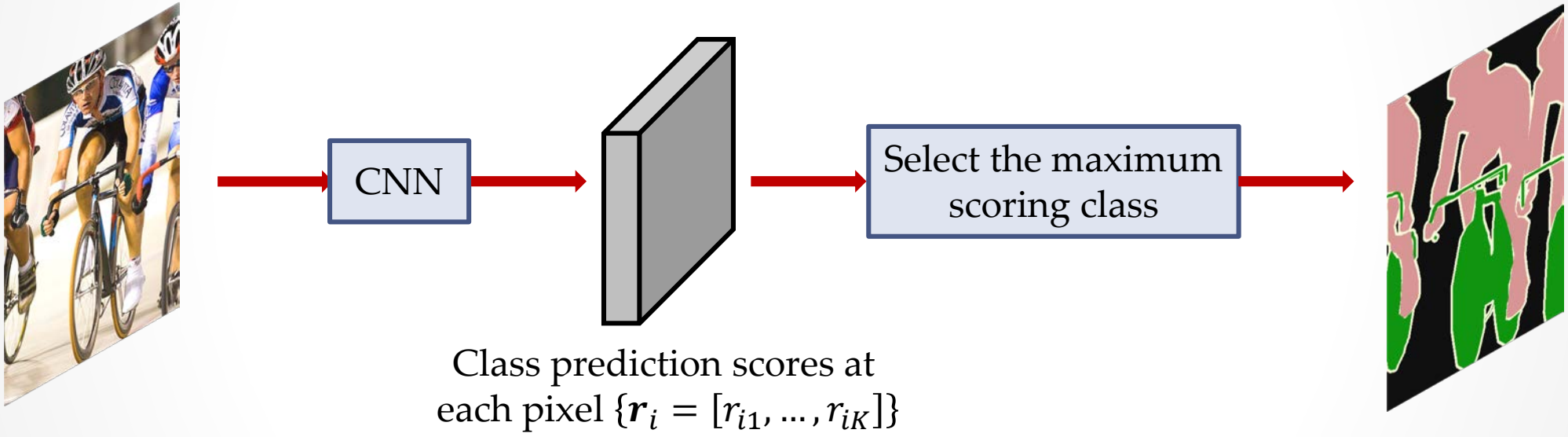
- Deep neural networks have been successfully used in various image processing and computer vision applications:
  - Image denoising, deconvolution and super-resolution
  - Depth estimation
  - Object detection and recognition
  - Semantic segmentation
  - Action detection and recognition

- Their success can be attributed to several factors:
  - Ability to represent complex input-output relationships
  - Feed-forward nature of their inference (no need to solve an optimization problem during run time)
  - Availability of large training datasets and fast computing hardware like GPUs

What is missing in these standard deep neural networks?

CNN-based Semantic Segmentation

- Standard deep networks do not explicitly model the interactions between output variables.



- Modeling the interactions between output variables is very important for structured prediction tasks such as semantic segmentation.

CNN + Discrete CRF

- CRF as a post-processing step

C. Farabet, C. Couprie, L. Najman, and Y. LeCun. *Learning Hierarchical Features for Scene Labeling*. IEEE Trans. Pattern Anal. Mach. Intell., 35(8):1915–1929, 2013.

S. Bell, P. Upchurch, N. Snavely, and K. Bala. *Material Recognition in the Wild with the Materials in Context Database*. In CVPR, 2015.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. *Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs*. In ICLR, 2015.

- Joint training of CNN and CRF

S. Zheng, S. Jayasumana, B. R.-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. *Conditional Random Fields as Recurrent Neural Networks*. In ICCV, 2015.

Discrete CRF vs Gaussian CRF

- Discrete CRF is a natural fit for discrete labeling tasks such as semantic segmentation.
- Inference techniques do not have optimality guarantees.
- Exact inference is possible in the case of a Gaussian CRF.
- Not clear if Gaussian CRF is a good fit for discrete labeling tasks.

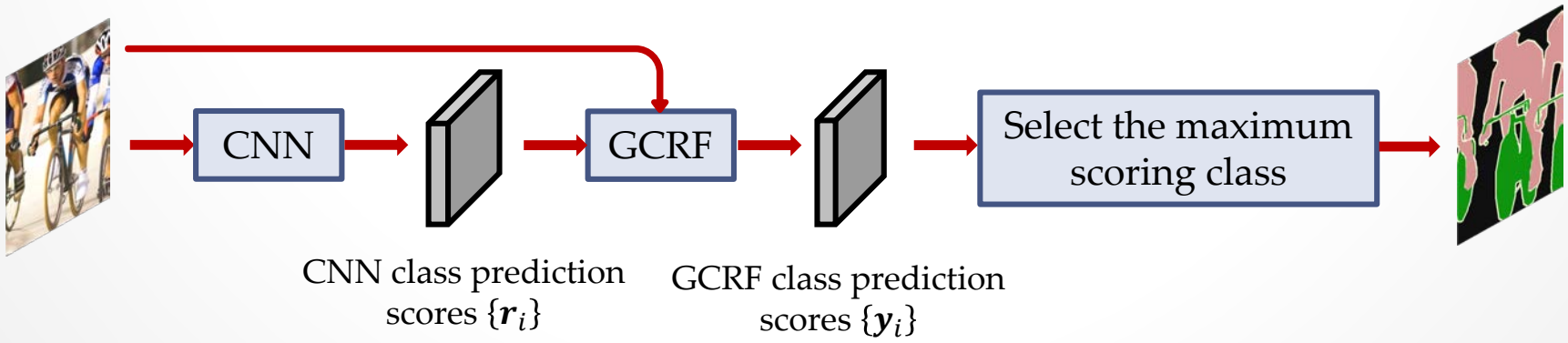
Should we use a better model with approximate inference
or an approximate model with better inference?

Gaussian CRF for Semantic Segmentation

- We use a Gaussian CRF model on top of a CNN to explicitly model the interactions between the class labels at different pixels.
- Semantic segmentation is a discrete labeling task.
- To use a Gaussian CRF model, we replace each discrete output variable with a vector of K continuous variables:

$$\boldsymbol{y}_i = [y_{i1}, \ldots, y_{iK}] \in R^K.$$

  - y_{ik} represents the score for k^{th} class at i^{th} pixel.
  - Class label for i^{th} pixel is given by $\underset{k}{\text{argmax}}\, y_{ik}$.

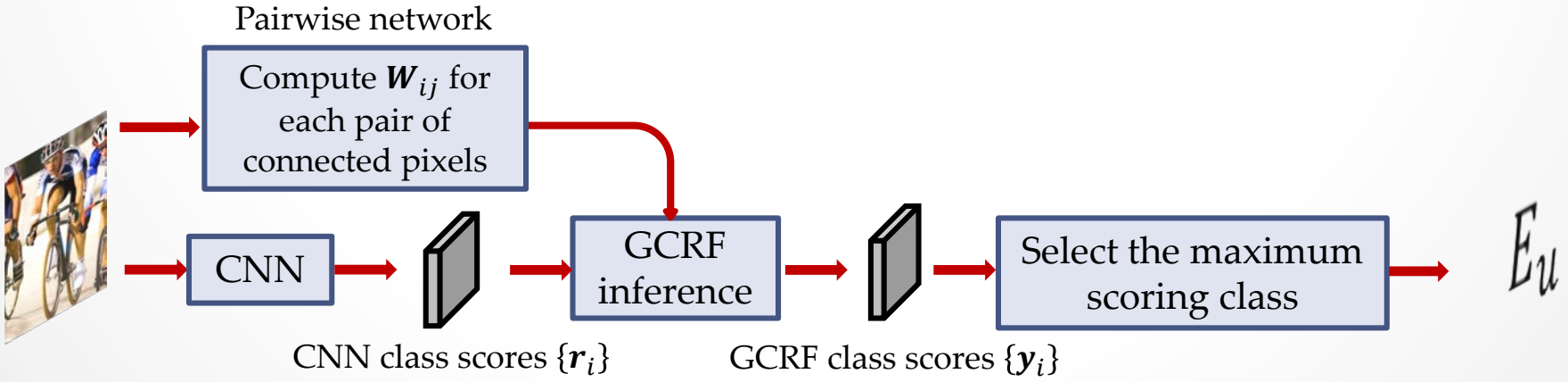


Gaussian CRF Model for Semantic Segmentation

- Let $\boldsymbol{X}$ represent the input image, and $\boldsymbol{Y}$ represent the output (K-dimensional vector at each pixel).
- We model the conditional probability density $P(\boldsymbol{Y}|\boldsymbol{X})$ as a Gaussian distribution given by $P(\boldsymbol{Y}|\boldsymbol{X}) \propto e^{-\frac{1}{2}E(\boldsymbol{Y}|\boldsymbol{X})}$, where

$$E(\boldsymbol{Y}|\boldsymbol{X}) = \sum_i \underbrace{\|\boldsymbol{y}_i - \boldsymbol{r}_i(\boldsymbol{X};\theta_u)\|_2^2}_{E_u} + \sum_{ij} \underbrace{(\boldsymbol{y}_i - \boldsymbol{y}_j)^T \boldsymbol{W}_{ij}(\boldsymbol{X};\theta_p)(\boldsymbol{y}_i - \boldsymbol{y}_j)}_{E_p};\ \ \boldsymbol{W}_{ij} \geqslant 0.$$

