

동영상에서의 움직임 추적 RAFT: Optical Flow 기법



RAFT의 구조와 성능

목 차

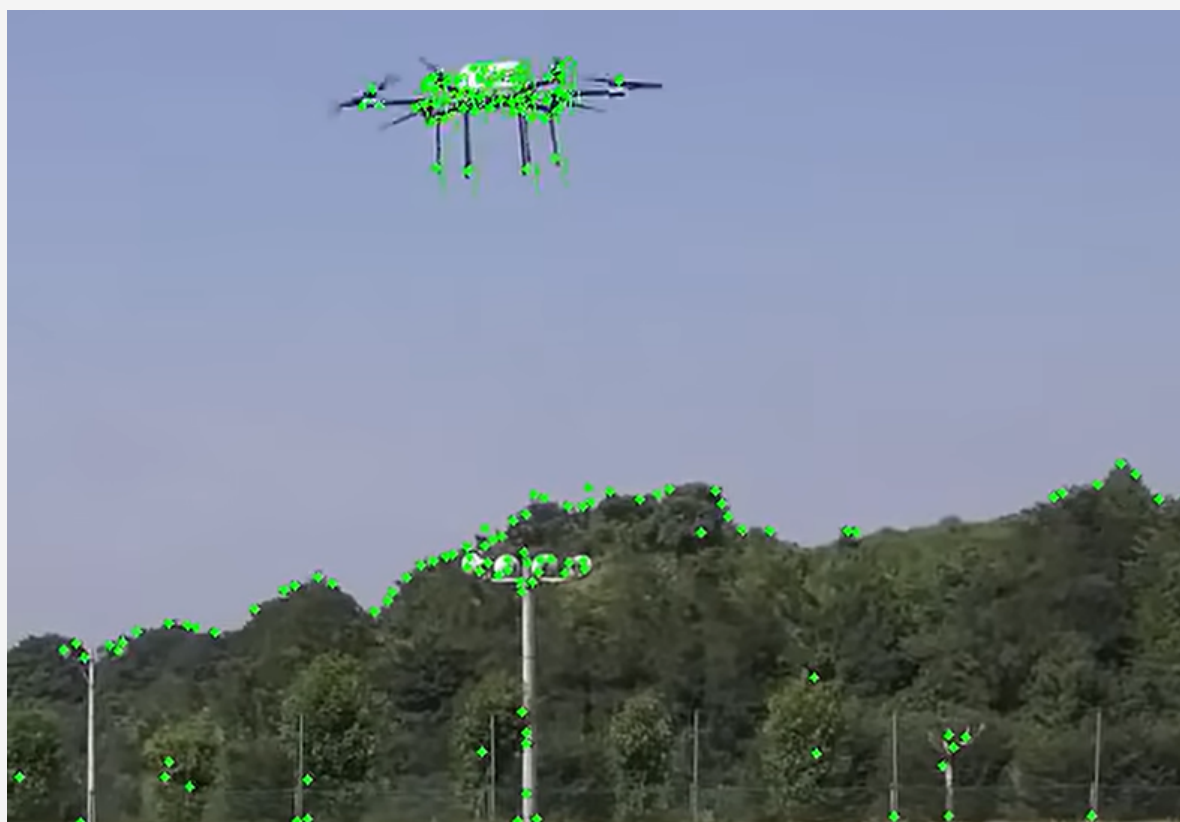
01	문제 정의
02	파이프라인
03	실험 결과
04	Ablation
05	결론

문제 정의

문제 정의

Optical Flow(광학 흐름) 추정

- 연속된 영상 프레임에서 각 픽셀이 어느 방향으로 얼마나 움직였는지 추정하는 기술입니다.



문제 정의



수중 환경에서 부유물 제거

- 1. 카메라가 이동하기 때문에 영상 전체(배경)가 함께 움직인다
- 2. Marine Snow는 카메라 움직임과는 독립적으로 불규칙하게 이동한다
- 3. 여러 프레임의 정보를 이용할 수 있다



Optical flow의 필요성

배경과 부유물은 모두 움직이기 때문에, 단순히 움직임의 유무만으로는 두 요소를 구분하기 어렵다.
깨끗한 배경을 다른 프레임에서 가져오기 위해서는 동일한 배경이 프레임 간 어디로 이동했는지를 추적하고,
이를 기반으로 정확하게 영상을 정렬(Alignment)해야 한다.
Optical Flow 추정은 이러한 픽셀 단위의 움직임을 추정하여 프레임 정렬을 가능하게 하는 핵심 기술이다.

RAFT

ECCV 2020년 Princeton University 논문

“RAFT: Recurrent All-Pairs Field Transforms for Optical Flow”

저자

Zachary Teed, Jia Deng (Princeton University)

학회

ECCV 2020 (European Conference on Computer Vision) - Best Paper Award 수상

핵심

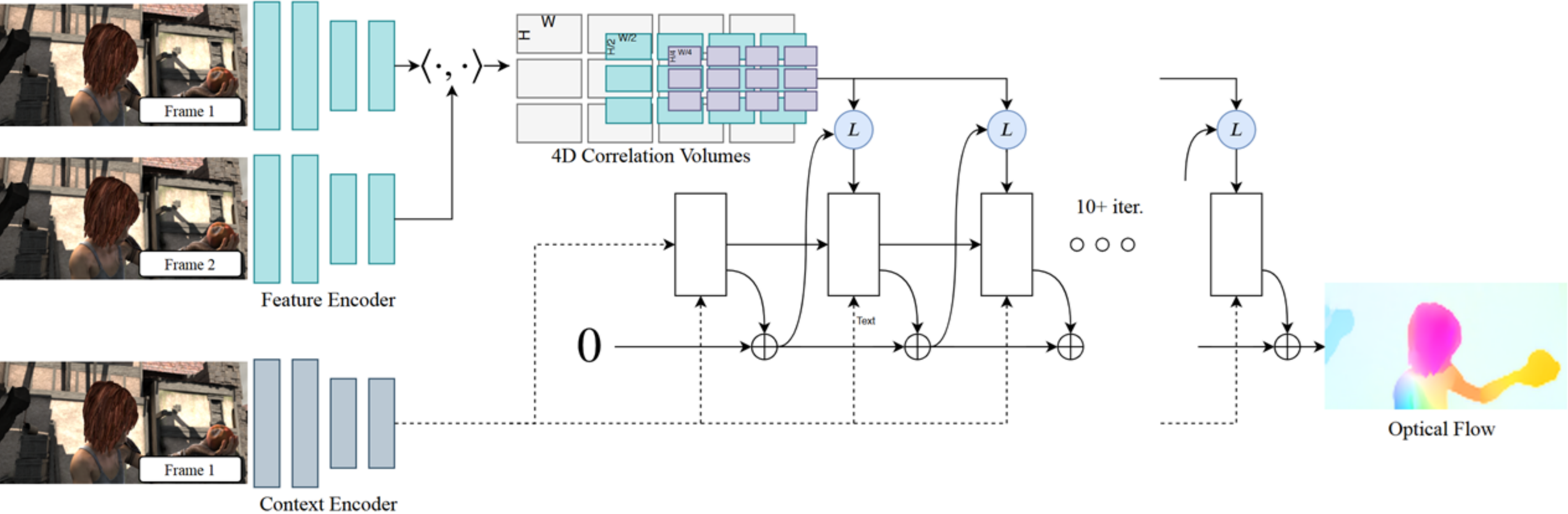
모든 픽셀 쌍의 관계를 반복적으로 갱신하여 Optical Flow를 추정하는 방법

