ECCV 2018

Deep Kalman Filtering Network for Video Compression Artifact Reduction

Guo Lu[1], Wanli Ouyang[2,3], Dong Xu[2], Xiaoyun Zhang[1], Zhiyong Gao[1], and Ming-Ting Sun[4]

[1] Shanghai Jiao Tong University, China
{luguo2014,xiaoyun.zhang, zhiyong.gao}@sjtu.edu.cn
[2] The University of Sydney, Australia
{wanli.ouyang,dong.xu}@sydney.edu.au
[3] The University of Sydney, SenseTime Computer Vision Research Group
[4] University of Washington, USA
mts@uw.edu

Introduced by Qing Lyu on 7/31/2019

Contribution

- Proposed a deep Kalman filtering network (DKFN) to implement video compression artifact reduction
- Bridged the gap between the model-based methods and learning-based methods by Kalman model and deep neural network
- Achieved high accuracy compared with SOTA methods

3D image restoration

- 2D CNN: U-Net
- 3D CNN: V-Net, 2D-3D ConvNet
- Tri-planar CNN
- RNN

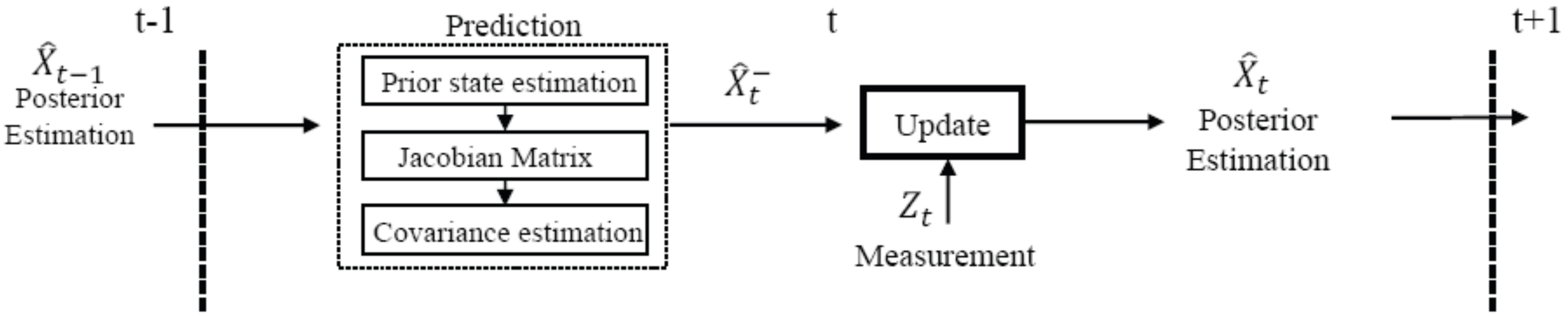
Kalman Filter

- Estimate a certain state according to this state’s prediction and measurement results.

- Example: Car position estimation
  - Known
    - previous state position
    - moving direction and speed
    - time

