Frame Rate Conversion

Dynamic Scene Analysis Project #1

120200153

이성훈

Basic idea

- **Frame rate conversion**
  - Frame rate conversion이란 frame간 interpolation을 통해 기존 fps(frame per second)와 다른 fps로 만들어줌
  - Figure 1은 frame up conversion을 나타내며 frame 사이에 interpolated frame을 생성
  - 이러한 방법을 통해 fps를 높여 흐름을 자연스럽게 만들거나,
    fps는 유지하며 frame 개수를 늘리는 방식으로 slow motion 효과를 만들 수 있음
  - Block matching algorithm을 통해 진행하기에 복원 성능 및 computing 속도에 한계로 deep learning을 이용

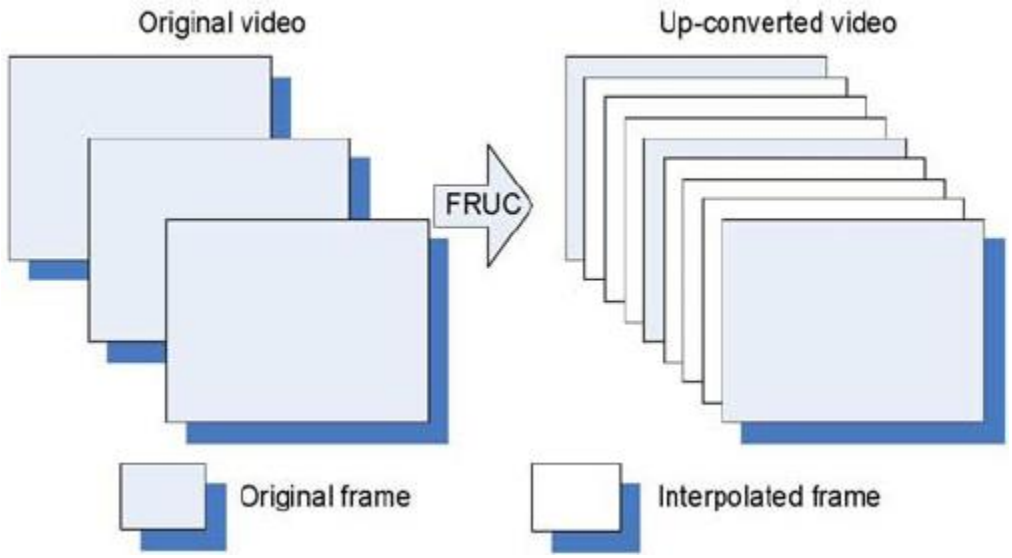

Figure 1. Frame rate up conversion(FRUC)

Basic idea

- **Slow motion with deep learning**
  - Slow motion은 Figure 2와 같이 주요한 장면을 느리게 재생하는 효과를 띄도록 조절하는 기술
  - Slow motion을 하기위해 240 fps정도의 성능을 갖는 높은 가격의 카메라로 촬영해야함
    일반적인 경우 30 fps 혹은 60 fps로 촬영되므로 카메라는 유지하며 software적으로 interpolation하여 보완
  - 잘 알려진 기존 방식인 block matching과 motion compensation algorithm을 사용하여 구현 가능
    하지만 rule based 방식을 사용해 낮은 성능과 느린 computing속도를 보완하기 위해 deep learning을 적용
  - 위 한계점을 극복하기 위해 deep learning을 통한 slow motion 기술인 Super SloMo[1]와 Unsupervised CC[2] 논문을 참고

Figure 2. Example of slow motion(Left : original video, Right : slow motion video)

Super SloMo

- **Super SloMo : High Quality Estimation of Multiple Intermediate Frames for Video Interpolation[1]**
  - 일반적인 slow motion task에서 bi-directional optical flow를 사용하는 task에서 두 가지 문제점이 존재
    Locally smooth된 영역에서만 잘 작동함
    motion boundary에서 인위적인 결과가 나옴
  - 위 문제점들을 해결하기 위해 optical flow를 추정하는 network의 loss를 적절히 정의함으로써 해결

$$\hat{I} = \alpha \odot g(I_0, F_{t\to 0}) + (1-\alpha_0) \odot g(I_1, F_{t\to 1}) where\ F_{t\to 0}, F_{t\to 1} \text{ are optical flow} - eq1$$