파이프라인

RAFT

ECCV 2020년 Princeton University 논문

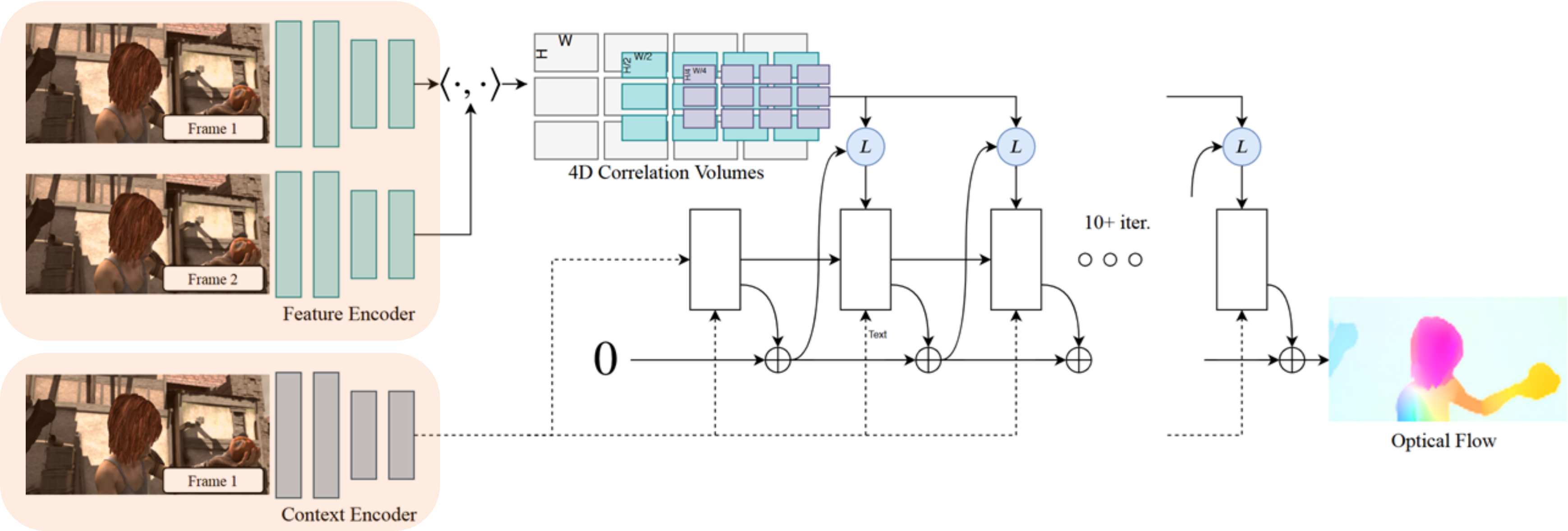
전체 파이프라인



RAFT

ECCV 2020년 Princeton University 논문

2개의 Encoder 구조



RAFT

ECCV 2020년 Princeton University 논문

2개의 Encoder 구조

두 인코더는 같은 방식으로 특징을 추출한다

Feature Encoder

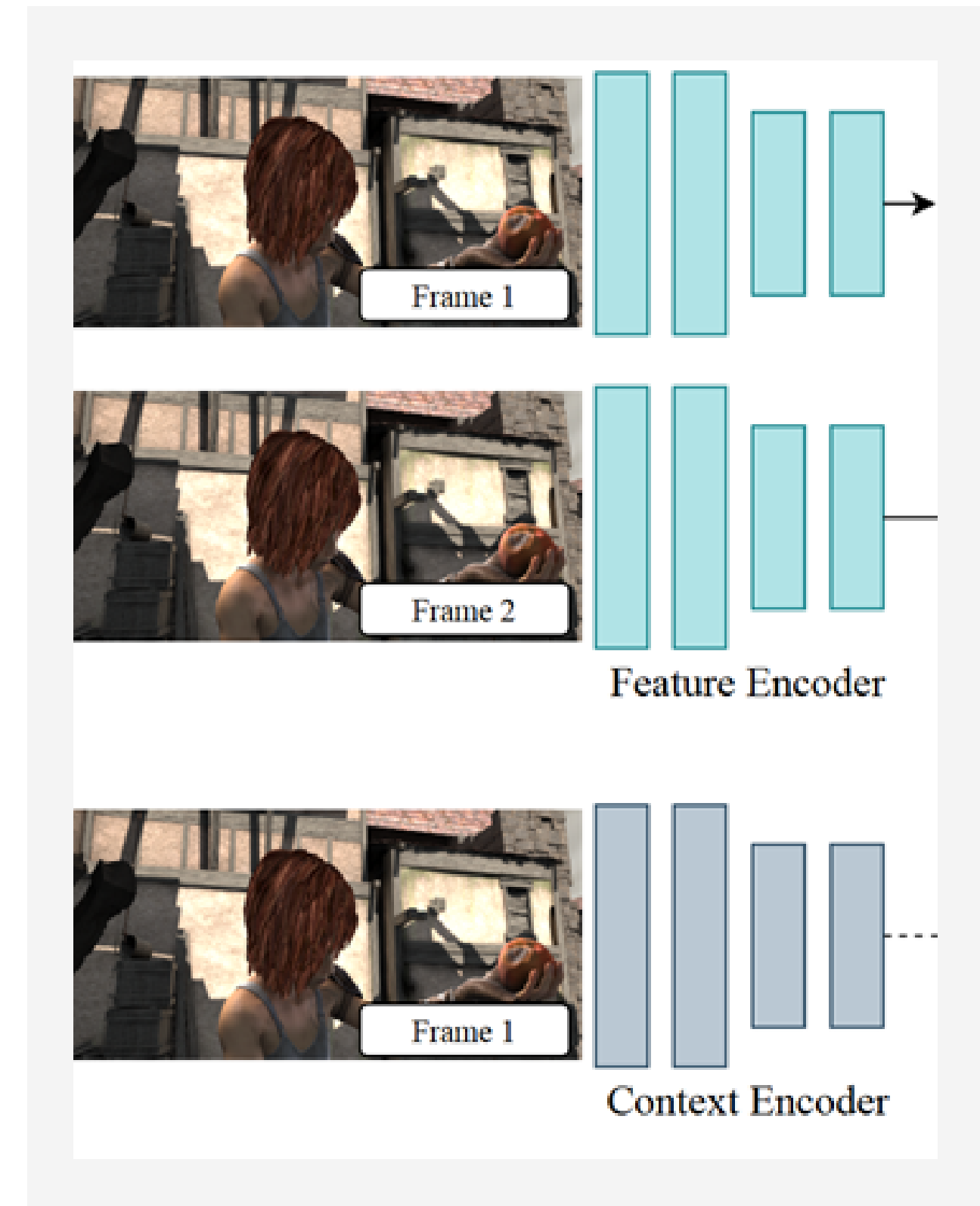
연속된 두 이미지 모두에서 특징을 추출.

=> '어디가 서로 닮았는지'를 찾는 유사도 정보를 나타낸다.

Context Encoder

첫 이미지에서만 특징을 추출.

=> '어떻게 움직이는 것이 자연스러운지'를 담은 문맥 정보를 나타낸다.