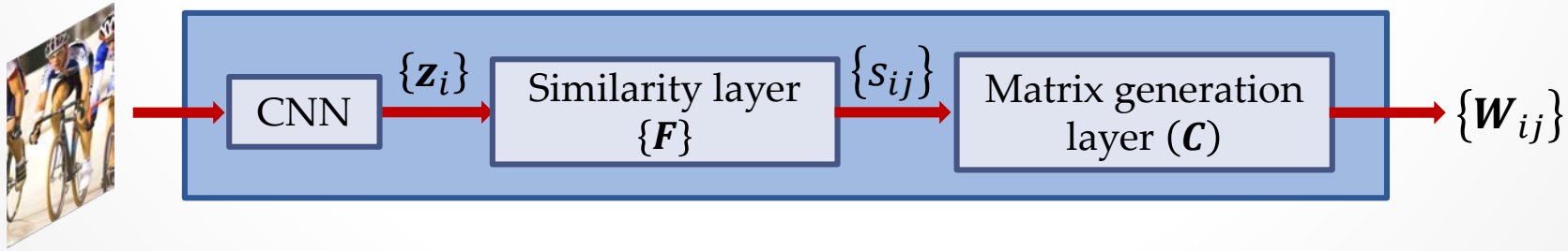
  - $\boldsymbol{r}_i(\boldsymbol{X};\theta_u)$ are the CNN class prediction scores, θ_u are the unary-CNN parameters.
  - $\boldsymbol{W}_{ij}(\boldsymbol{X};\theta_p)$ are the input-dependent parameters of the pairwise potential function E_p.



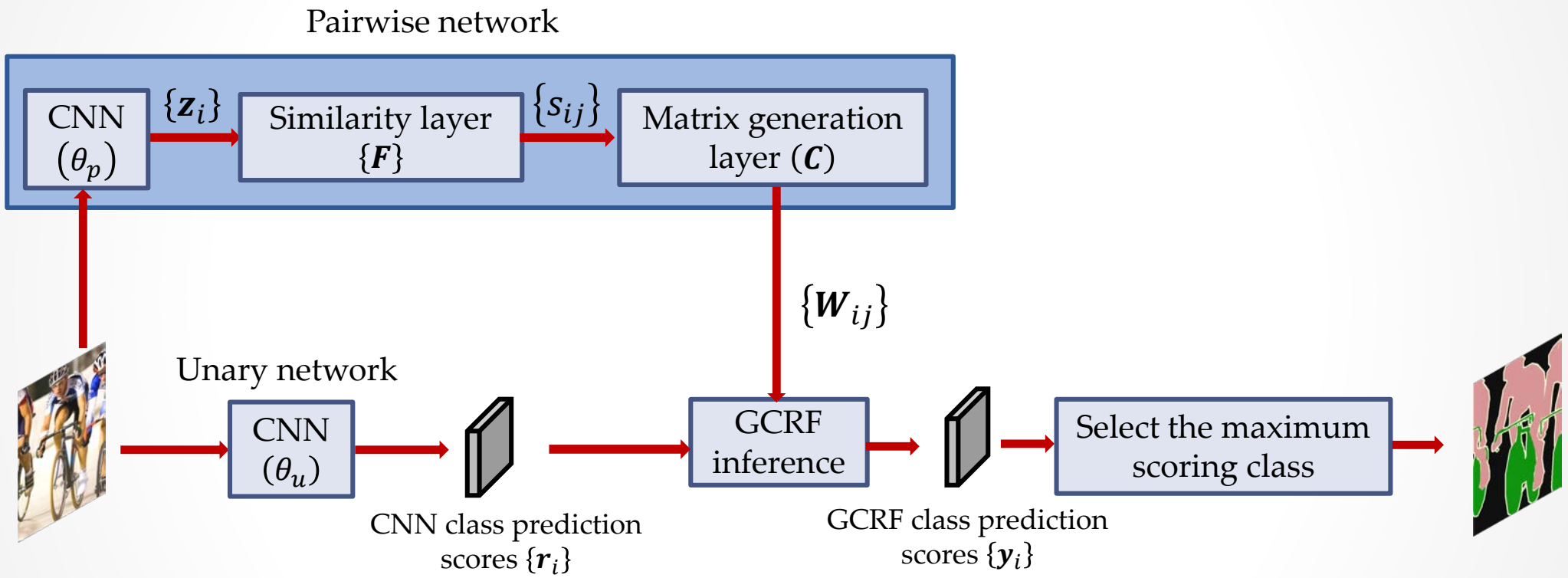
Pairwise Network

- We compute each $\boldsymbol{W}_{ij}$ as $\boldsymbol{W}_{ij} = s_{ij}\boldsymbol{C}; \boldsymbol{C} \succcurlyeq 0$:
  - $s_{ij} \in [0,1]$ is a similarity measure between pixels i and j.
  - $\boldsymbol{C}$ is a parameter matrix that encodes the class compatibility information.

- The similarity measure s_{ij} is computed as $s_{ij} = e^{-(\boldsymbol{z}_i - \boldsymbol{z}_j)^T \boldsymbol{F} (\boldsymbol{z}_i - \boldsymbol{z}_j)}$:
  - $\boldsymbol{z}_i$ is a feature vector extracted at pixel i using a CNN.
  - $\boldsymbol{F} \succcurlyeq 0$ is a parameter matrix that defines a Mahalanobis distance function.



Gaussian CRF Network



GCRF Inference

- Given the unary network output $\{\boldsymbol{r}_i\}$ and the pairwise network output $\{\boldsymbol{W}_{ij}\}$, GCRF inference solves the following optimization problem:

$$\boldsymbol{Y}^* = \underset{\boldsymbol{Y}}{\operatorname{argmax}}\ P(\boldsymbol{Y}|\boldsymbol{X}) = \underset{\boldsymbol{Y}}{\operatorname{argmin}} \sum_i \|\boldsymbol{y}_i - \boldsymbol{r}_i\|_2^2 + \sum_{ij} (\boldsymbol{y}_i - \boldsymbol{y}_j)^T \boldsymbol{W}_{ij} (\boldsymbol{y}_i - \boldsymbol{y}_j).$$

  - Unconstrained quadratic program and hence be solved in closed form.
  - Closed form solution requires solving a linear system with number of variables equal to the number of pixels times the number of classes.

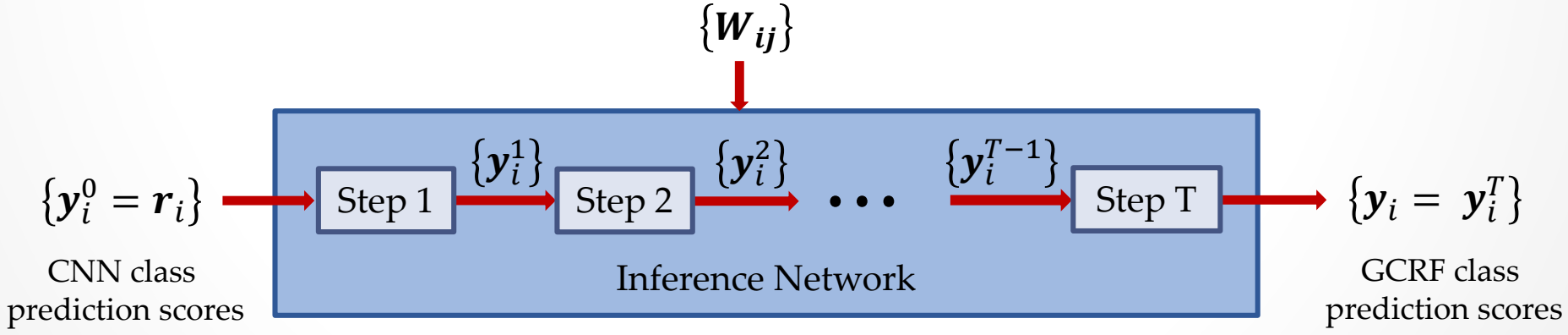
- Instead of exactly solving the full linear system, we perform approximate inference using the iterative Gaussian mean field procedure.

Gaussian Mean Field Inference

- We unroll the iterative Gaussian mean field (GMF) inference into a deep network.

- Parallel GMF inference: Update all the variables in parallel using

$$\boldsymbol{y}_i^{t+1} = \left(I + \sum_j \boldsymbol{W}_{ij}\right)^{-1} \left(\boldsymbol{r}_i + \sum_j \boldsymbol{W}_{ij}\boldsymbol{y}_j^t\right).$$



Convergence of GMF Inference

- Parallel GMF inference is guaranteed to converge to the global optimum if the precision matrix of the Gaussian distribution $P(\boldsymbol{Y}|\boldsymbol{X})$ is diagonal dominant.

- Imposing such constraints on $P(\boldsymbol{Y}|\boldsymbol{X})$ is very difficult and could restrict the model capacity in practice.

$$E(\boldsymbol{Y}|\boldsymbol{X}) = \sum_i \|\boldsymbol{y}_i - \boldsymbol{r}_i\|_2^2 + \sum_{ij} (\boldsymbol{y}_i - \boldsymbol{y}_j)^T \boldsymbol{W}_{ij} (\boldsymbol{y}_i - \boldsymbol{y}_j).$$

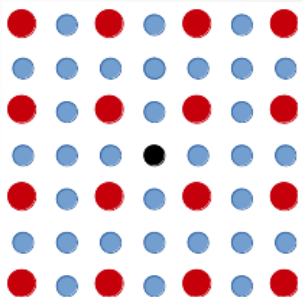
  - Making all the matrices $\boldsymbol{W}_{ij}$ diagonal satisfies the diagonal dominance constraint.
  - This is very restrictive as this removes the inter-class interactions across pixels.