$$\hat{I_t} = \frac{1}{Z} \odot \left((1-t)V_{t\leftarrow 0} \odot g(I_0, F_{t\to 0}) + tV_{t\leftarrow 1} \odot g(I_1, F_{t\to 1})\right) where\ V_t\ is\ visibility\ map - \ eq2$$

$$\hat{F_{t\to 0}} = -(1-t)t\,F_{0\to 1} + t^2 F_{1\to 0}, \qquad \hat{F_{t\to 1}} = (1-t)^2\,F_{0\to 1} - t\,(1-t)F_{1\to 0}$$

  - Frame I_t를 추정한다고 가정하면 optical flow를 이용한 interpolation g를 이용한 Equation 1, 2 를 통해 구할 수 있음
  - 기존의 occlusion problem을 해소하기 위해 visibility map 개념을 도입하였으며 $V_{t\leftarrow 0}$는 t frame이 0 frame에 occlude된 정도를 나타냄
  - $Z = (1-\mathrm{t})\mathrm{V}_{\mathrm{t}\to 0} + \mathrm{tV}_{\mathrm{t}\to 1}$으로 식을 normalize 해주는 term

Super SloMo

- **Super SloMo network**
  - Network 전체 구조는 Figure 3 우측과 같으며 두 network 모두 Figure 3 좌측과 같은 U-Net을 기반으로 구성됨

  - U-Net은 segmentation에서 사용하는 network로 skip layer를 통해 기존 spatial 정보를 유지하며 U자 구조를 띔
  - Flow computation network는 optical flow를 추정하는 network
  - Arbitrary-time flow interpolation network는 기 추정된 정보들을 바탕으로 interpolation frame을 추정하는 network

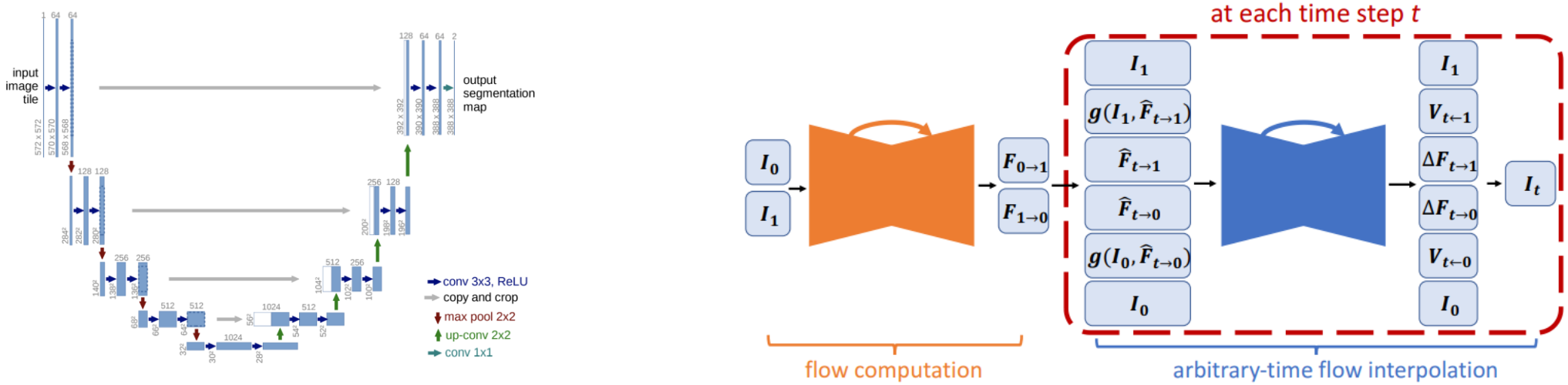

Figure 3. Network overview(left : U-Net, right : Super SloMo network)