RAFT

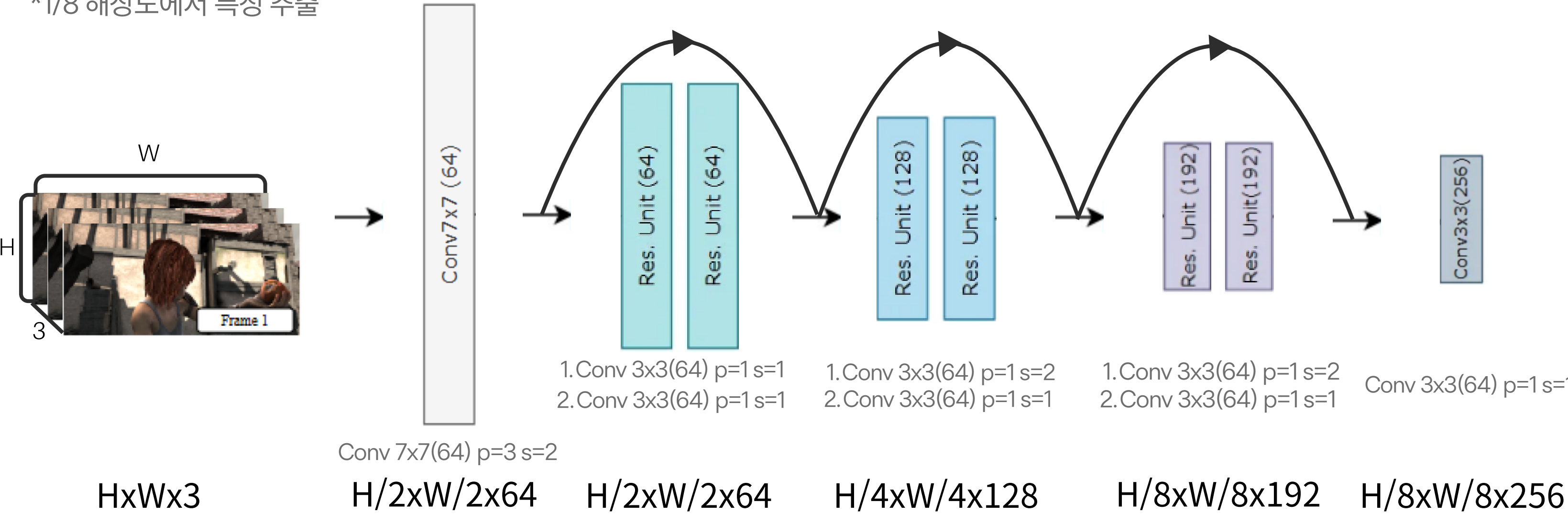
ECCV 2020년 Princeton University 논문

특징 추출

Feature Encoder와 Context Encoder 에 쓰일 특징을 추출합니다
*1/8 해상도에서 특징 추출

$$O = \frac{I - F + 2P}{S} + 1$$

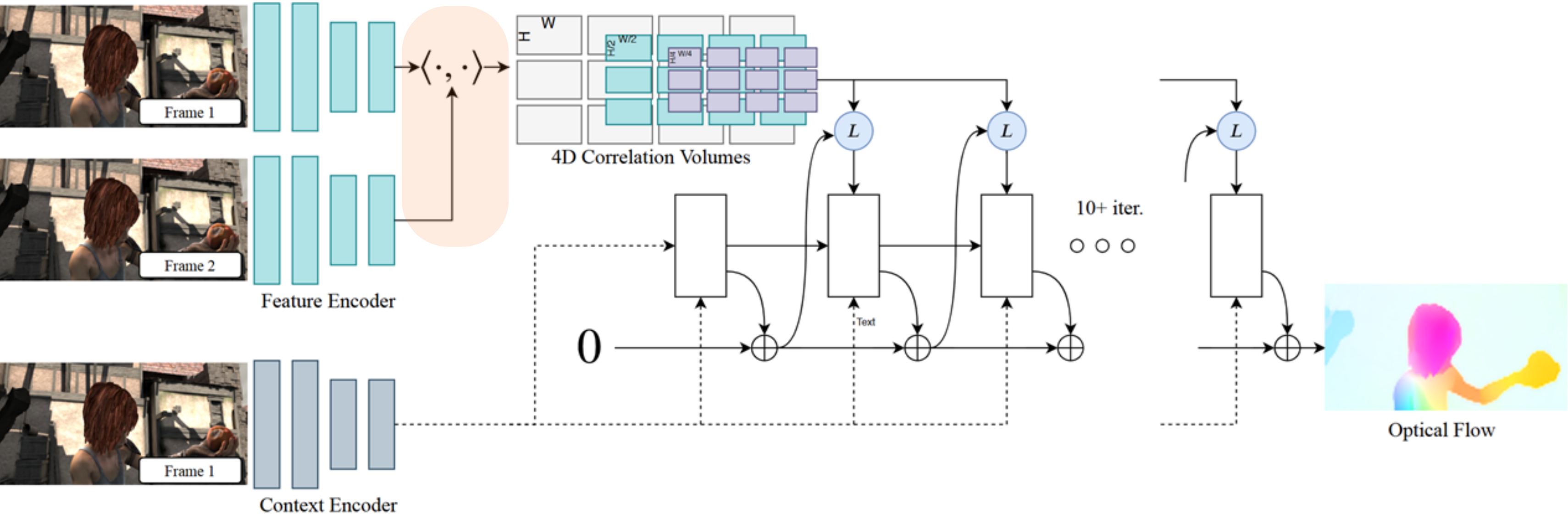
출력 Feature map 크기 공식



RAFT

ECCV 2020년 Princeton University 논문

시각적 유사도 계산

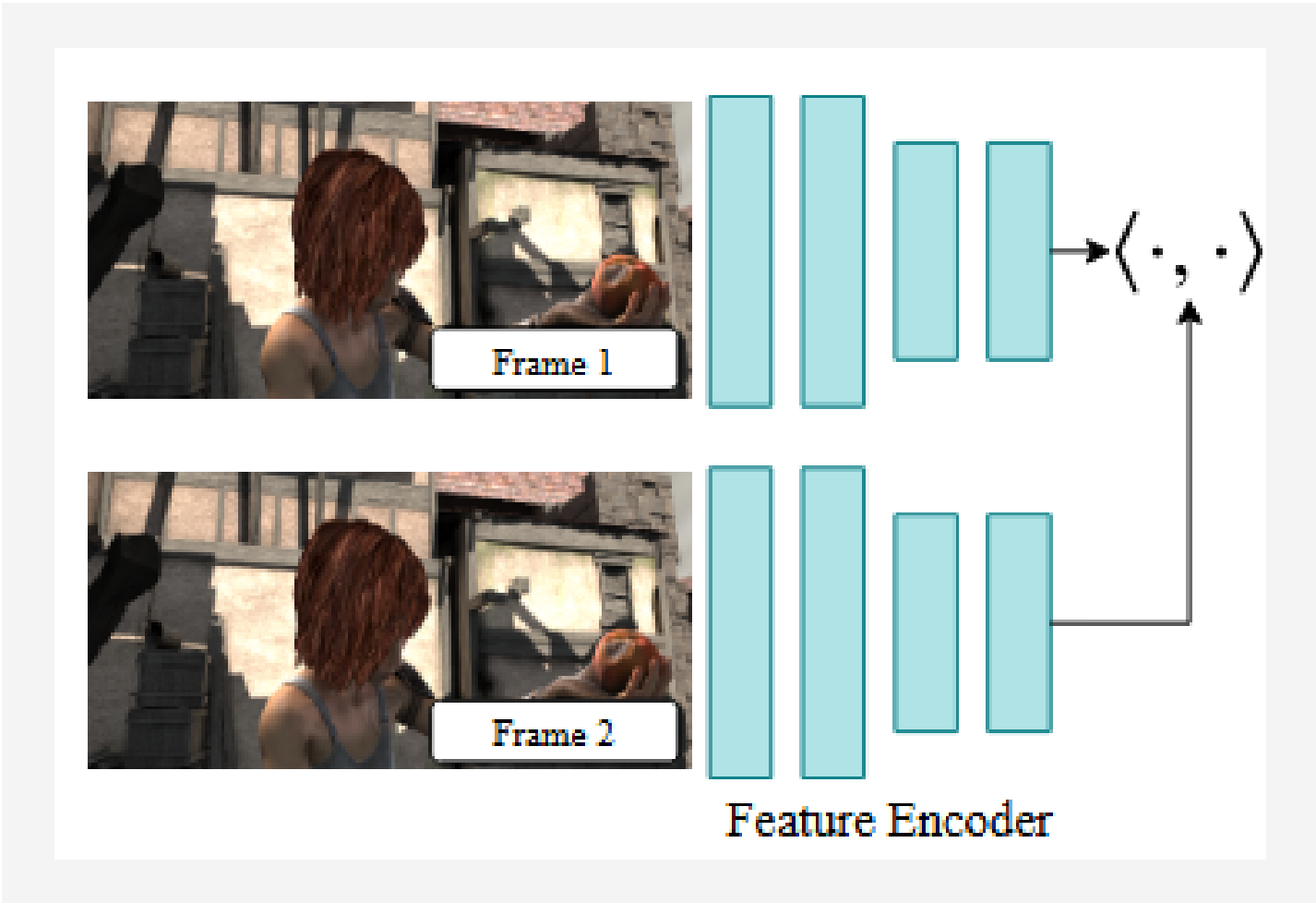


RAFT

ECCV 2020년 Princeton University 논문

시각적 유사도 계산

Feature Encoder는 연속된 프레임에서의 특징의 유사도를 계산합니다.



수식 표현

$$\begin{array}{l} g_{\theta}(I_1) \in \mathbb{R}^{H \times W \times D} \\ g_{\theta}(I_2) \in \mathbb{R}^{H \times W \times D} \end{array} \xrightarrow{\text{내적}} \mathbf{C}(g_{\theta}(I_1), g_{\theta}(I_2)) \in \mathbb{R}^{H \times W \times H \times W}$$

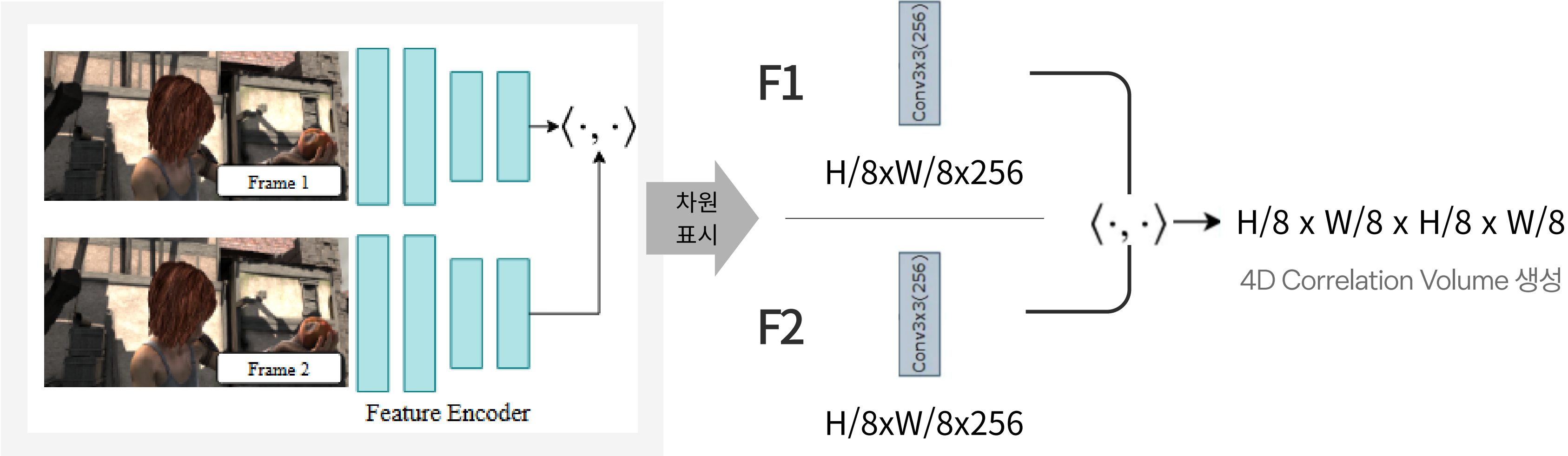
4D Correlation Volume 생성

RAFT

ECCV 2020년 Princeton University 논문

시각적 유사도 계산

Feature Encoder는 연속된 프레임에서의 특징의 유사도를 계산합니다.



RAFT

ECCV 2020년 Princeton University 논문

시각적 유사도 계산

4D Correlation Volume ($\mathbb{R}^{H \times W \times H \times W}$)

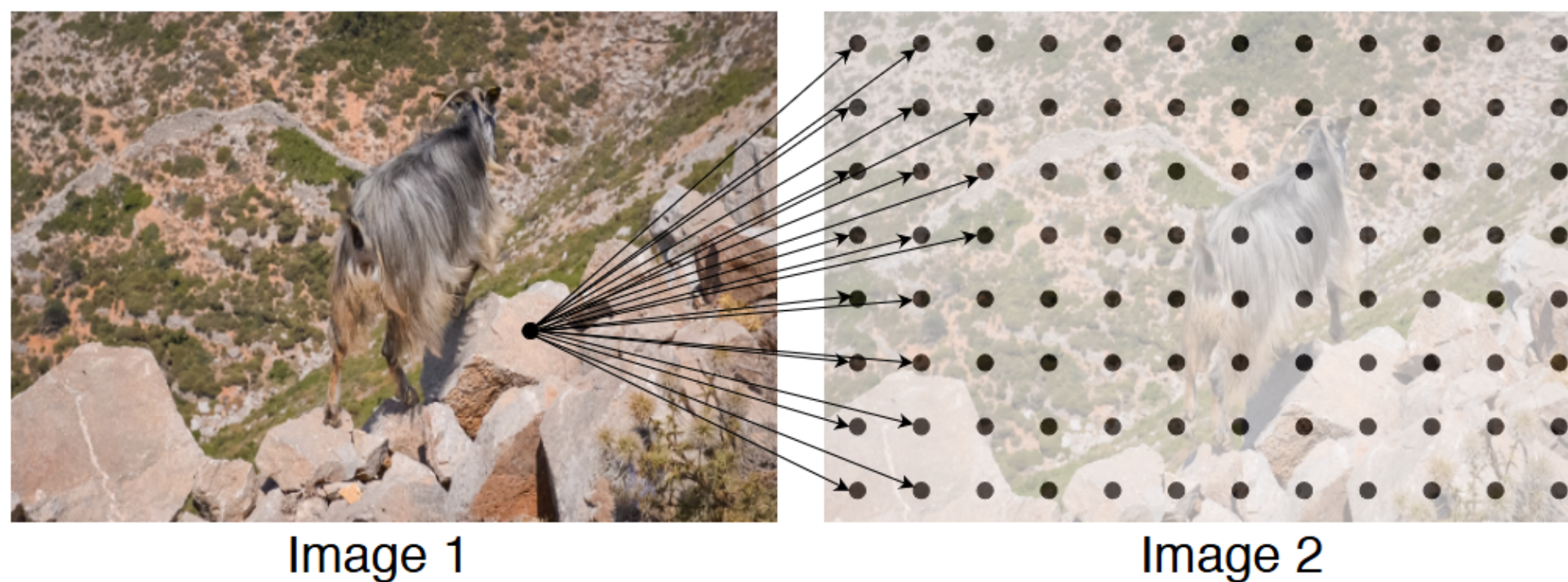
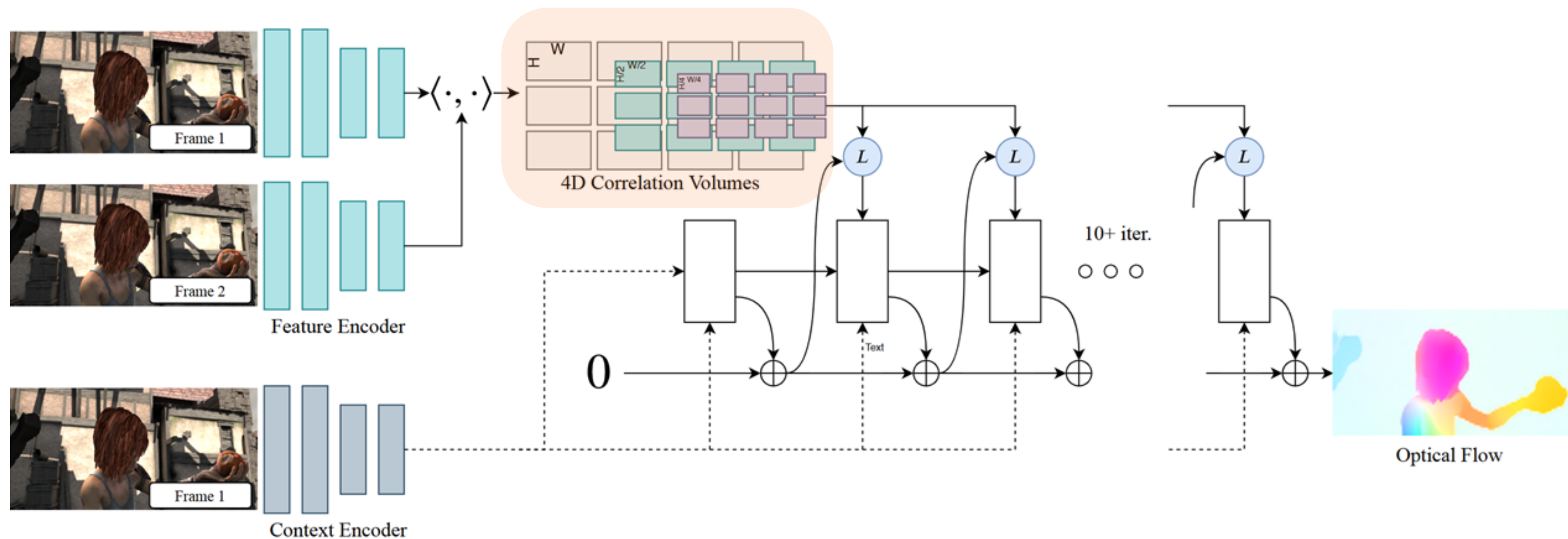


Image 1의 각 픽셀마다 Image 2의 모든 픽셀의 유사도를 담고 있기에
이후 반복적인 Optical flow 업데이트 과정에서 적절한 대응점을 제공한다.

RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup table 제작



RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup table 제작

$N(x')_r$		C1	$H \times W \times H \times W$
$N(x'/2)_r$		C2	$H \times W \times H/2 \times W/2$
$N(x'/4)_r$		C3	$H \times W \times H/4 \times W/4$
$N(x'/8)_r$		C4	$H \times W \times H/8 \times W/8$
Neighborhood 탐색		Correlation Pyramid	

Correlation Lookup table

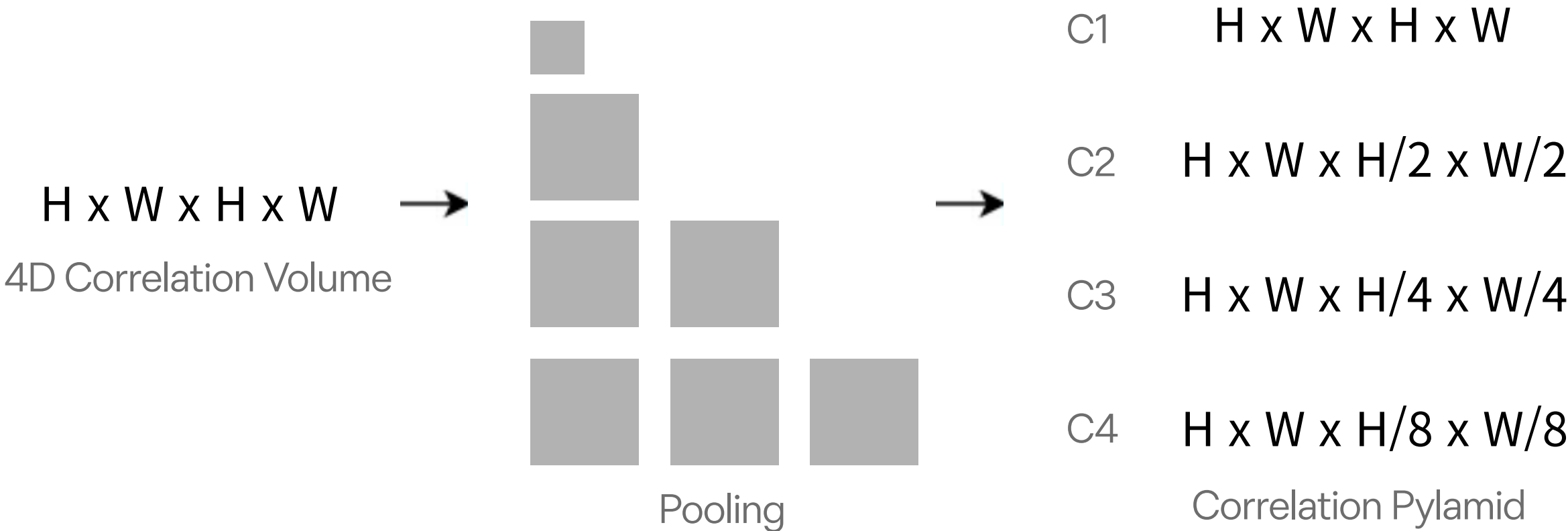
RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup table 제작

#Correlation Pylamid

마지막 두 차원에 1, 2, 4, 8 크기의 풀링을 진행하여 4계층의 피라미드 {C1, C2, C3, C4} 를 구성



RAFT

ECCV 2020년 Princeton University 논문

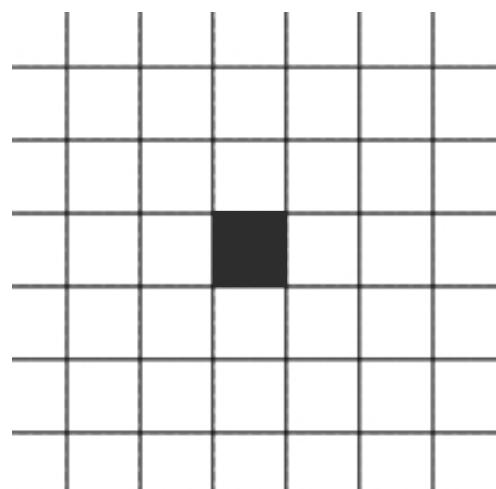
Correlation Lookup table 제작

#Neighborhood 탐색

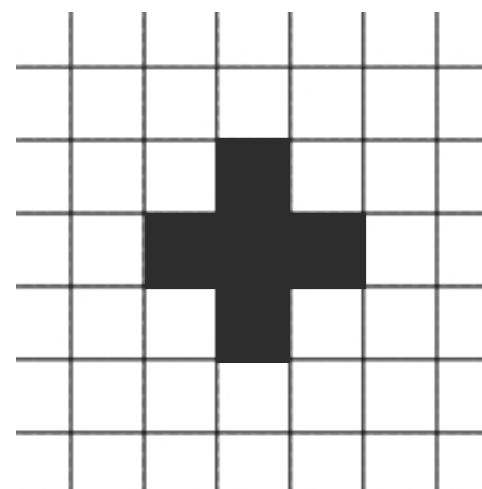
추정 대응점(x')가 Input으로 들어오면 주변의 로컬 그리드를 다음과 같이 정의한다.

$$N(x')_r = \{x' + dx \mid dx \in \mathbb{Z}^2, \|dx\|_1 \leq r\}$$

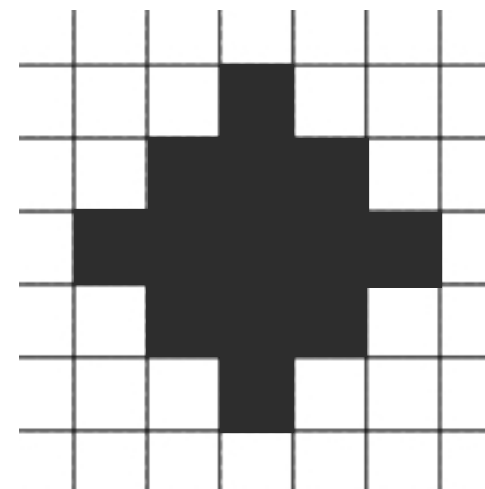
*r (Lookup radius) : 픽셀 수를 의미한다



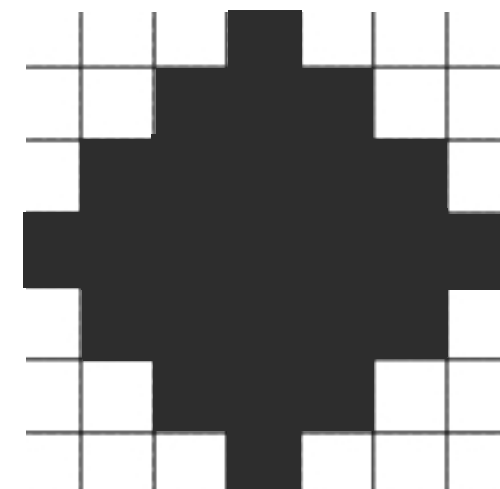
r=0



r=1



r=2



r=3

RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup table 제작

#Neighborhood 탐색

각각 다른 스케일로 $N(x')$ Correlation Pyramid에 적용됩니다. (뒤의 2개 차원에만 적용)

$N(x')_r$		C1	$H \times W \times H \times W$
$N(x'/2)_r$		C2	$H \times W \times H/2 \times W/2$
$N(x'/4)_r$		C3	$H \times W \times H/4 \times W/4$
$N(x'/8)_r$		C4	$H \times W \times H/8 \times W/8$

Neighborhood 탐색

Correlation Pyramid

모델은 $N(x')$ 를 계산 후 대응되는 Correlation Pyramid만을 반환합니다.

Correlation Lookup table

RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup table 제작

#Correlation Lookup table의 시사점

1. 정밀도 유지와 큰 탐색 범위

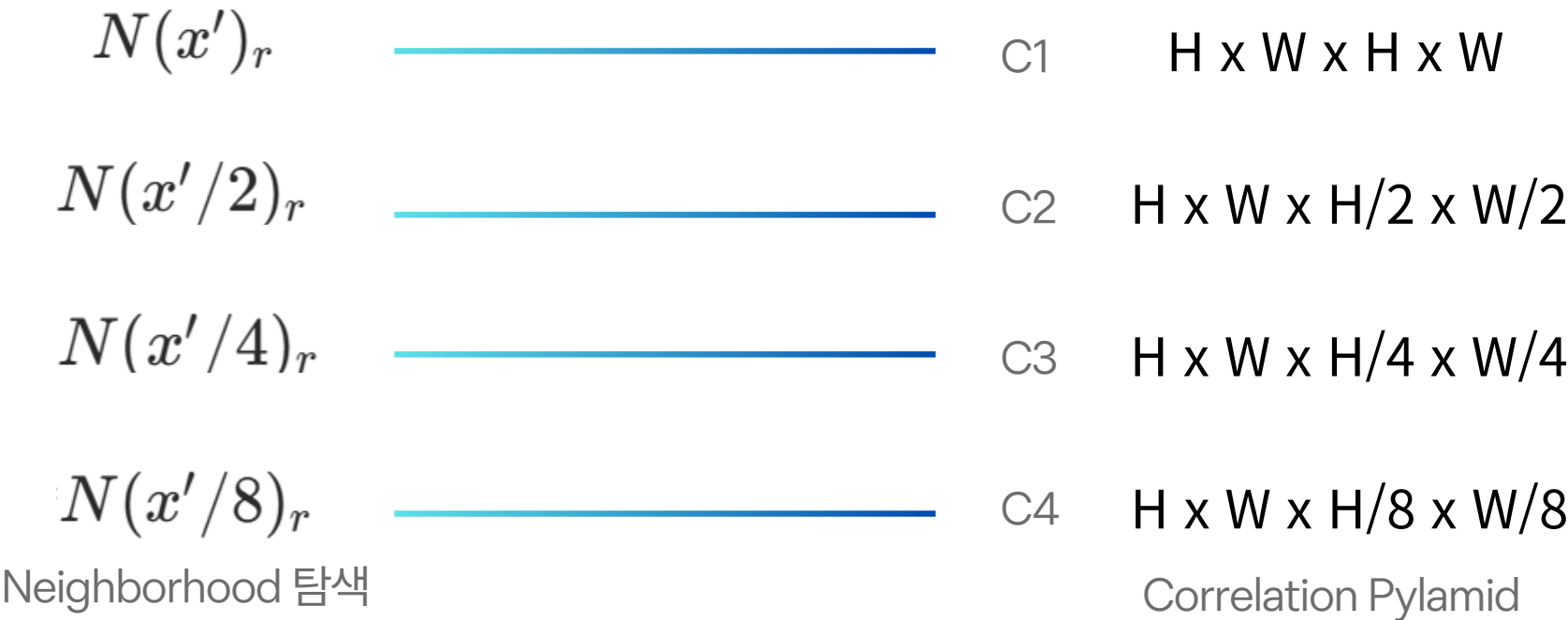
→ Correlation Pyramid : C1은 정밀도,
C4는 큰 Range를 보기 적합하다.