- If we update the variables serially, then GMF inference will converge even without the diagonal dominance constraints.

- But serial updates are not practical since we have a huge number of variables.

Convergence of GMF Inference

- Ideally, we want to
  - update as many variables as possible in parallel
  - avoid diagonal dominance constraints
  - have convergence guarantee



- When using graphical models, each pixel is usually connected to every pixel within a spatial neighborhood.

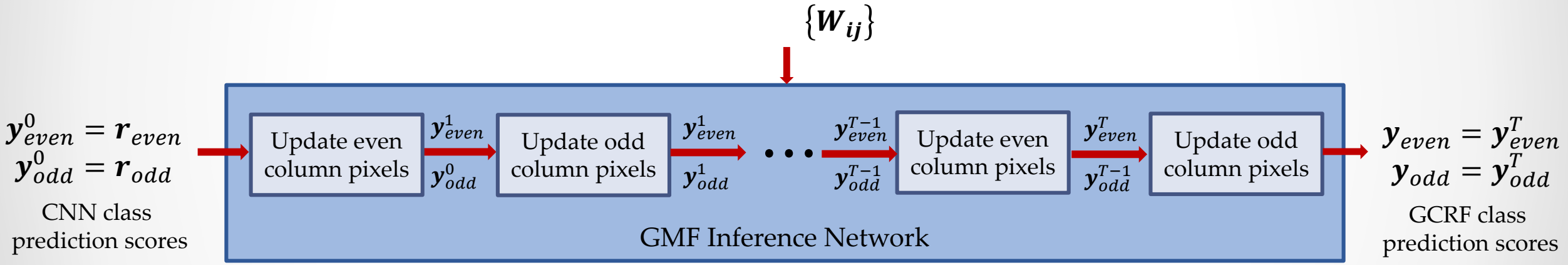
- We connect each pixel to every other pixel along both rows and columns within a spatial neighborhood.

- If we partition the image into even and odd columns, this connectivity ensures that there are no edges within the partitions.

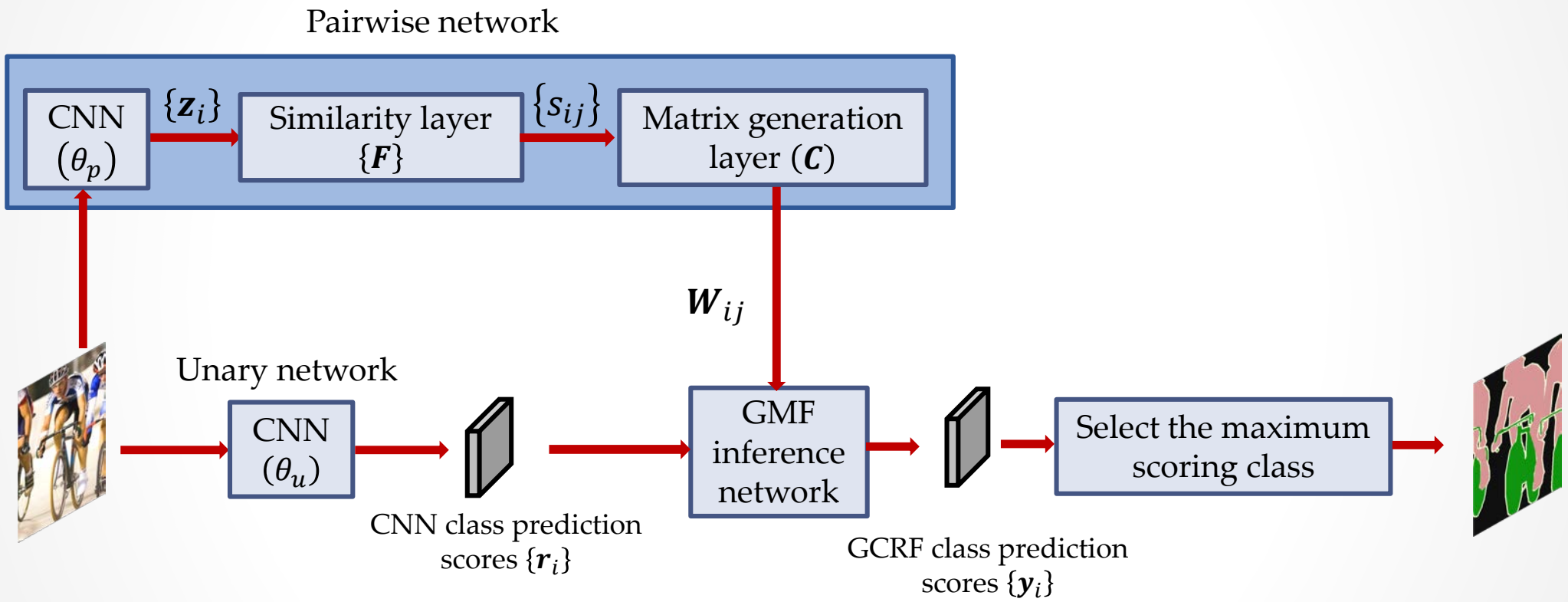
- We can update all even column pixels in parallel and all the odd column pixels in parallel and still have convergence guarantee without the diagonal dominance constraints.

GMF Inference Network

- Each layer of our network produces an output that is closer to the optimal solution compared to its input (unless the input itself is the optimal solution, in which case the output will be equal to the input).



GCRF Network



Training

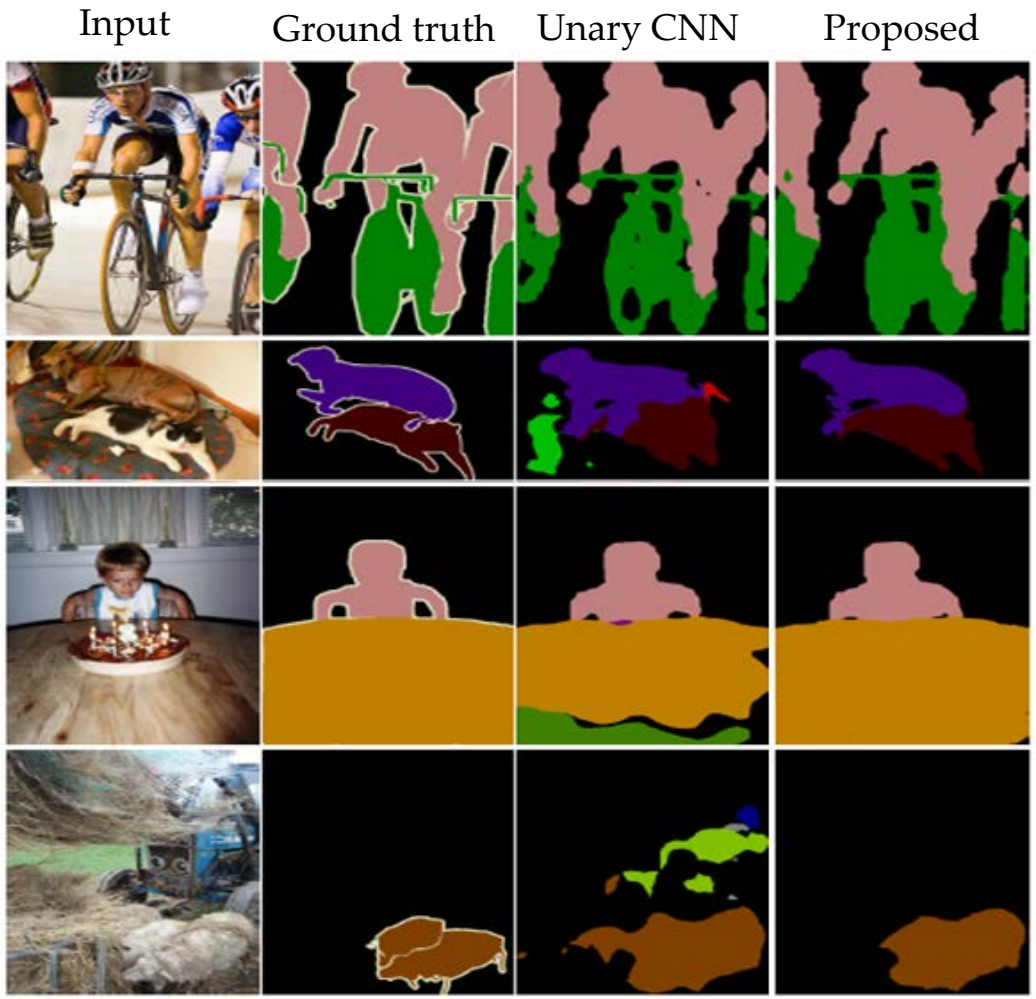
- CNNs were pre-trained on ImageNet dataset, pairwise network was pre-trained like a Siamese network at pixel level.
- Trained the entire GCRF network (unary, pairwise and GMF networks) end-to-end discriminatively.
- Training loss function: $L(\{\boldsymbol{y}_i, l_i\}) = -\frac{1}{N}\sum_{i=1}^{N} \min\left(0,\ y_{il_i} - \max_{k \neq l_i} y_{ik} - S\right)$.
  - l_i is the true class label of pixel i.
  - This cost function encourages the prediction score for the true class to be greater than the prediction scores of all the other classes by a margin S.
- Used standard back-propagation to compute the gradient of the network parameters.
- We have a constrained optimization because of the symmetry and positive semi-definiteness constraints on the parameter matrix $\boldsymbol{C}$.
- Parametrized $\boldsymbol{C}$ as $\boldsymbol{C} = \boldsymbol{R}\boldsymbol{R}^{\boldsymbol{T}}$ where $\boldsymbol{R}$ is a lower triangular matrix, and used stochastic gradient descent.