Super SloMo

- **Super SloMo loss configuration**

$$l = \lambda_r l_r + \lambda_p l_p + \lambda_w l_w + \lambda_s l_s \ - eq\ 3$$

$$l_r = \frac{1}{N}\Sigma_{i=1}^{N} \left|\left|\hat{I}_{t_i} - I_{t_i}\right|\right|_1 \ - eq\ 4$$

$$l_p = \frac{1}{N}\Sigma_{i=1}^{N} \left|\left|\phi(\hat{I}_t) - \phi(I_t)\right|\right|_2 \ - eq\ 5$$

$$l_w = \left|\left|I_0 - g(T_1, F_{0\to1})\right|\right|_1 + \left|\left|I_1 - g(I_0, F_{1\to0})\right|\right|_1 + \frac{1}{N}\Sigma_{i=1}^{N} \left|\left|I_{t_i} - g(I_0, \hat{F}_{t_i\to0})\right|\right|_1 + \frac{1}{N}\Sigma_{i=1}^{N} \left|\left|I_{t_1} - g(I_1, \hat{F}_{t_i\to1})\right|\right|_1 \ - eq\ 6$$

$$l_s = \left|\left|\nabla F_{0\to1}\right|\right|_1 + \left|\left|\nabla F_{1\to0}\right|\right|_1 \ - eq\ 7$$

  - Total loss는 모든 loss를 합친 Equation 3
  - l_r은 reconstruction loss로 복원한 image와 ground truth가 잘 복원이 되도록 l1 loss를 적용
  - l_p는 perceptual loss로 이미지의 특징을 network ϕ를 통해 나온 특징이 유사하도록 l2 loss를 적용
  - l_w는 warping loss로 optical flow를 이용하여 추정한 frame과 l1 loss를 통해 기존 interpolation 방법과 유사하도록 학습
  - l_s는 smoothing loss로 motion boundary에서 인위적인 결과가 나오는 것을 smoothing을 통해 방지함

Unsupervised Cycle Consistency

- **Unsupervised Video Interpolation Using Cycle Consistency[2]**
  - Super SloMo에선 supervised 방법을 통해 training하여 slow motion을 구현 하지만,
    일반적인 경우 240 fps가 아닌 30 fps 또는 60 fps dataset만 있는 상황에서 slow motion을 training해야 하는 경우가 존재
  - 다른 영상으로 학습된 pre-trained model을 이용하여 fine-tuning을 통해 unsupervised learning을 가능케 함
  - Pre-trained model은 Super SloMo에서 학습된 model을 사용하지만 다른 종류의 model을 사용해도 지장 없음
  - Unsupervised learning방식의 학습에 Super SloMo[1] 논문이 motivation이 됨

  - Figure 4는 training data와 test data가 다르다면 video interpolation에 실패할 수 있어 fine tuning을 통해 low fps에서 잘 동작하도록 함