      Prediction
    - current state measurement
  - Goal: find current car position

Kalman Filter



(a) Basic Kalman model

Predict

Control-input model

$$\hat{X}_t^- = F_t \hat{X}_{t-1} + B_t u_t$$ Control vector

State-transition model

Posterior error covariance matrix

$$P_t^- = F_t P_{t-1} F_t^T + Q_t$$ Covariance of process noise

Prior error covariance matrix

Update

Measurement profit residual

$$\hat{Y}_t = Z_t - H_t \hat{X}_t^-$$

Observation model

Profit residual covariance

$$S_t = H_t P_t^- H_t^T + R_t$$ Covariance of observation noise

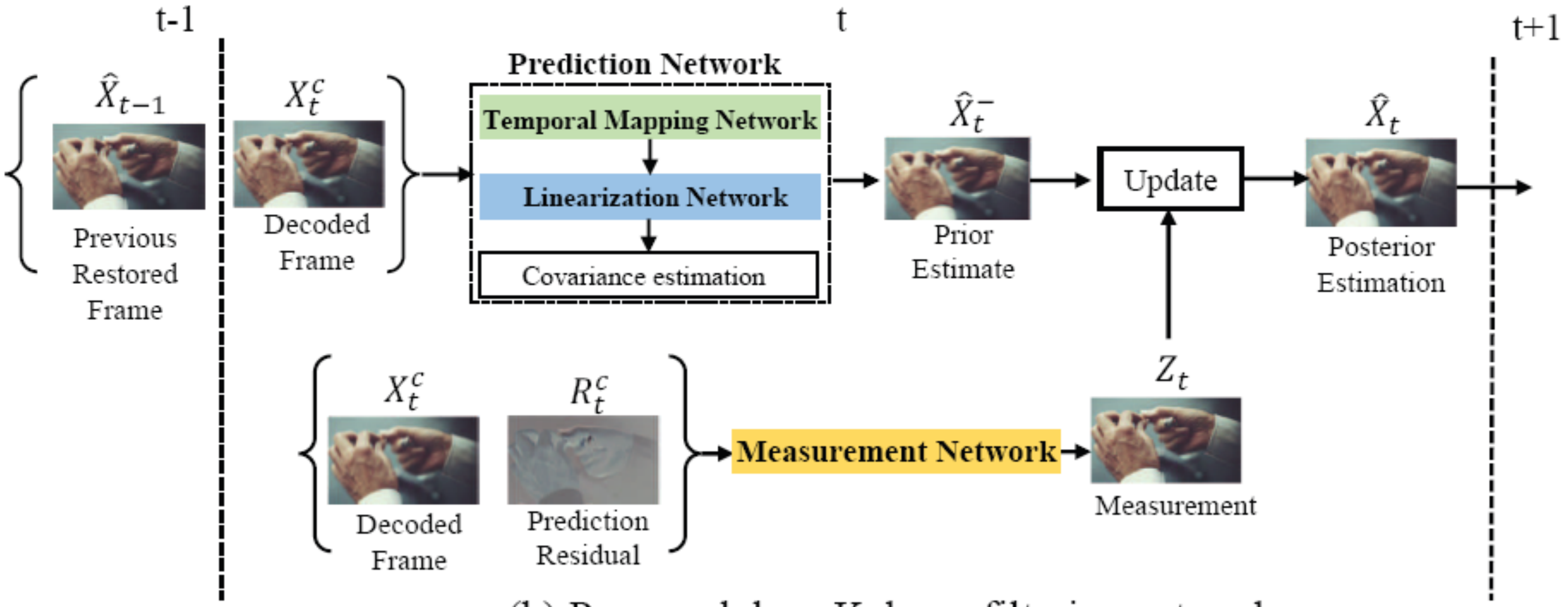
Optimal Kalman gain

$$K_t = P_t^- H_t^T S_t^{-1}$$

$$\hat{X}_t = \hat{X}_t^- + K_t \hat{Y}_t$$

$$P_t = (I - K_t H_t) P_t^-$$

DKFN



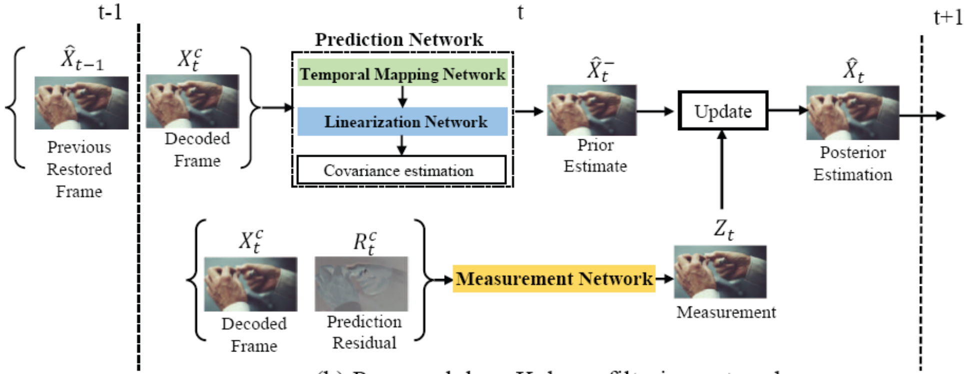
(b) Proposed deep Kalman filtering network

DKFN

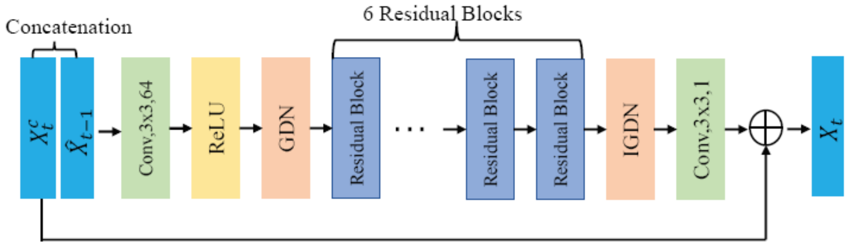
- Temporal Mapping Network
  - Generate $\hat{X}_t^-$
    $$\mathcal{L}_f(\theta_f) = ||X_t - \mathcal{F}(\hat{X}_{t-1}, X_t^c; \theta_f)||_2^2$$
- Linearization Network
  - Learn F_t
    $$\mathcal{L}_m(\theta_m) = ||\hat{X}_t^- - \mathcal{G}(\hat{X}_{t-1}, X_t^c; \theta_m)\hat{X}_{t-1}||_2^2$$
- Measurement Network
  - Generate Z_t
    $$\mathcal{L}_z(\theta_z) = ||X_t - (\mathcal{M}(X_t^c, R_t^c; \theta_z) + X_t^p)||_2^2$$