Experimental Results

- PASCALVOC2012 dataset: 10,582 training images and 1456 test images.
- Mean IOU score: 73.2 (better than the unary CNN by 6.2 points)

Method	bkg	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mean
MSRA-CFM [9]	87.7	75.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5	61.8
FCN-8s [29]	91.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Hypercolumns [18]	89.3	68.7	33.5	69.8	51.3	70.2	81.1	71.9	74.9	23.9	60.6	46.9	72.1	68.3	74.5	72.9	52.6	64.4	45.4	64.9	57.4	62.6
DeepLab CNN [7]	91.6	78.7	51.5	75.8	59.5	61.9	82.5	76.6	79.4	26.9	67.7	54.7	74.3	70.0	79.8	77.3	52.6	75.2	46.6	66.9	57.3	67.0
ZoomOut [30]	91.1	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3	69.6
Approaches that use CNNs and discrete CRFs																						
Deep structure models [27]	93.6	86.7	36.9	82.3	63.0	74.2	89.8	84.1	84.1	32.8	65.4	52.1	79.7	72.1	77.6	81.7	55.6	77.4	37.4	81.4	68.4	70.3
DeconvNet + CRF [31]	92.9	87.8	41.9	80.6	63.9	67.3	88.1	78.4	81.3	25.9	73.7	61.2	72.0	77.0	79.9	78.7	59.5	78.3	55.0	75.2	61.5	70.5
object clique potentials [36]	92.8	80.0	53.8	80.8	62.5	64.7	87.0	78.5	83.0	29.0	82.0	60.3	76.3	78.4	83.0	79.8	57.0	80.0	53.1	70.1	63.1	71.2
DeepLab CNN-CRF [7]	93.3	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	71.6
CRF-RNN [54]	94.0	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1	72.0
DeconvNet + FCN + CRF [31]	93.1	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
Proposed GCRF network	93.4	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1	73.2

Experimental Results



Experimental Results

Summary

- We proposed to use a Gaussian CRF model for the task of semantic image segmentation.
- We designed a new deep network, referred to GMF network, by unrolling the Gaussian mean field inference procedure.
- The proposed GMF network has the desired property that each of its layers produces an output that is closer to the optimal solution of the Gaussian CRF compared to its input.
- By combining the proposed GMF network with deep CNNs we proposed a new end-to-end trainable Gaussian conditional random field network.
- The proposed network performs better than various other approaches that use CNNs and discrete CRF models.

Gaussian Conditional Random Field Network for Non-blind Image Denoising

Image Denoising with Multilayer Perceptrons

- State-of-the-art MLP-based image denoising approach [Burger 2012]:
  - Decompose the noisy input image into $d \times d$ overlapping patches and denoise each patch using an MLP.
  - Obtain a clean image by placing the denoised patches at the locations of their noisy counterparts.
- What is missing in this MLP-based denoising approach?
  - MLPs do not explicitly model the input noise variance σ^2 and hence a single MLP cannot handle multiple noise levels.
  - [Burger 2012] trained different MLPs for different input noise levels.
  - Training a separate model for each specific noise level is not practical.
- Ideally, we would want a deep network that can handle a range of noise levels.
- We propose a new deep network architecture based on a Gaussian CRF model that can handle multiple input noise levels.

H. C. Burger, C. J. Schuler, and S. Harmeling. Image Denoising: Can Plain Neural Networks Compete with BM3D?, CVPR, 2012.

GCRF Model for Image Denoising

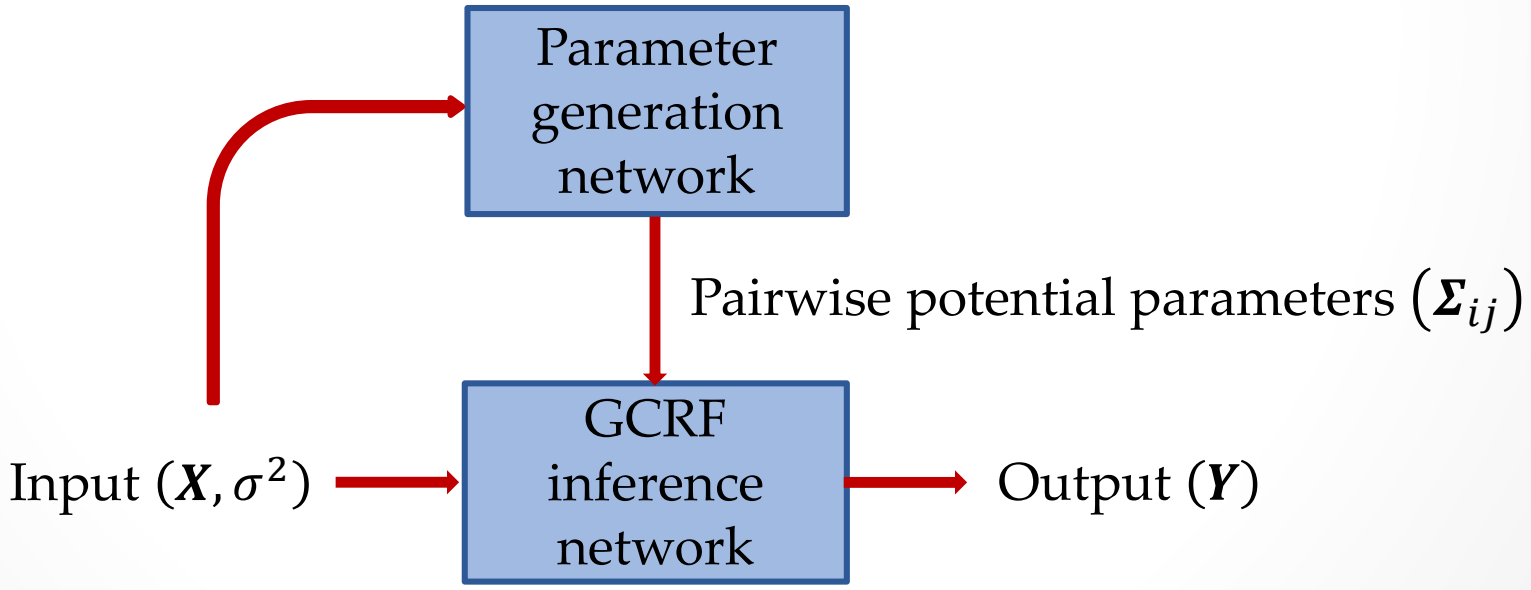
- Let $\boldsymbol{X}$ represent the noisy input image with noise variance σ^2, and $\boldsymbol{Y}$ represent the output image.
- We model the conditional probability density $P(\boldsymbol{Y}|\boldsymbol{X})$ as a Gaussian distribution given by $P(\boldsymbol{Y}|\boldsymbol{X}) \propto e^{-E(\boldsymbol{Y}|\boldsymbol{X})}$, where

$$E(\boldsymbol{Y}|\boldsymbol{X}) = \underbrace{\frac{1}{2\sigma^2}\sum_{ij}\big(\boldsymbol{Y}(i,j) - \boldsymbol{X}(i,j)\big)^2}_{\text{Quadratic unary potential}} + \underbrace{\frac{1}{2d^2}\sum_{ij}\boldsymbol{y}_{ij}^T\boldsymbol{G}^T\boldsymbol{\Sigma}_{ij}^{-1}\boldsymbol{G}\boldsymbol{y}_{ij}}_{\text{Quadratic pairwise potential}},\ \boldsymbol{\Sigma}_{ij} \succcurlyeq 0.$$

- $\boldsymbol{y}_{ij}$: column vector representing a $d \times d$ patch centered on pixel (i,j) in the image $\boldsymbol{Y}$.
- $\boldsymbol{\Sigma}_{ij}$: data-dependent parameters of the pairwise potential function
- $\boldsymbol{G}$: Mean subtraction matrix

GCRF Network for Image Denoising

- GCRF-based image denoising consists of the following two steps:
  - *Parameter generation*: Compute appropriate pairwise potential parameters $\boldsymbol{\Sigma}_{ij}$ based on the input image $\boldsymbol{X}$.
  - *Inference*: Obtain the output image $\boldsymbol{Y}$ by minimizing the energy function $E(\boldsymbol{Y}|\boldsymbol{X})$.