2. 고해상도 유지

→ Motion Boundary(두 물체의 이동 속도나 방향이 달라서 생기는 경계선)
가 뭉개지지 않는다.

3. 보여줄 부분만 모델에 전달

→ 효율적으로 연산을 할 수 있도록 합니다.

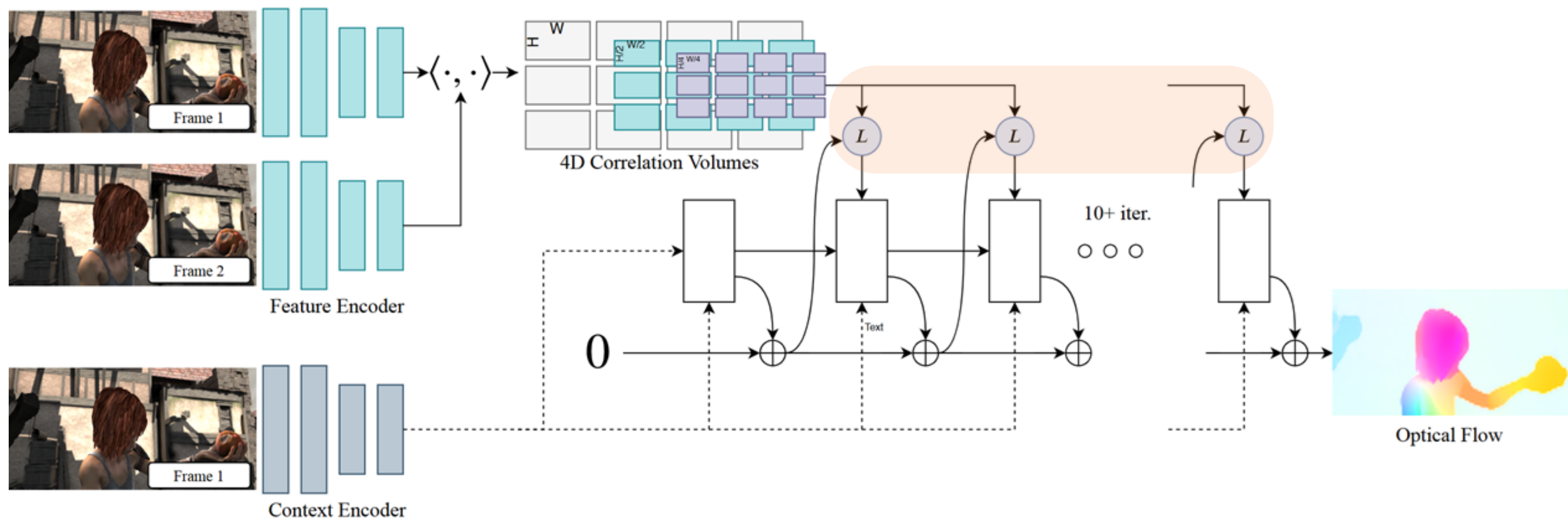


Correlation Lookup table

RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup Operator

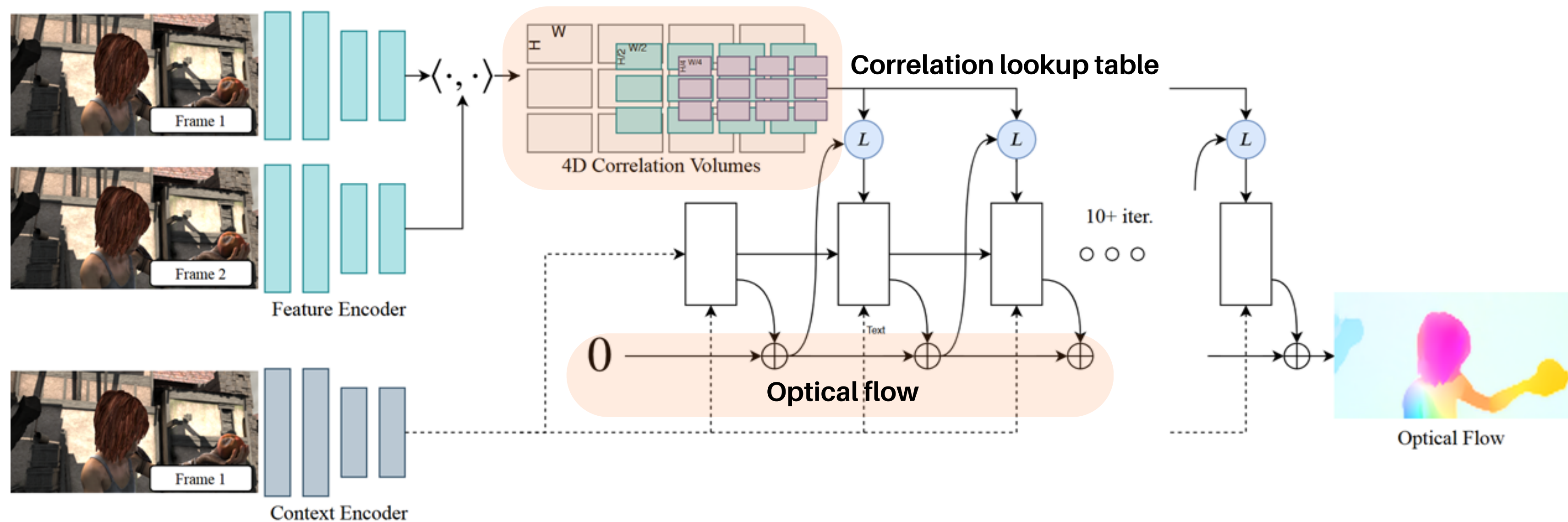


RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup operator

Lookup Operator는 Optical flow와 Correlation lookup table을 입력 받습니다.



RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup operator

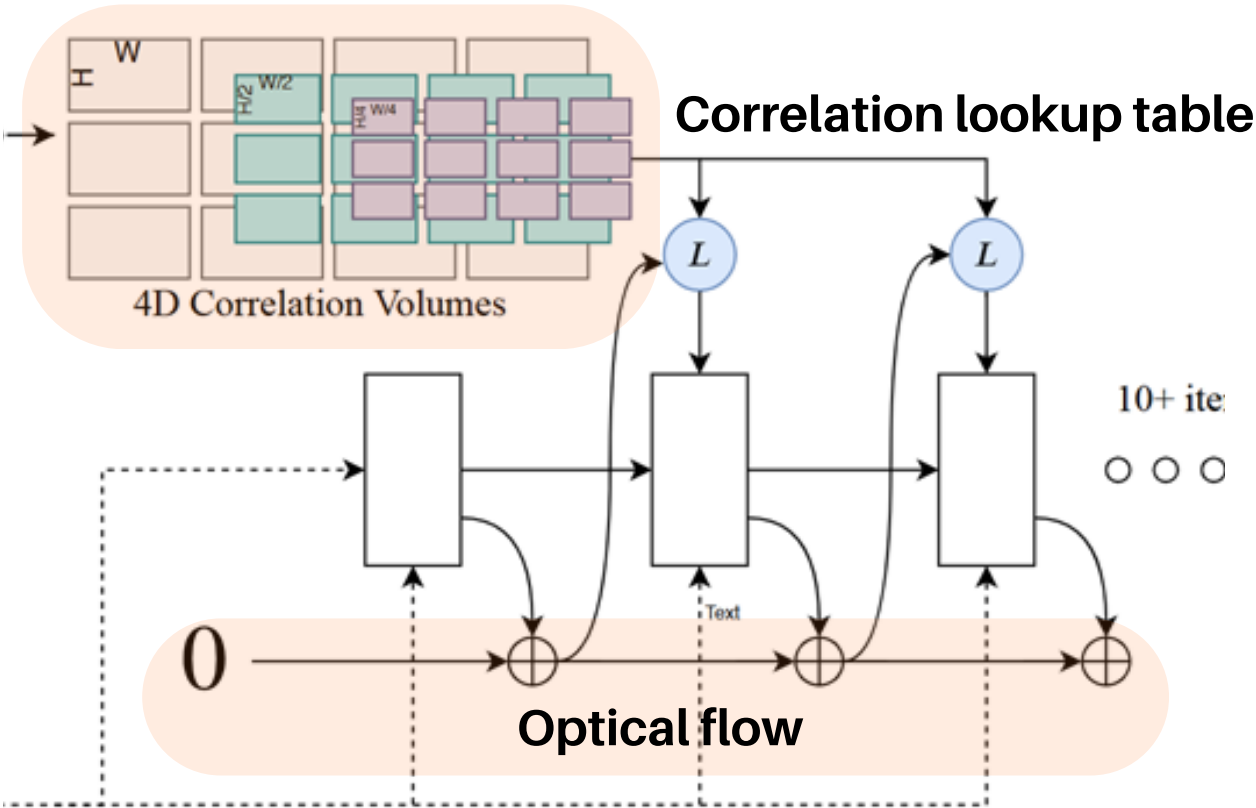
1. Optical flow로 예상 대응 위치인 (u', v') 를 찾습니다.

$$(u', v') = (u + f_1(u), v + f_2(v))$$

2. (u', v') 를 $N(x')$ 에 넣어 대응되는 C1, C2, C3, C4를 반환합니다

$N(x')_r$	_____	C1	$H \times W \times H \times W$
$N(x'/2)_r$	_____	C2	$H \times W \times H/2 \times W/2$
$N(x'/4)_r$	_____	C3	$H \times W \times H/4 \times W/4$
$N(x'/8)_r$	_____	C4	$H \times W \times H/8 \times W/8$
Neighborhood 탐색		Correlation Pyramid	