Fine-tune

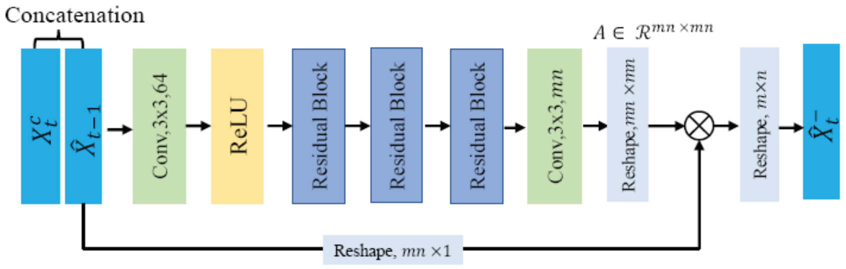
$$\mathcal{L}(\theta) = ||X_t - \hat{X}_t||_2^2$$



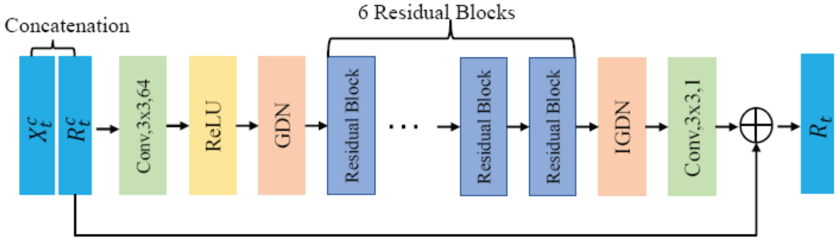
(b) Proposed deep Kalman filtering network



(a) Temporal Mapping Network.



(b) Linearization Network.



(c) Measurement network.

Experiment

- Dataset
  - Vimeo-90K
    - 4278 videos with 89800 independent clips
    - 448×256
    - 64612 clips for training
    - 7824 clips for evaluation
- Training
  - First train the temporal mapping network for 40 epochs
  - Then train the linearization network
  - Next train the measurement network for 40 epochs
  - Fine-tune the whole model