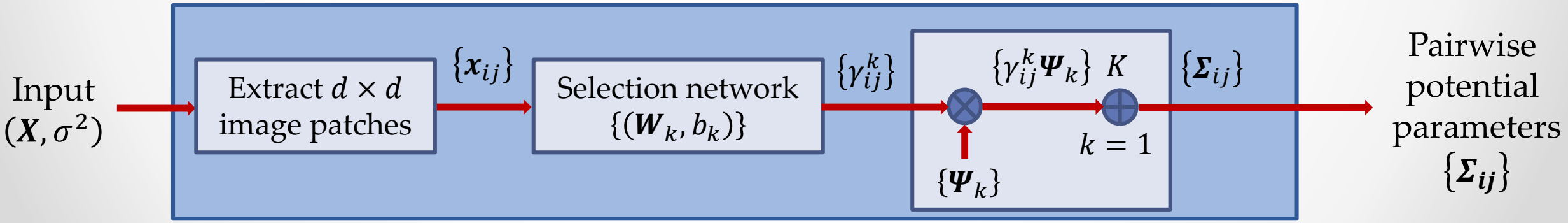
Parameter Generation Network (PgNet)

- We model the pairwise potential parameters $\boldsymbol{\Sigma}_{ij}$ as convex combinations of K symmetric positive semidefinite matrices $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \ldots, \boldsymbol{\Psi}_K$:

$$\boldsymbol{\Sigma_{ij}} = \sum_{k=1}^{K} \gamma_{ij}^{k}\, \boldsymbol{\Psi}_k\,, \qquad \gamma_{ij}^{k} \geq 0, \qquad \sum_{k=1}^{K} \gamma_{ij}^{k} = 1.$$

- We compute the combination weights γ_{ij}^{k} from the input image patches $\boldsymbol{x}_{ij}$ using:

$$\gamma_{ij}^{k} = \frac{e^{s_{ij}^{k}}}{\sum_{p=1}^{K} e^{s_{ij}^{p}}}, \quad s_{ij}^{k} = -\frac{1}{2}\boldsymbol{x}_{ij}^{T}\boldsymbol{G}^{T}(\boldsymbol{W}_k + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{G}\boldsymbol{x}_{ij} + b_k;\ \boldsymbol{W}_k \succcurlyeq 0.$$



Gaussian CRF Inference

- Given the noisy input image $\boldsymbol{X}$ and the pairwise potential function parameters $\boldsymbol{\Sigma}_{ij}$, GCRF inference solves the following optimization problem:

$$\boldsymbol{Y}^* = \underset{\boldsymbol{Y}}{\operatorname{argmax}}\ P(\boldsymbol{Y}|\boldsymbol{X}) = \underset{\boldsymbol{Y}}{\operatorname{argmin}}\ \frac{1}{2\sigma^2}\sum_{ij}\big(\boldsymbol{Y}(i,j) - \boldsymbol{X}(i,j)\big)^2 + \frac{1}{2d^2}\sum_{ij}\boldsymbol{y}_{ij}^T\boldsymbol{G}^T\big(\boldsymbol{\Sigma}_{ij}\big)^{-1}\boldsymbol{G}\boldsymbol{y}_{ij}\,.$$

  - Unconstrained quadratic program and hence be solved in closed form.
  - Closed form solution requires solving a linear system with number of variables equal to the number of pixels.

- Instead of exactly solving the full linear system, we perform approximate inference using the iterative inference procedure proposed in [Zoran 2011].

D. Zoran and Y. Weiss, "From Learning Models of Natural Image Patches to Whole Image Restoration", ICCV, 2011.

GCRF Inference [Zoran 2011]

- Modify the original cost function by introducing auxiliary variables $\boldsymbol{z}_{ij}$:

$$\frac{1}{\sigma^2}\sum_{ij}[\boldsymbol{Y}(i,j)-\boldsymbol{X}(i,j)]^2+\sum_{ij}\boldsymbol{y}_{ij}^T\boldsymbol{G}^T\left(\boldsymbol{\Sigma}_{ij}\right)^{-1}\boldsymbol{G}\boldsymbol{y}_{ij}$$

$$J\left(\boldsymbol{Y},\{\boldsymbol{z}_{ij}\},\beta\right)=\frac{1}{\sigma^2}\sum_{ij}[\boldsymbol{Y}(i,j)-\boldsymbol{X}(i,j)]^2+\beta\left\|\boldsymbol{y}_{ij}-\boldsymbol{z}_{ij}\right\|_2^2+\sum_{ij}\boldsymbol{z}_{ij}^T\boldsymbol{G}^T\left(\boldsymbol{\Sigma}_{ij}\right)^{-1}\boldsymbol{G}\boldsymbol{z}_{ij}\,.$$

D. Zoran and Y. Weiss, “From Learning Models of Natural Image Patches to Whole Image Restoration”, ICCV, 2011.

GCRF Inference [Zoran 2011]

- Initialize $\boldsymbol{Y} = \boldsymbol{X}$ and $\beta = \frac{1}{\sigma^2}$.
- For t = 1 to T
  - Minimize J with respect to $\{\boldsymbol{z}_{ij}\}$ fixing $\boldsymbol{Y}$:

$$\underset{\boldsymbol{z}_{ij}}{\operatorname{argmin}}\ \boldsymbol{z}_{ij}^T \boldsymbol{G}^T \left(\boldsymbol{\Sigma}_{ij}\right)^{-1} \boldsymbol{G}\boldsymbol{z}_{ij} + \beta \left\|\boldsymbol{y}_{ij} - \boldsymbol{z}_{ij}\right\|_2^2 = \left(\boldsymbol{I} - \boldsymbol{G}^T \left(\beta \boldsymbol{\Sigma}_{ij} + \boldsymbol{G}\boldsymbol{G}^T\right)^{-1} \boldsymbol{G}\right) \boldsymbol{y}_{ij}$$

  - Minimize J with respect to $\boldsymbol{Y}$ fixing $\{\boldsymbol{z}_{ij}\}$:

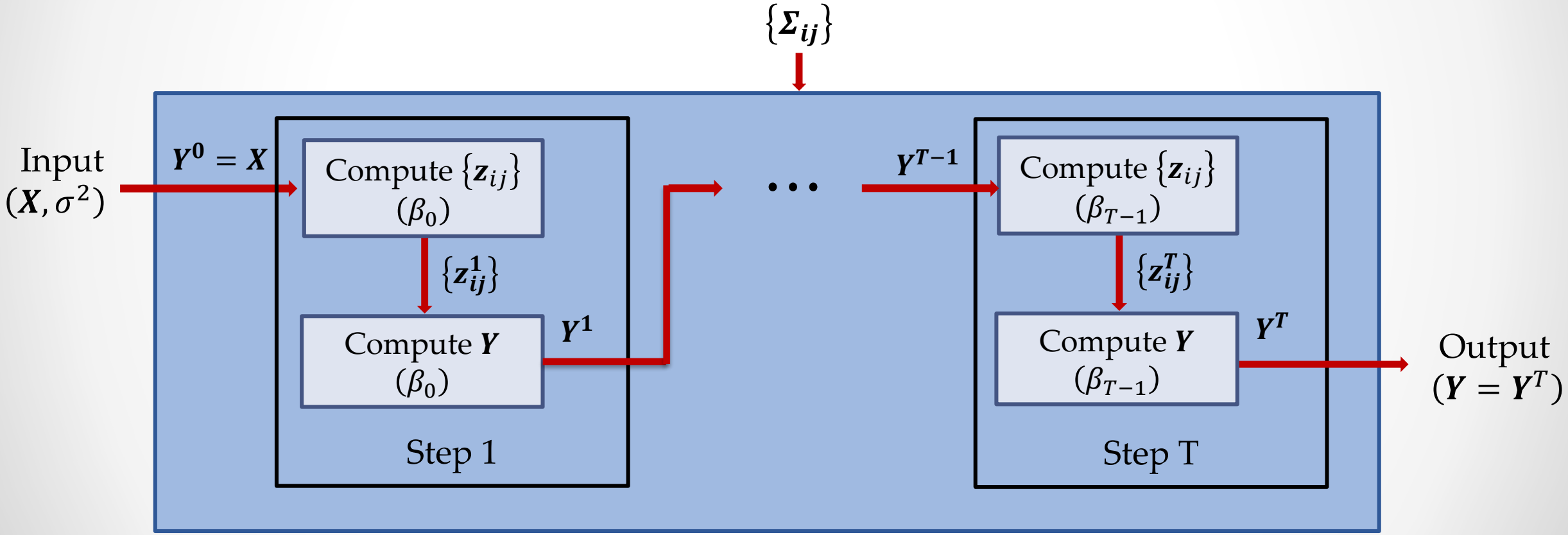
$$\underset{\boldsymbol{Y}(i,j)}{\operatorname{argmin}}\ \frac{1}{\sigma^2} \sum_{ij} [\boldsymbol{Y}(i,j) - \boldsymbol{X}(i,j)]^2 + \beta \sum_{p,q=-\left\lfloor \frac{d-1}{2} \right\rfloor}^{\left\lceil \frac{d-1}{2} \right\rceil} \left[\boldsymbol{Y}(i,j) - \boldsymbol{z}_{pq}(i,j)\right]^2 = \frac{\boldsymbol{X}(i,j)}{1 + \beta\sigma^2 d^2} + \frac{\beta\sigma^2}{1 + \beta\sigma^2 d^2} \sum_{p,q=-\left\lfloor \frac{d-1}{2} \right\rfloor}^{\left\lceil \frac{d-1}{2} \right\rceil} \boldsymbol{z}_{pq}(i,j)$$

  - Increase β to β_t.
- This iterative procedure has been shown to work very well for image restoration tasks even with few (5-6) iterations with a beta schedule given by $\beta_t = \frac{2^{t+1}}{\sigma^2}$.