Correlation Lookup table



RAFT

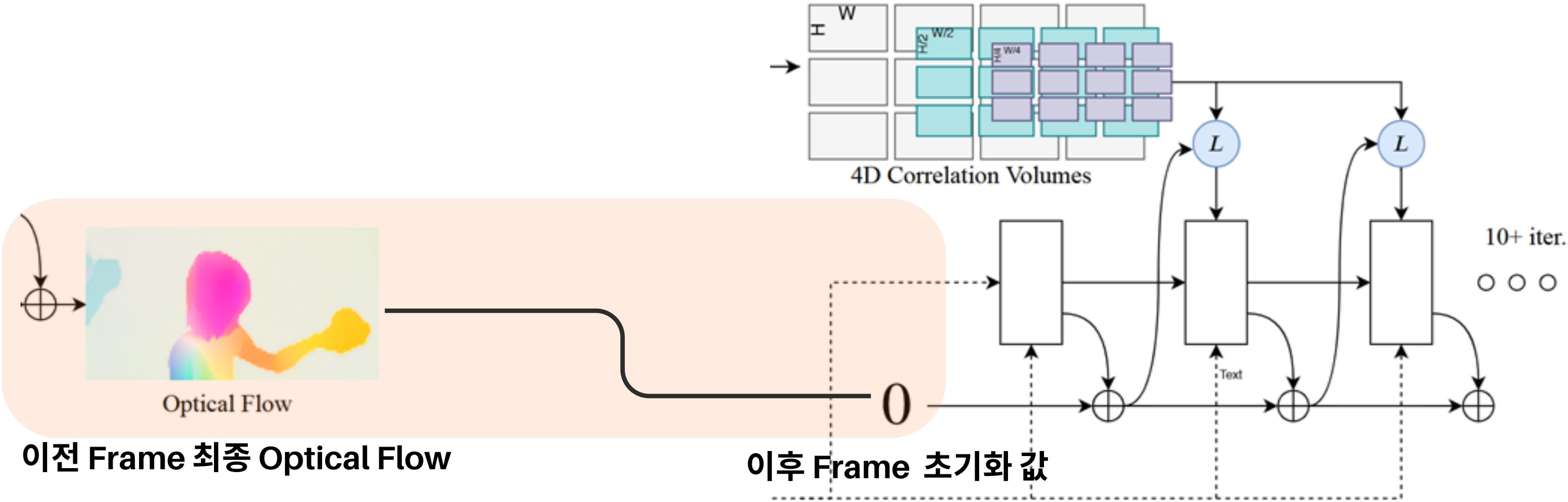
ECCV 2020년 Princeton University 논문

Correlation Lookup operator

#Warm-Start

현재 입력으로 들어오는 Optical flow를 이전 frame의 최종 Optical Flow로 초기화 시킵니다.

→ 빠른 수렴/Occlusion gap 해결



RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup operator

#Warm Start 전 추가 처리 과정

가져온 이전 Frame의 Flow를 전방 투영, 없는부분을
Nearest Neighbor Interpolation(최근접 이웃 보간)을 통해 채워넣어
Occlusion gap(폐색 간극)을 해결합니다.

→ *Occlusion*(폐색, 가림현상) 이란?

이전 프레임에는 있던 객체가 다른 객체에 의해 가려지는 현상.
이로 인해 Flow가 정확히 계산될 수 없다.

이전 Frame
Optical Flow를
전방 투영한 이미지



객체가 가려짐



이후 Optical Flow

?

Flow 계산 불가

RAFT

ECCV 2020년 Princeton University 논문

Correlation Lookup operator

#Occlusion gap 해결

Warmup 때 가져온 이전 Frame의 Flow를 전방 투영, 없는부분을
Nearest Neighbor Interpolation(최근접 이웃 보간)을 통해 채워넣어
Occlusion gap(폐색 간극)을 해결합니다.

이전 Frame
Optical Flow를
전방 투영한 이미지



객체가 가려짐

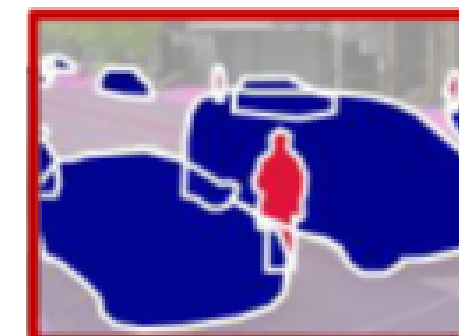


가려진 부분에 대해
가장 근접한 픽셀로 채워넣음

Nearest Neighbor Interpolation 수행



이후 Optical Flow

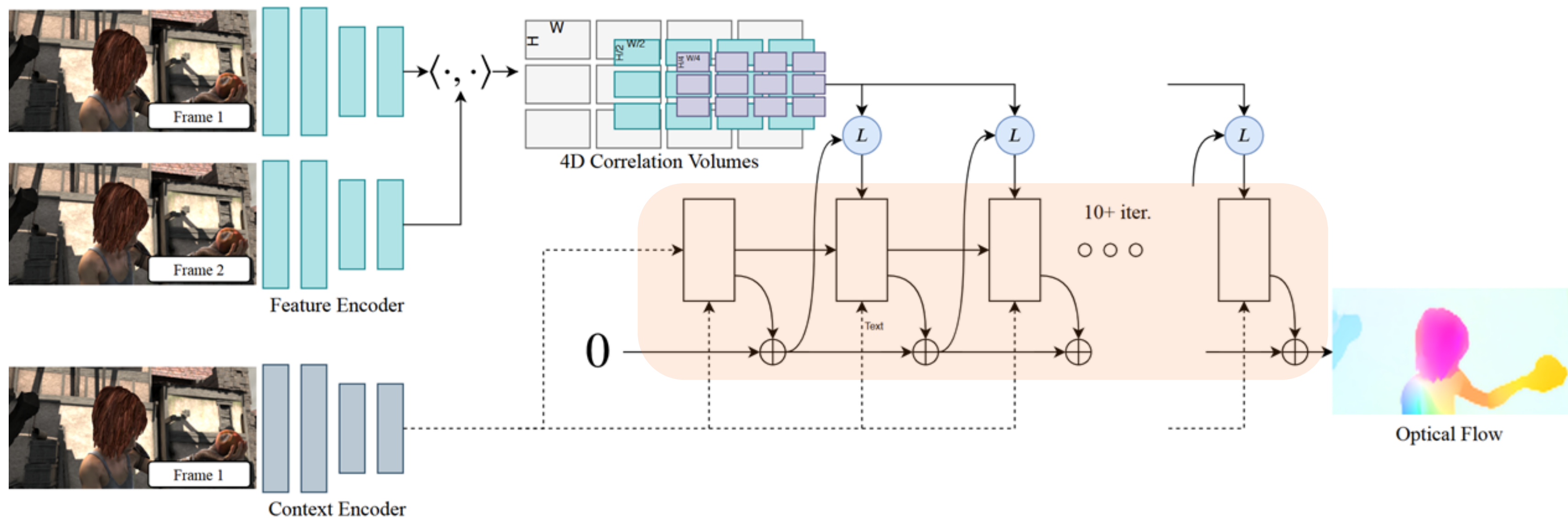


Flow 계산 가능

RAFT

ECCV 2020년 Princeton University 논문

Separable Conv GRU

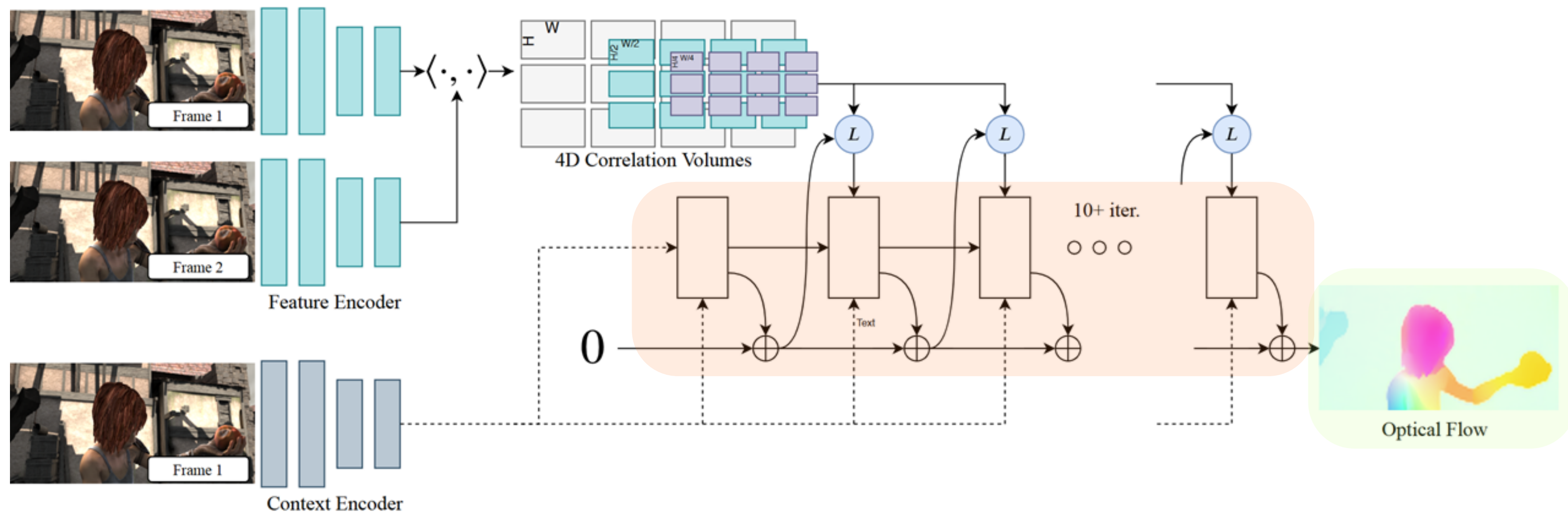


RAFT

ECCV 2020년 Princeton University 논문

Separable Conv GRU

Separable ConvGRU는 반복적 업데이트를 통해 최종 Optical flow를 산출하는 핵심 블록



RAFT

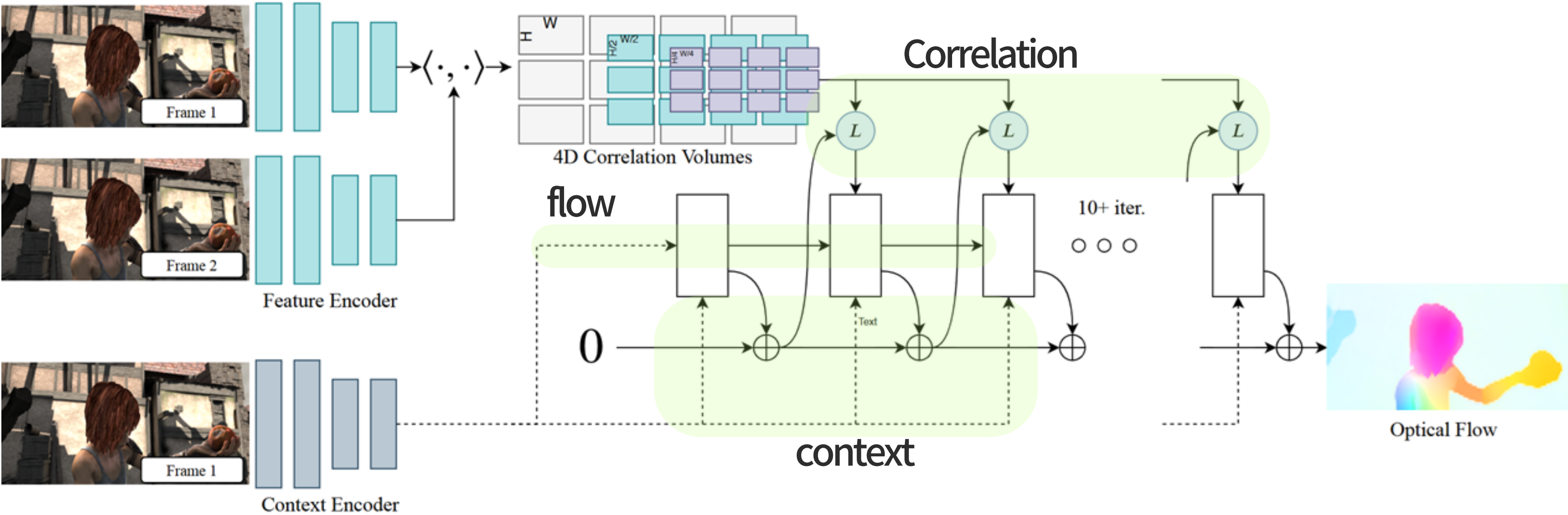
ECCV 2020년 Princeton University 논문

Separable Conv GRU

총 3개의 피쳐를 인풋으로 받아 최종 Optical flow를 산출합니다.