Result

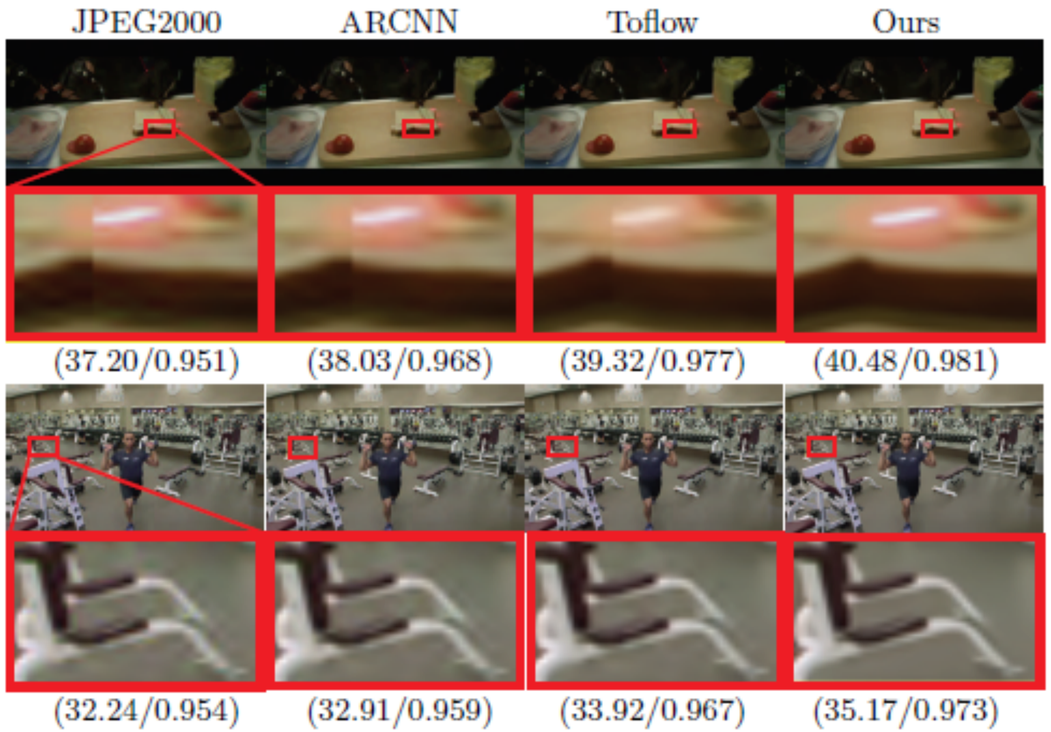


Fig. 4. Quantitative (PSNR/SSIM) and visual comparison of JPEG2000 artifact reduction on the Vimeo dataset for q=20.

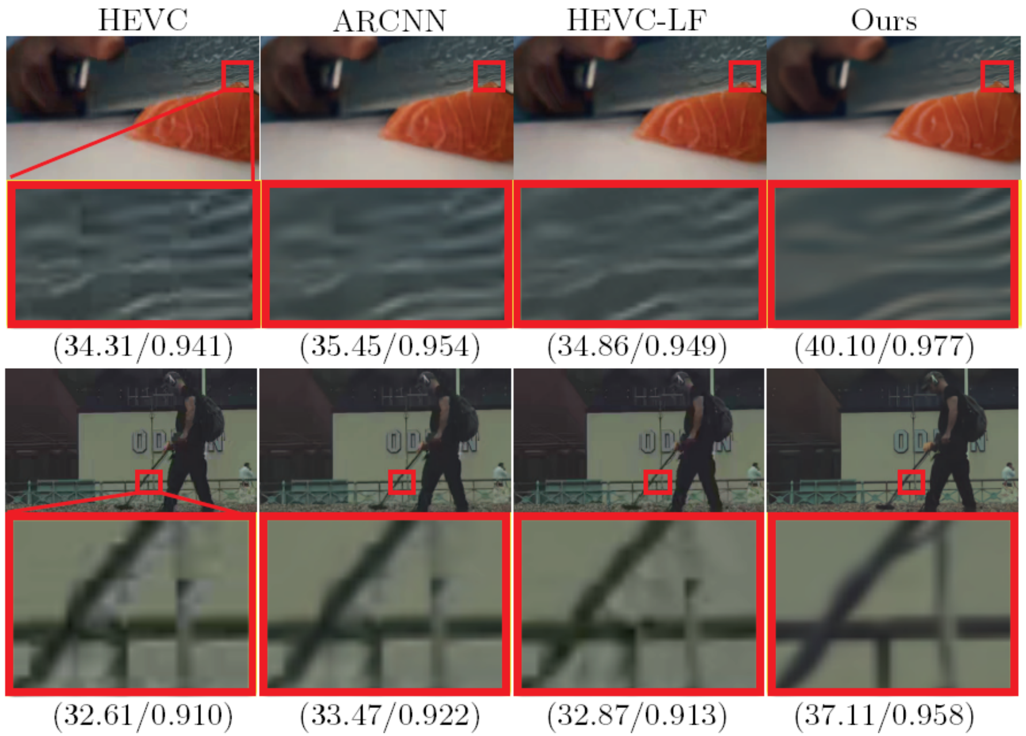


Fig. 5. Quantitative (PSNR/SSIM) and visual comparison of different methods for HEVC artifact reduction on the Vimeo dataset at qp=37.

Dataset	Setting	ARCNN [18]	DnCNN [17]	V-BM4D [47]	Toflow [21]	Ours
Vimeo	q=20	36.11/0.960	37.26/0.967	35.75/0.959	36.92/0.966	**37.93/0.971**
	q=40	34.21/0.944	35.22/0.953	33.99/0.940	34.97/0.953	**35.88/0.958**

Table 1. Average PSNR/SSIM results on the Vimeo dataset for JPEG2000 artifact reduction (q=20,40).

Dataset	Setting	ARCNN [18]	DnCNN [17]	HEVC-LF [1]	Ours
Vimeo	qp=32	34.87/0.954	35.58/0.961	34.19/0.950	**35.81/0.962**
	qp=37	32.54/0.930	33.01/0.936	31.98/0.923	**33.23/0.939**

Table 2. Average PSNR/SSIM results on the Vimeo test sequences for HEVC artifact reduction (qp=32,37).

Conclusion

- Propose a DKFN model combines Kalman Filter with neural network for video compression artifact reduction
- Take advantage of
  - the recursive nature of Kalman filter
  - Representation learning ability of neural network
- Superiority over SOAT methods