D. Zoran and Y. Weiss, "From Learning Models of Natural Image Patches to Whole Image Restoration", ICCV, 2011.

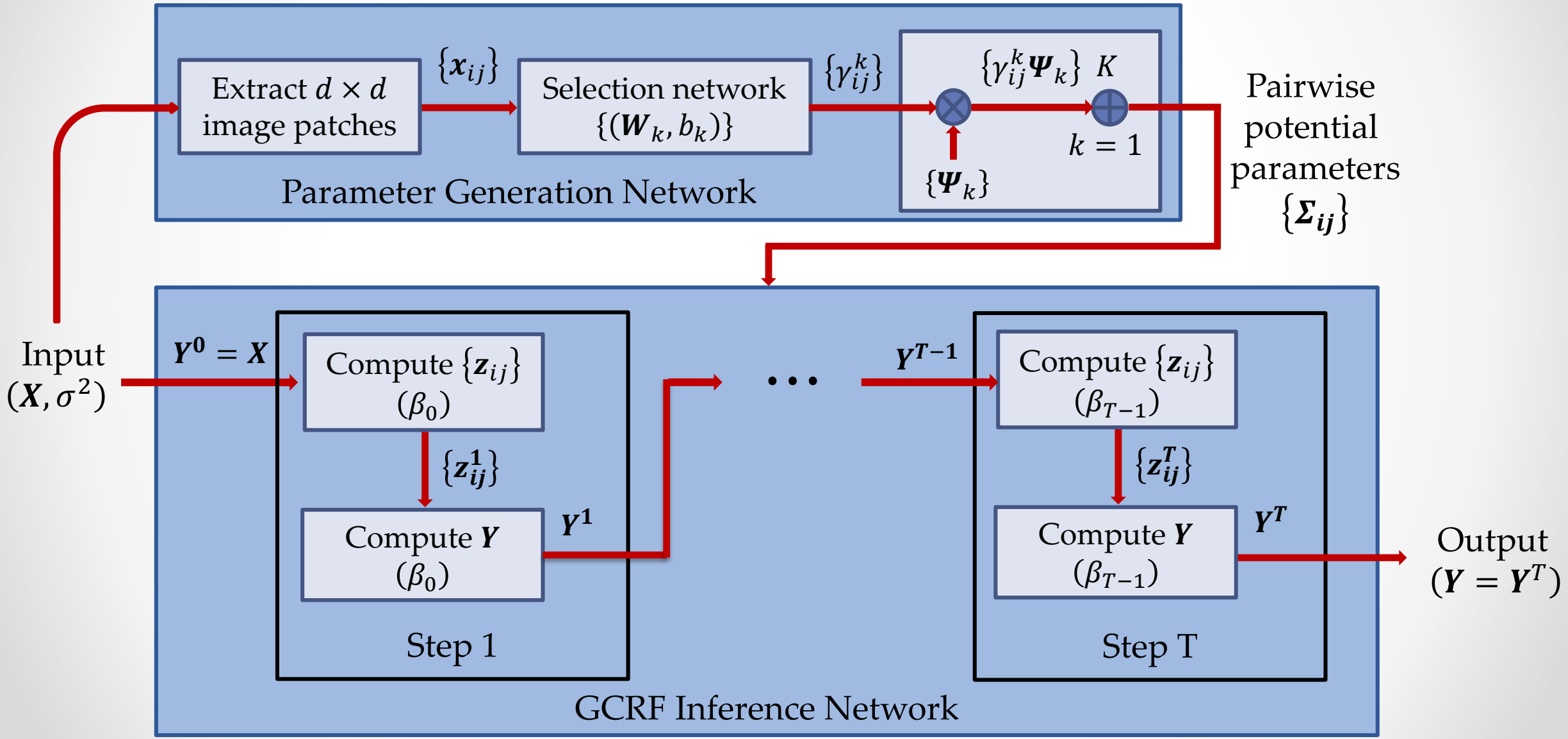
GCRF Inference Network

- We unroll the iterative inference procedure used in [Zoran 2011] into a deep network.

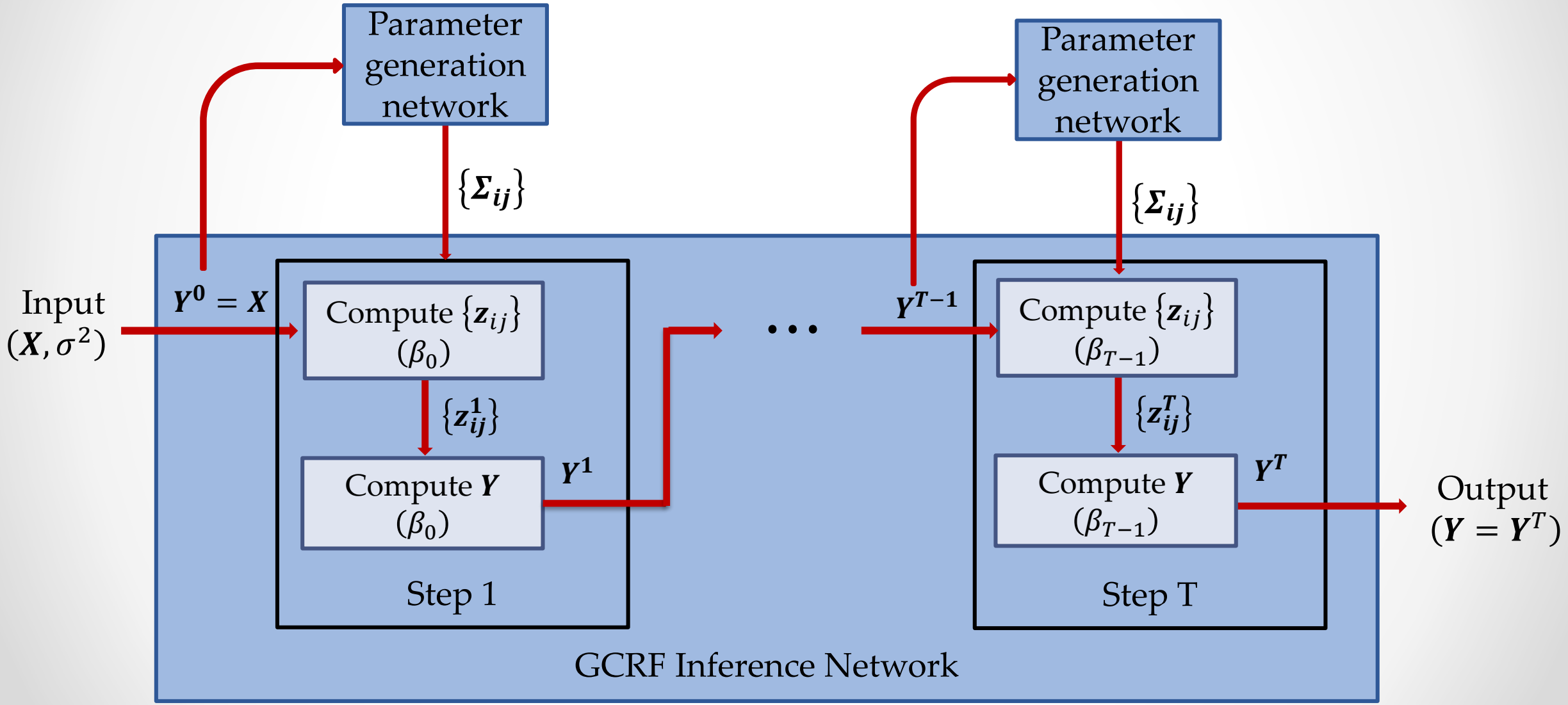


D. Zoran and Y. Weiss, "From Learning Models of Natural Image Patches to Whole Image Restoration", ICCV, 2011.

GCRF Network



Final GCRF Network



Training

- Trained the entire deep GCRF network (all PgNets and InfNet) end-to-end discriminatively by maximizing the average PSNR measure on a training set.
- Used standard back-propagation to compute the gradient of the network parameters.
- We have a constrained optimization because of the symmetry and positive semi-definiteness constraints on the parameters $\boldsymbol{\Psi}_k$ and $\boldsymbol{W}_k$.
- Parametrized $\boldsymbol{\Psi}_k$ and $\boldsymbol{W}_k$ as $\boldsymbol{\Psi}_k = \boldsymbol{R}_k \boldsymbol{R}_k^T$ and $\boldsymbol{W}_k = \boldsymbol{P}_k \boldsymbol{P}_k^T$, where $\boldsymbol{R}_k$ and $\boldsymbol{P}_k$ are lower triangular matrices.
- Used L-BFGS for training.