$$\mathbf{x}_t = [\mathbf{F}_{corr}, \mathbf{F}_{flow}, \mathbf{F}_{context}]$$

현재 시점의 인풋



RAFT

ECCV 2020년 Princeton University 논문

Separable Conv GRU

#GRU

필요한 정보만 기억하고 불필요한 정보는 잊도록 설계된 순환 신경망

한계: 입력을 1차원 벡터로 처리하므로 이미지의 공간 정보(H×W)를 유지하지 못함

Update Gate 계산

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

Reset Gate 계산

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

Candidate Hidden State 생성

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h)$$

최종 Hidden State 업데이트

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

RAFT

ECCV 2020년 Princeton University 논문

Separable Conv GRU

#Conv GRU → 3차원에서 사용하는 GRU

GRU의 Linear 연산을 Convolution으로 대체하여 Feature Map(C×H×W) 을 직접 처리하는 구조
한계: 3×3 Conv만 사용할 경우 수용영역이 좁아 장거리 공간 관계를 학습하기 어려움

Update Gate 계산	$z_t = \sigma(\text{Conv}_{3 \times 3}([h_{t-1}, x_t], W_z))$
Reset Gate 계산	$r_t = \sigma(\text{Conv}_{3 \times 3}([h_{t-1}, x_t], W_r))$
Candidate Hidden State 생성	$\tilde{h}_t = \tanh(\text{Conv}_{3 \times 3}([r_t \odot h_{t-1}, x_t], W_h))$
최종 Hidden State 업데이트	$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$

RAFT

ECCV 2020년 Princeton University 논문

Separable Conv GRU

#Separable Conv GRU → 기존 3x3 Conv 을 1x5 , 5x1 로 나누어서 실행
더 넓은 수용영역을 효율적으로 확보한 구조

Update Gate 계산

$$z_t = \sigma(\text{Conv}_{1 \times 5}([h_{t-1}, x_t], W_z))$$

Reset Gate 계산

$$r_t = \sigma(\text{Conv}_{1 \times 5}([h_{t-1}, x_t], W_r))$$

Candidate Hidden State 생성

$$\tilde{h}_t = \tanh(\text{Conv}_{1 \times 5}([r_t \odot h_{t-1}, x_t], W_h))$$

최종 Hidden State 업데이트

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

$$z_t = \sigma(\text{Conv}_{5 \times 1}([h_t, x_t], W_z))$$

$$r_t = \sigma(\text{Conv}_{5 \times 1}([h_t, x_t], W_r))$$

$$\tilde{h}_t = \tanh(\text{Conv}_{5 \times 1}([r_t \odot h_t, x_t], W_h))$$

$$h_t = (1 - z_t) \odot h_t + z_t \odot \tilde{h}_t$$

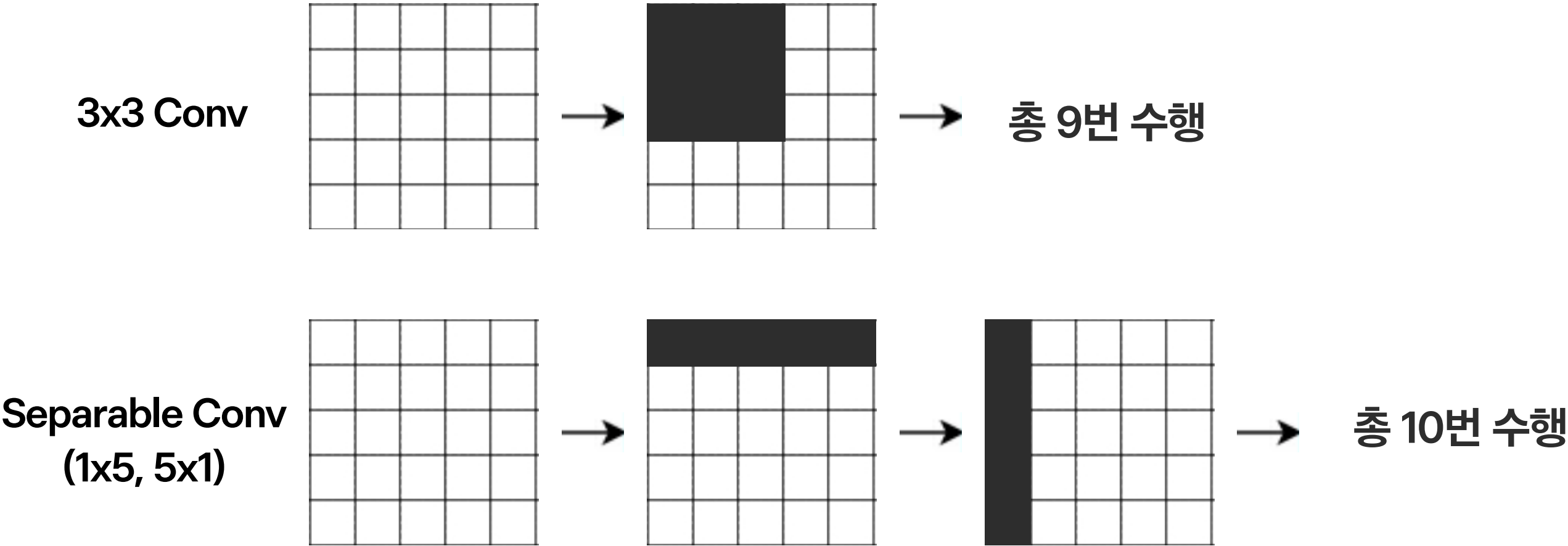
RAFT

ECCV 2020년 Princeton University 논문

Separable Conv GRU

#Separable Conv GRU → 기존 3x3 Conv 을 1x5 , 5x1 로 나누어서 실행
더 넓은 수용영역을 효율적으로 확보한 구조

<5x5격자에서의 예시>



연산량을 크게 늘리지 않으면서
수용영역 증가
즉, 더 넓은 영역의 정보를 볼 수 있다.

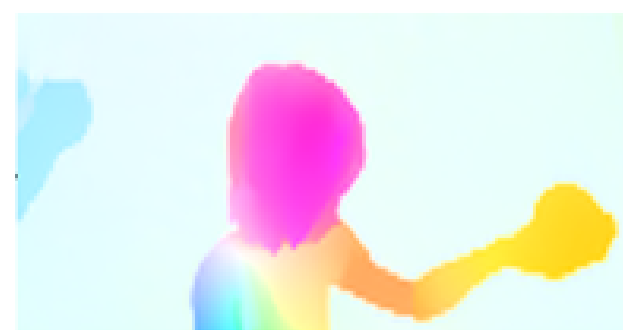
RAFT

ECCV 2020년 Princeton University 논문

Upsampling & 지도학습

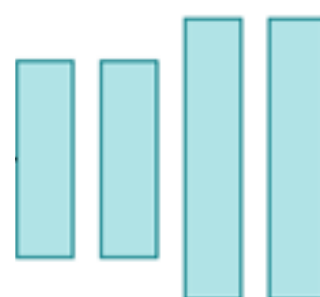
#Upsampling

Convex Upsampling: Ground Truth Optical Flow 데이터셋을 이용해 학습된 CNN이 Upsampling 가중치를 예측합니다.



Optical Flow

$H/8 \times W/8$



Optical Flow

$H \times W$

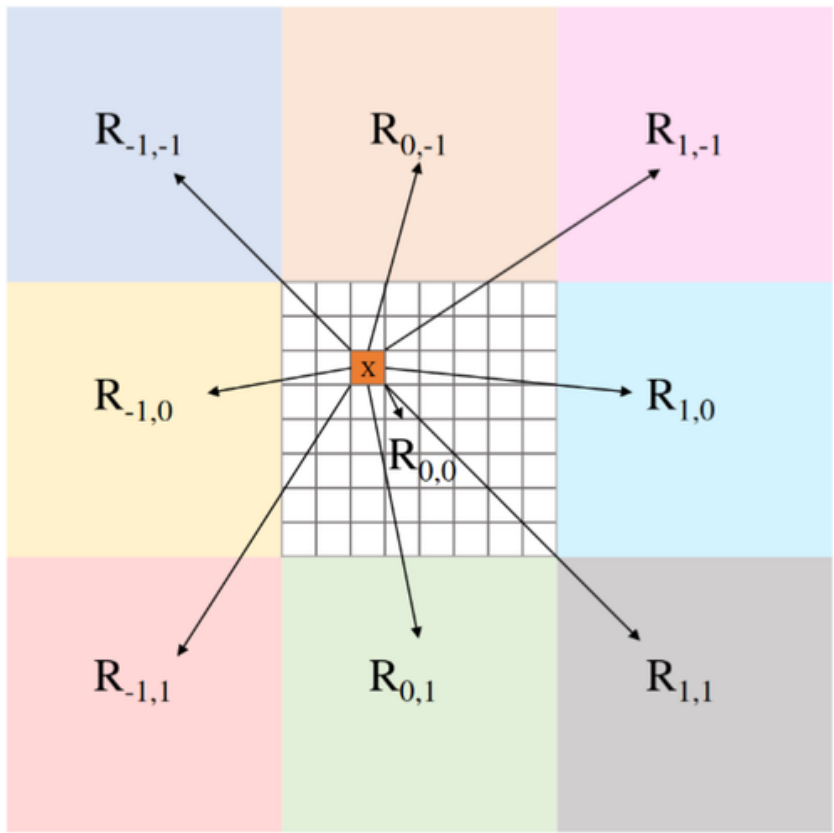
RAFT

ECCV 2020년 Princeton University 논문

Upsampling & 지도학습

#Upsampling

Convex Upsampling: Ground Truth Optical Flow 데이터셋을 이용해 학습된 CNN이 Upsampling 가중치를 예측합니다.



*w는 Softmax를 통과한 값

$$x = \sum_{i=-1}^1 \sum_{j=-1}^1 w_{i,j} \cdot R_{i,j}$$

RAFT

ECCV 2020년 Princeton University 논문

Upsampling & 지도학습

#지도학습

:예측한 Optical Flow를 Ground Truth Optical Flow와 비교하여 오차를 최소화하도록 네트워크 전체를 지도학습합니다."