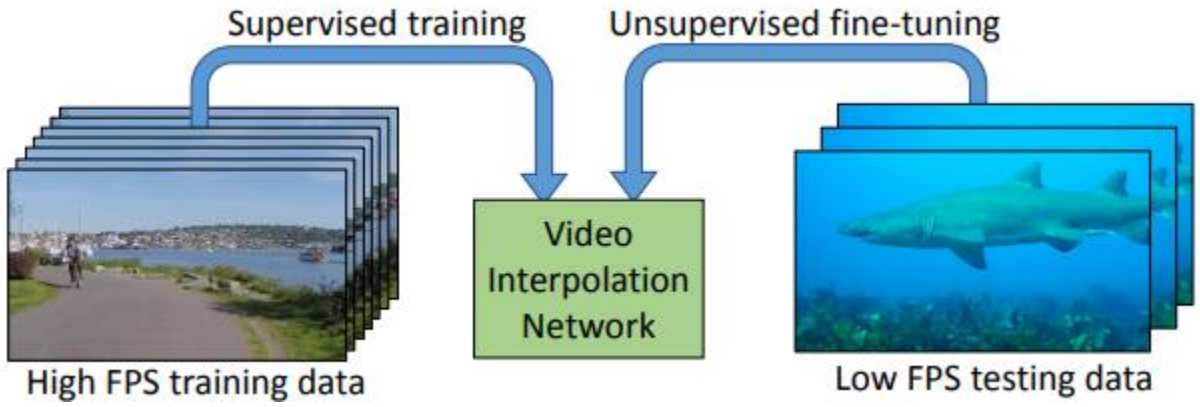

Figure 4. Fine tuning flow

Unsupervised Cycle Consistency

- **Cycle consistency idea**
  - Figure 4는 pre-trained model을 사용한 fine tuning 방식을 나타냄
  - Frame t가 0, 1, 2로 존재할 때, supervised의 경우 0과 1 사이의 frame I_t, 1과 2 사이의 frame I_{t+1}의 image가 존재
  - 하지만 unsupervised이므로 model $\mathcal{M}$로 생성한 $\hat{I}_t$와 $\hat{I}_{t+1}$을 pre-trained model $\mathcal{M}_{pre}$을 통해 비교
  - 또한 알고있는 frame I_1을 활용하기 위해서 $\hat{I}_t$와 $\hat{I}_{t+1}$을 통해 I_1을 생성하고 ground truth와 비교

  - 이러한 방법은 추정한 frame을 바탕으로 다시 frame을 추정하기 때문에 cycle consistency라고 함
  - 위 basic idea와 Super SloMo에서 motivation이 된 idea를 결합하여 loss를 구성함으로써 최종 fine tuning 방법을 제시

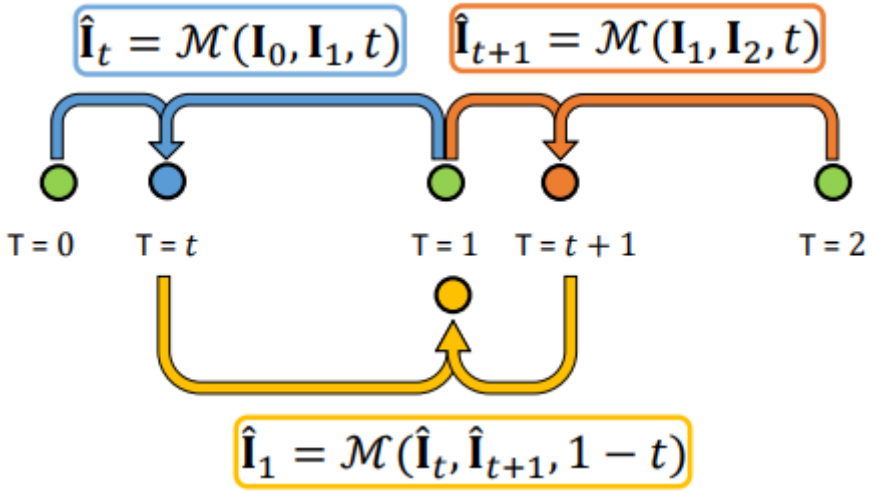

Figure 4. Fine tuning idea overview

Unsupervised Cycle Consistency

- **Loss configuration**

$$L = \lambda_{rc}L_{rc} + \lambda_{rp}L_{rp} + \lambda_{p}L_{p} + \lambda_{w}L_{w} + \lambda_{s}L_{s} - eq\ 8$$

$$L_{rc} = \left|\left|\hat{I}_1 - I_1\right|\right|_1 - eq\ 9$$

$$L_{rp} = \left|\left|\hat{I}_{t_i} - \mathcal{M}_{pre}(I_0, I_1, t_i)\right|\right|_1 + \left|\left|\hat{I}_{t_i+1} - \mathcal{M}_{pre}(I_1, I_2, t_i)\right|\right|_1 - eq\ 10$$

$$L_{p} = \left|\left|\Psi(\hat{I}_1) - \Psi(I_1)\right|\right|_2 - eq\ 11$$

$$L_{w} = \left|\left|\mathcal{T}(I_0, F_{1\to0}) - I_1\right|\right|_1 + \left|\left|\mathcal{T}(I_1, F_{0\to1}) - I_0\right|\right|_1 + \left|\left|\mathcal{T}(I_1, F_{2\to1}) - I_2\right|\right|_1 + \left|\left|\mathcal{T}(I_2, F_{1\to2}) - I_1\right|\right|_1$$
$$+ \left|\left|\mathcal{T}(\hat{I}_t, F_{t+1\to t}) - \hat{I}_{t+1}\right|\right|_1 + \left|\left|\mathcal{T}(\hat{I}_{t+1}, F_{t\to t+1}) - \hat{I}_t\right|\right|_1 - eq\ 12$$

$$L_{s} = \left|\left|\nabla F_{t\to t+1}\right|\right|_1 + \left|\left|\nabla F_{t+1\to t}\right|\right|_1 + \left|\left|\nabla F_{0\to1}\right|\right|_1 + \left|\left|\nabla F_{1\to0}\right|\right|_1 + \left|\left|\nabla F_{1\to2}\right|\right|_1 + \left|\left|\nabla F_{2\to1}\right|\right|_1 - eq\ 13$$