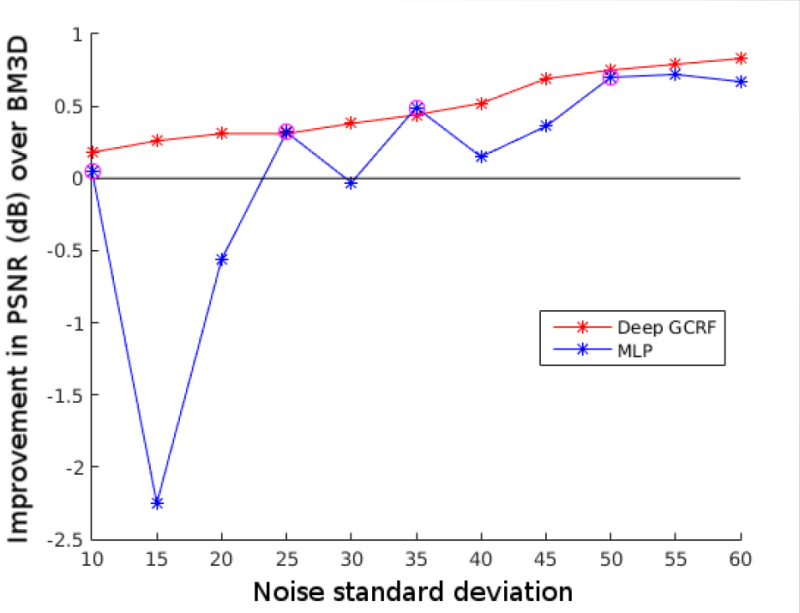
Experimental Results

- Dataset: 400 training images and 300 test images
- Used six inference steps.
- Performed experiments with 5×5 and 8×8 image patches, and Gaussian noise.
- We trained two deep GCRF networks: Low noise network for $\sigma \leq 25$, High noise network for $\sigma \geq 25$

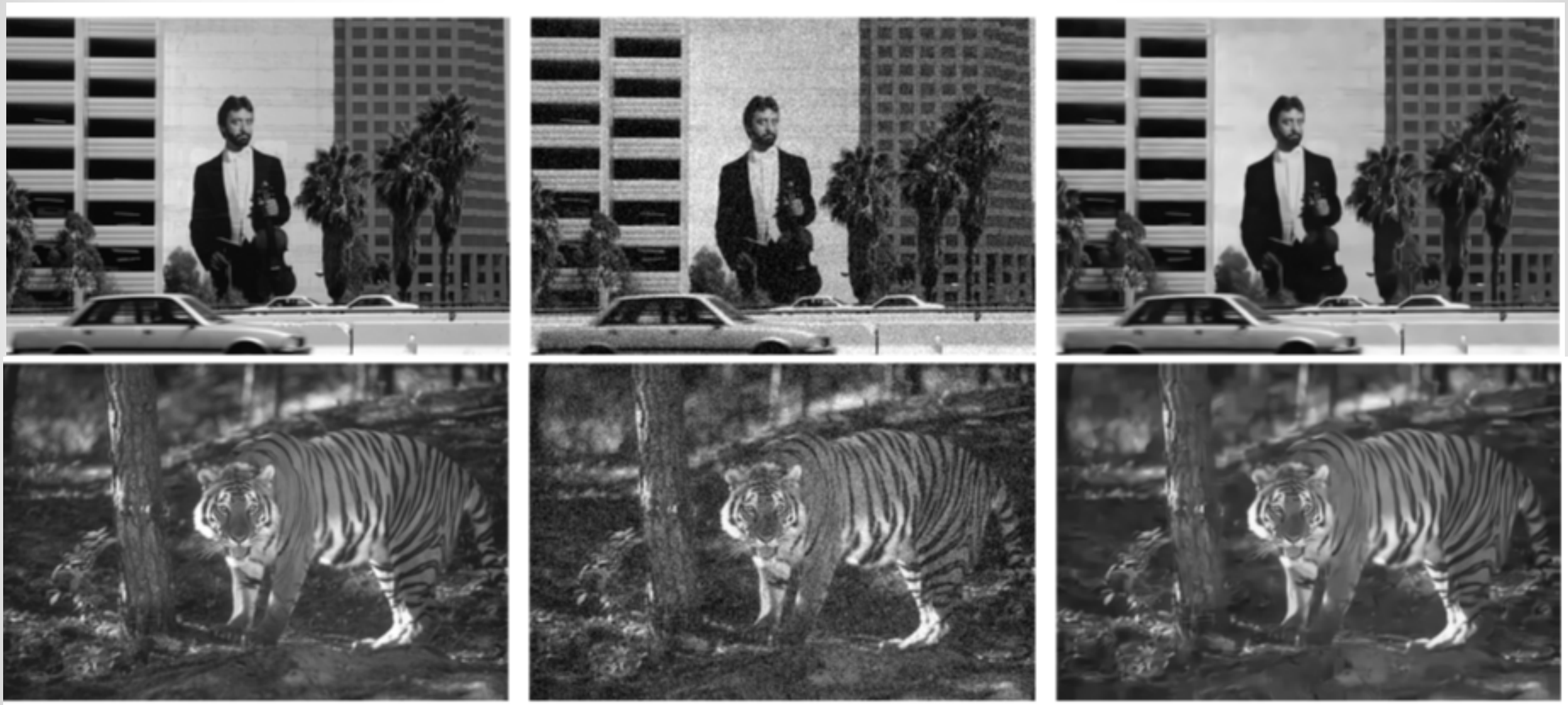
Average PSNR values

Test σ	10	15	20	25	30	35	40	45	50	55	60
ClusteringSR [7]	33.27	30.97	29.41	28.22	27.25	26.30	25.56	24.89	24.28	23.72	23.21
EPLL [40]	33.32	31.06	29.52	28.34	27.36	26.52	25.76	25.08	24.44	23.84	23.27
BM3D [6]	33.38	31.09	29.53	28.36	27.42	26.64	25.92	25.19	24.63	24.11	23.62
NL-Bayes [20]	33.46	31.11	29.63	28.41	27.42	26.57	25.76	25.05	24.39	23.77	23.18
NCSR [8]	33.45	31.20	29.56	28.39	27.45	26.32	25.59	24.94	24.35	23.85	23.38
WNNM [14]	**33.57**	31.28	29.70	28.50	27.51	26.67	25.92	25.22	24.60	24.01	23.45
CSF [31]	-	-	-	28.43	-	-	-	-	-	-	-
MLP [3]	33.43	-	-	**28.68**	-	**27.13**	-	-	**25.33**	-	-
$DGCRF_5$	33.53	**31.29**	**29.76**	28.58	**27.68**	26.95	**26.30**	**25.73**	25.23	**24.76**	**24.33**
$DGCRF_8$	**33.56**	**31.35**	**29.84**	**28.67**	**27.80**	**27.08**	**26.44**	**25.88**	**25.38**	**24.90**	**24.45**

Table 1: Comparison of various denoising approaches on 300 test images.



Results (standard deviation 25)



Original Noisy Denoised

Summary

- We proposed a novel end-to-end trainable deep network for image denoising based on a Gaussian CRF model.
- The proposed network consists of two sub-networks: a parameter generation network and an inference network.
- The proposed deep network explicitly models the input noise variance and is capable of handling a range of noise levels.
- We achieved results on par with the state-of-the-art without training a separate network for each individual noise level.

Classification of Manifold Representations

- Manifold representations like linear subspaces and symmetric positive definite matrices are used in various computer vision application such as face recognition, object recognition, texture segmentation, action recognition, etc.
- Due to the non-Euclidean nature of the underlying space, these representations are often classified using kernel-based approaches such as kernel-LDA, kernel-PLS, kernel SVM, etc.
- For kernel based approaches, the choice of kernel is crucial for achieving good performance.
- Addressed the problem of kernel selection for the classification of manifold representations using the multiple kernel learning approach.
- Proposed two criteria for learning a suitable kernel from data and showed that the learning problem can be formulated as a convex optimization problem for the SVM classifier.

Results: Image Set-based Recognition

- Youtube: Face dataset
- ETH80: Object dataset

Dataset	NN	Standard MKL	GDA [Hamm 2008]	Proj + PLS [Wang 2012]	Proposed Approach
YouTube	62.8	64.3	65.7	67.7	**70.8**
ETH80	93.2	93.7	92.8	95.3	**96.0**

Recognition rates using linear subspaces.

Dataset	NN	Standard MKL	CDL-LDA [Wang 2012]	CDL-PLS [Wang 2012]	Proposed Approach
YouTube	40.7	69.7	67.5	70.1	**73.2**
ETH80	92.7	93.7	94.5	96.5	**98.2**

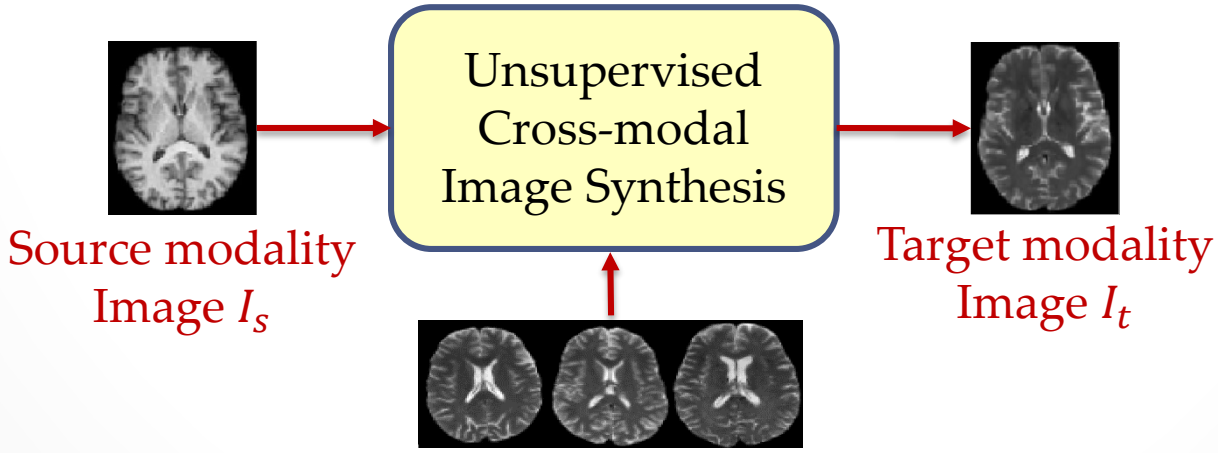
Recognition rates using covariance matrices.