예측한 Optical Flow

VS



Ground Truth Optical Flow

N: GRU 블록 반복수

$$\mathcal{L} = \sum_{i=1}^N \gamma^{N-i} \|\mathbf{f}_{gt} - \mathbf{f}_i\|_1$$

픽셀 값의
오차 범위

Experiments

결과

1st

#Benchmark
Sintel: CG 기반 데이터셋
KITTI: 실제 도로 주행 환경을 촬영한 데이터셋
Sintel은 가려진 픽셀에 대해서도
Optical flow 채점 실행
KITTI는 보여지는 픽셀에만 채점 실행

Training Data
C + T: FlyingChairs(C), FlyingThings(T)
S/K/H: Sintel(S), KITTI(K), HD1K(H)
데이터를 추가하여 정밀 튜닝
H(*고해상도 주행 데이터셋)

Training Data	Method	Sintel (train)		KITTI-15 (train)		Sintel (test)		KITTI-15 (test)
		Clean	Final	F1-epe	F1-all	Clean	Final	F1-all
-	FlowFields[7]	-	-	-	-	3.75	5.81	15.31
-	FlowFields++[40]	-	-	-	-	2.94	5.49	14.82
S	DCFlow[47]	-	-	-	-	3.54	5.12	14.86
S	MRFlow[46]	-	-	-	-	2.53	5.38	12.19
C + T	HD3[50]	3.84	8.77	13.17	24.0	-	-	-
	LiteFlowNet[22]	2.48	4.04	10.39	28.5	-	-	-
	PWC-Net[42]	2.55	3.93	10.35	33.7	-	-	-
	LiteFlowNet2[23]	2.24	3.78	8.97	25.9	-	-	-
	VCN[49]	2.21	3.68	8.36	25.1	-	-	-
	MaskFlowNet[52]	2.25	3.61	-	23.1	-	-	-
	FlowNet2[25]	2.02	3.54 ¹	10.08	30.0	3.96	6.02	-
	Ours (small)	2.21	3.35	7.51	26.9	-	-	-
	Ours (2-view)	1.43	2.71	5.04	17.4	-	-	-
C+T+S/K	FlowNet2 [25]	(1.45)	(2.01)	(2.30)	(6.8)	4.16	5.74	11.48
	HD3 [50]	(1.87)	(1.17)	(1.31)	(4.1)	4.79	4.67	6.55
	IRR-PWC [24]	(1.92)	(2.51)	(1.63)	(5.3)	3.84	4.58	7.65
	ScopeFlow[8]	-	-	-	-	3.59	4.10	6.82
	Ours (2-view)	(0.77)	(1.20)	(0.64)	(1.5)	2.08	3.41	5.27
C+T+S+K+H	LiteFlowNet2 ² [23]	(1.30)	(1.62)	(1.47)	(4.8)	3.48	4.69	7.74
	PWC-Net+[41]	(1.71)	(2.34)	(1.50)	(5.3)	3.45	4.60	7.72
	VCN [49]	(1.66)	(2.24)	(1.16)	(4.1)	2.81	4.40	6.30
	MaskFlowNet[52]	-	-	-	-	2.52	4.17	6.10
	Ours (2-view)	(0.76)	(1.22)	(0.63)	(1.5)	1.94	3.18	5.10
	Ours (warm-start)	(0.77)	(1.27)	-	-	1.61	2.86	-

결과

1st

평가지표

Clean: Clean Mean EPE
Final: Final Mean EPE
F1-epe: train set에서 epe
F1-all: 테스트 set에서 epe

$$EPE = \sqrt{(u - u_{gt})^2 + (v - v_{gt})^2}$$

각 픽셀의 예측 오차(픽셀 단위)

$$F1-all = \frac{\text{Outlier Pixels}}{\text{Valid Ground Truth Pixels}} \times 100\%$$

Outlier: EPE > 3px 이고 상대오차 > 5%

Training Data	Method	Sintel (train)		KITTI-15 (train)		Sintel (test)		KITTI-15 (test)
		Clean	Final	F1-epe	F1-all	Clean	Final	F1-all
-	FlowFields[7]	-	-	-	-	3.75	5.81	15.31
-	FlowFields++[40]	-	-	-	-	2.94	5.49	14.82
S	DCFlow[47]	-	-	-	-	3.54	5.12	14.86
S	MRFlow[46]	-	-	-	-	2.53	5.38	12.19
C + T	HD3[50]	3.84	8.77	13.17	24.0	-	-	-
	LiteFlowNet[22]	2.48	4.04	10.39	28.5	-	-	-
	PWC-Net[42]	2.55	3.93	10.35	33.7	-	-	-
	LiteFlowNet2[23]	2.24	3.78	8.97	25.9	-	-	-
	VCN[49]	2.21	3.68	8.36	25.1	-	-	-
	MaskFlowNet[52]	2.25	3.61	-	23.1	-	-	-
	FlowNet2[25]	2.02	3.54 ¹	10.08	30.0	3.96	6.02	-
	Ours (small)	2.21	3.35	7.51	26.9	-	-	-
	Ours (2-view)	1.43	2.71	5.04	17.4	-	-	-
C+T+S/K	FlowNet2 [25]	(1.45)	(2.01)	(2.30)	(6.8)	4.16	5.74	11.48
	HD3 [50]	(1.87)	(1.17)	(1.31)	(4.1)	4.79	4.67	6.55
	IRR-PWC [24]	(1.92)	(2.51)	(1.63)	(5.3)	3.84	4.58	7.65
	ScopeFlow[8]	-	-	-	-	3.59	4.10	6.82
	Ours (2-view)	(0.77)	(1.20)	(0.64)	(1.5)	2.08	3.41	5.27
C+T+S+K+H	LiteFlowNet2 ² [23]	(1.30)	(1.62)	(1.47)	(4.8)	3.48	4.69	7.74
	PWC-Net+[41]	(1.71)	(2.34)	(1.50)	(5.3)	3.45	4.60	7.72
	VCN [49]	(1.66)	(2.24)	(1.16)	(4.1)	2.81	4.40	6.30
	MaskFlowNet[52]	-	-	-	-	2.52	4.17	6.10
	Ours (2-view)	(0.76)	(1.22)	(0.63)	(1.5)	1.94	3.18	5.10
	Ours (warm-start)	(0.77)	(1.27)	-	-	1.61	2.86	-

Ablations

Ablations

C로 학습 100,000 번 학습, K로 60,000번 추가 학습

반복적으로 Flow를 업데이트하는 연산자

반복마다 같은 가중치를 사용할지

Context Encoder 사용할지

Feature를 단일 해상도로 사용할지 여러 해상도로 사용할지

Correlation Lookup 시 주변 몇 칸을 참조할지

Correlation Pyramid(다중 해상도)를 사용할지

Correlation을 얼마나 넓게 계산할지

업데이트 시 어떤 정보를 사용할지

C로 학습 100,000 번 학습, K로 100,000번 추가 학습

최종 Flow를 확대하는 방법

추론 시 반복 횟수

Experiment	Method	Sintel (train)		KITTI-15 (train)		Parameters
		Clean	Final	F1-epe	F1-all	
Reference Model (bilinear upsampling), Training: 100k(C) $\rightarrow$ 60k(T)						
Update Op.	ConvGRU	1.63	2.83	5.54	19.8	4.8M
	Conv	2.04	3.21	7.66	26.1	4.1M
Tying	Tied Weights	1.63	2.83	5.54	19.8	4.8M
	Untied Weights	1.96	3.20	7.64	24.1	32.5M
Context	Context	1.63	2.83	5.54	19.8	4.8M
	No Context	1.93	3.06	6.25	23.1	3.3M
Feature Scale	Single-Scale	1.63	2.83	5.54	19.8	4.8M
	Multi-Scale	2.08	3.12	6.91	23.2	6.6M
Lookup Radius	0	3.41	4.53	23.6	44.8	4.7M
	1	1.80	2.99	6.27	21.5	4.7M
	2	1.78	2.82	5.84	21.1	4.8M
	4	1.63	2.83	5.54	19.8	4.8M
Correlation Pooling	No	1.95	3.02	6.07	23.2	4.7M
	Yes	1.63	2.83	5.54	19.8	4.8M
Correlation Range	32px	2.91	4.48	10.4	28.8	4.8M
	64px	2.06	3.16	6.24	20.9	4.8M
	128px	1.64	2.81	6.00	19.9	4.8M
	All-Pairs	1.63	2.83	5.54	19.8	4.8M
Features for Refinement	Correlation	1.63	2.83	5.54	19.8	4.8M
	Warping	2.27	3.73	11.83	32.1	2.8M
Reference Model (convex upsampling), Training: 100k(C) $\rightarrow$ 100k(T)						
Upsampling	Convex	1.43	2.71	5.04	17.4	5.3M
	Bilinear	1.60	2.79	5.17	19.2	4.8M
Inference Updates	1	4.04	5.45	15.30	44.5	5.3M
	3	2.14	3.52	8.98	29.9	5.3M
	8	1.61	2.88	5.99	19.6	5.3M
	32	1.43	2.71	5.00	17.4	5.3M
	100	1.41	2.72	4.95	17.4	5.3M
	200	1.40	2.73	4.94	17.4	5.3M

Ablations

Conv GRU

단순 Convolution보다 반복적인 Flow 보정에 더 효과적이다.

Tied Weight

블럭마다 같은 가중치는 성능은 높히고 파라미터는 크게 감소시킨다.

Context Encoder

Context Encoder는 공간적 문맥 정보를 반영하여 정확도가 향상된다.

Single-Scale

RAFT에서는 단일 해상도 Feature만으로도 충분한 성능을 달성한다.

Lookup radius = 4

더 넓은 주변 Correlation을 참조할수록 정확도가 향상된다.

Correlation Pooling

Correlation Pyramid는 다양한 크기의 움직임을 효과적으로 처리한다.

Correlation Range All-Pairs

모든 픽셀 쌍의 Correlation을 사용하는 것이 가장 높은 성능을 보인다.

Update: Correlation

Warping보다 Correlation Feature를 이용한 업데이트가 훨씬 정확하다.

Convex Upsampling

학습 기반 Convex Upsampling이 Bilinear보다 더 정확한 Flow를 복원한다.

Iteration=32

반복 횟수를 늘릴수록 성능이 향상되며 약 32회 이후부터는 향상 폭이 작아진다.

Conclusion

Conclusion

Point 1

단일 고해상도 유지

작은 물체의 빠른 움직임을 놓치지 않고 세밀한 디테일을 보존할 수 있습니다.

Point 2

반복적
업데이트 연산자

100번 이상의 반복에서도 결과가 발산하지 않고 수렴하는 안정성을 보입니다.

Point 3

전체 픽셀 쌍
상관관계

4D Correlation Volume은 네트워크가 전역적인 문맥과 지역적인 세부 정보를 동시에 고려할 수 있게 합니다.

Point 4

압도적인 성능

Sintel 및 KITTI 벤치마크에서 SOTA 성능을 달성

감사합니다.