  - Equation 8은 total loss를 나타내며 전체적으로 Super SloMo에서의 loss와 유사한 형태를 가짐
  - L_{rc}는 reconstruction loss, L_p는 perceptual loss, L_s는 smoothing loss로 Super SloMo loss와 유사함
  - L_{rp}는 cycle reconstruction frame을 동작하게하는 loss로 ground truth가 아닌 pre-trained model을 사용하여 추정된 frame과 비교
  - L_w는 warping loss로 Super SloMo case와 유사하나 기존 frame 0, 1로만 추정한 것을 frame 0, 1, 2로 추정한 것
    이때, $\mathcal{T}$는 Super SloMo에서의 g와 유사함

Evaluation Method

- **PSNR(Peak Signal-to-noise ratio)**
  - 영상 또는 동영상 손실 압축에서 화질의 손실 정보를 평가할 때 사용하는 method
  - PSNR의 의미를 직역하면 최대 신호중 잡음의 비율로 해석 가능
  - $PSNR = 20 * \log_{10}(MAX_I) - 10 * \log_{10}(MSE)$

- **SSIM(Structural SIMilarity)**
  - PSNR과 유사하게 화질의 구조적 유사성을 평가할 때 사용하는 method
  - Structure, Luminance, Contrast에 대해 평가하며 각 식을 곱하여 사용함

$$SSIM = l(x,y) * c(x,y) * s(x,y)$$

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$$

Experiment environment

- **Experiment environment setting**
  - 총 3가지 metric에 대해서 비교 - Super SloMo[1], Unsupervised CC[2], 기존 구현한 block matching
  - Dataset은 Adobe240fps중 경향 약간 다른 동영상 2가지를 추출하여 evaluation 진행
  - Block matching의 경우 image를 reconstruction하는데 성능이 좋지 않고 시간이 오래걸려 image로만 성능 비교 진행

  - Block matching의 경우 성능 보존을 위해 full search block size 8, search range 4로 진행
  - Super SloMo의 경우 Adobe240fps를 target으로 만들어진 model을 사용
  - Unsupervised CC의 경우 Adobe240fps을 pre-trained model로 다른 dataset에 대해 fine tuning된 model을 사용

Metric Comparison

- **Validation dataset**
  - Adobe240fps dataset 중 2개 동영상

valid1

valid2

Metric Comparison

- **Video comparison**
  - Valid 1

Super SloMo

PSNR 30.29

Unsupervised CC

PSNR 28.86

Metric Comparison

- **Video comparison**
  - Valid 2

Super SloMo

PSNR 31.45

Unsupervised CC

PSNR 30.38

Metric Comparison

- **Image comparison**
  - Valid 1

BMA

PSNR 29.68
SSIM 49.85

Super SloMo

PSNR 33.21
SSIM 93.07

Unsupervised CC

PSNR 33.19
SSIM 76.67

Metric Comparison

- **Image comparison**
  - Valid 2

BMA

PSNR 32.00
SSIM 69.94

Super SloMo

PSNR 32.49
SSIM 93.30

Unsupervised CC

PSNR 37.30
SSIM 97.12

Summary

- **Super slow motion**
  - 기존엔 rule-based 방식을 통해 interpolation을 많이 진행했지만 최근엔 deep learning을 통한 방법이 활발
  - PC 뿐만 아니라 스마트폰에서도 gpu를 많이 사용함에 따라 유사한 연구가 활발히 진행
  - 성능은 검증되었으나 실질적인 사용에 있어서 model의 경량화 필요

Reference

[1] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, “Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[2] F. A. Reda, D. Sun, A. Dundar, M. Shoeybi, G. Liu, K. J. Shih, A. Tao, J. Kautz, and B. Catanzaro,“Unsupervised Video Interpolation using Cycle Consistency,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.