J. Hamm and D. D. Lee, "Grassmann Discriminant Analysis: a Unifying View on Subspace-Based Learning," In ICML, 2008.

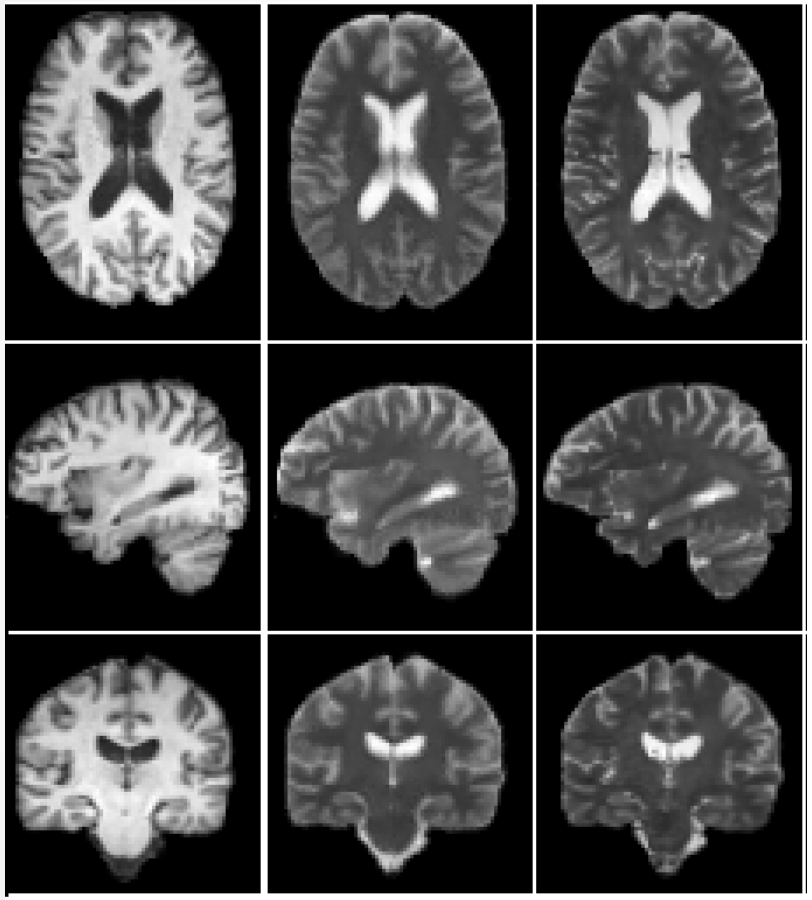
R. Wang, H. Guo, L. S. Davis and Q. Dai, "Covariance Discriminative Learning: A Natural and Efficient Approach to Image Set Classification", In CVPR, 2012.

Unsupervised Cross-Modal Image Synthesis

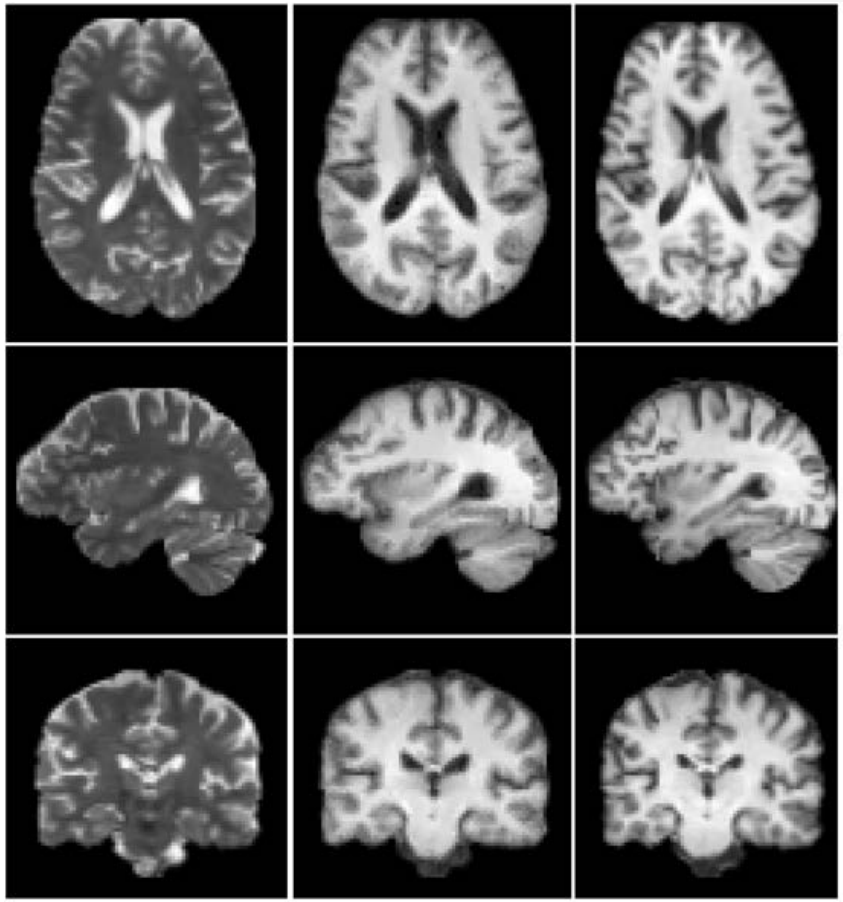
- Collecting medical images is time consuming, expensive, and exposes patients to radiation.
- Image synthesis is a potential solution to this problem.
- Formulated image synthesis as an optimization problem that maximizes the mutual information between the input source modality image and the synthesized target modality image.



Results



T1 Input | Proposed | Ground truth T2



T2 Input | Proposed | Ground truth T1

Conclusion

- In this thesis we focused on both the *representation* and *context modeling* aspects of computer vision.
- We introduced relative 3D geometry based skeletal representations for human action recognition.
- We showed how rolling can be used to reduce the distortions while mapping temporal sequences from the special orthogonal group to its Lie algebra, and how this can be used for skeleton-based human action recognition.
- We proposed new deep network architectures based on Gaussian conditional random field models for semantic image segmentation and image denoising.

Future work

- We used the relative 3D geometry between all pairs of body parts in our skeletal representation. Using feature selection approaches such as multiple kernel learning to automatically identify the set of body parts that differentiates a given action from the rest could further improve the action recognition performance.

- The idea of rolling can be used for classification of temporal sequences on other manifolds such as Grassmann manifold and the manifold of symmetric positive definite matrices.

- The Gaussian CRF-based deep network architecture can be extended to other applications such as image deblurring, super resolution, depth estimation, etc.

- We can also explore other GCRF inference procedures such as Gaussian belief propagation to design new GCRF inference networks.

Publications and Awards

- [Journal] **R. Vemulapalli**, F. Arrate, and R. Chellappa, "R3DG Features: Relative 3D Geometry-based Skeletal Representations for Human Action Recognition", Accepted to **Computer Vision and Image Understanding**.

- **R. Vemulapalli**, O. Tuzel, M.-Y. Liu, and R. Chellappa, "Gaussian Conditional Random Field Network for Semantic Segmentation", **CVPR (Spotlight)**, 2016.
- **R. Vemulapalli**, O. Tuzel, and M.-Y. Liu, "Deep Gaussian Conditional Random Field Network: A Model-based Deep Network for Discriminative Denoising", **CVPR**, 2016.
- **R. Vemulapalli** and R. Chellappa, "Rolling Rotations for Recognizing Human Actions from 3D Skeletal Data", **CVPR**, 2016.
- **R. Vemulapalli**, H. V. Nguyen, and S. K. Zhou, "Unsupervised Cross-modal Synthesis of Subject-specific Scans", **ICCV**, 2015.
- H. V. Nguyen, S. K. Zhou, and **R. Vemulapalli**, "Cross-Domain Synthesis of Medical Images Using Efficient Location-Sensitive Deep Network", **MICCAI**, 2015.
- **R. Vemulapalli**, F. Arrate, and R. Chellappa, "Human Action Recognition by Representing 3D Human Skeletons as Points in a Lie Group", **CVPR (Oral)**, 2014.
- **R. Vemulapalli**, J. Pillai, and R. Chellappa, "Kernel Learning for Extrinsic Classification of Manifold Features", **CVPR**, 2013.

Book chapter:

- **R. Vemulapalli**, H. V. Nguyen, and S. K. Zhou, "Deep Networks and Mutual Information Maximization for Cross-modal Medical Image Synthesis", **Elsevier's book on Deep Learning for Medical Image Analysis**, To be published.

Awards:

- University of Maryland A. James Clark School of Engineering Dean's Doctoral Research Award, 2016 (second prize).
- University of Maryland ECE Distinguished Dissertation Fellowship, 2016.

Thank You

. . .