

# 얼음 파괴모드 가시화 기반 기술 개발

최종보고회의  
2019.01.29

충남대학교 메카트로닉스 공학과  
강한솔

# Contents

**1. 서론**

**2. 얼음 파괴모드 가시화**

**3. 실험 결과**

**4. 결론**

# Contents

1. 서론

2. 얼음 파괴모드 가시화

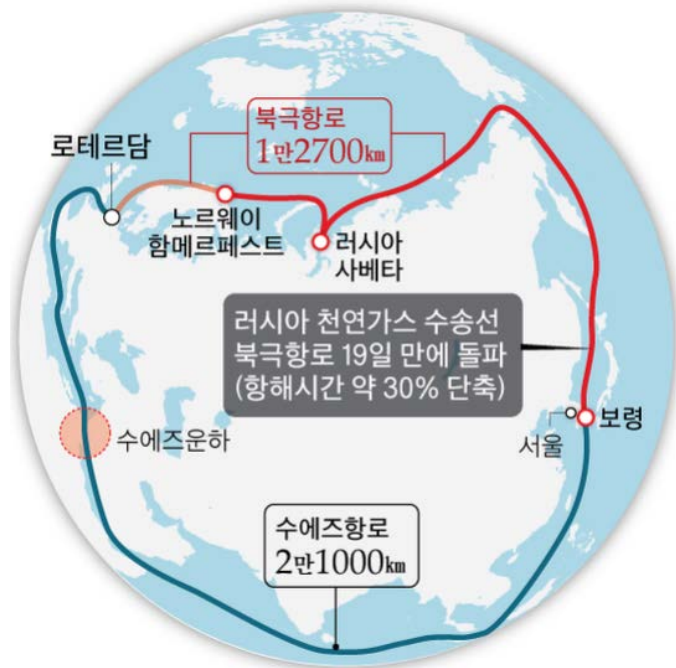
3. 실험 결과

4. 결론

1. 서론

# 1. 서론

- 최근 시간과 비용을 절약할 수 있는 북극 항로에 대한 관심이 높아짐.
- 동일한 조건으로 얼음을 생성하고, 동일 압력으로 파괴를 가하더라도 그 특성이 매번 상이함.
- 이러한 얼음 파괴 특성을 분석하기 위해 다양한 방법들이 제안되었지만, 얼음 균열을 직접적으로 시각화 하기에는 용이하지 않음.



<https://news.joins.com/article/21876796>

# Contents

1. 서론

2. 얼음 파괴모드 가시화

3. 실험 결과

4. 결론

## 2. 얼음 파괴모드 가시화

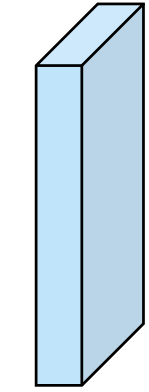
얼음의 특성 분석

얼음 크랙 전파 알고리즘

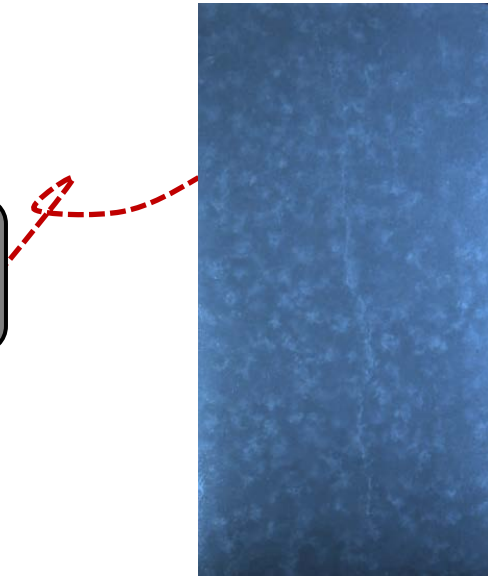
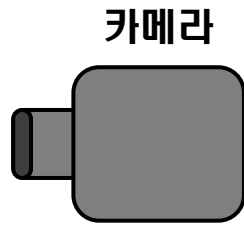
얼음 영역 식별을 위한 세그멘테이션 알고리즘

## 2. 얼음 파괴모드 가시화

### ● 얼음의 특성 분석



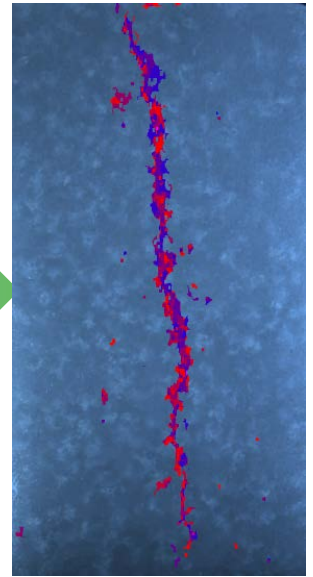
얼음 시편



획득된 영상

전처리 및 가시화 알고리즘

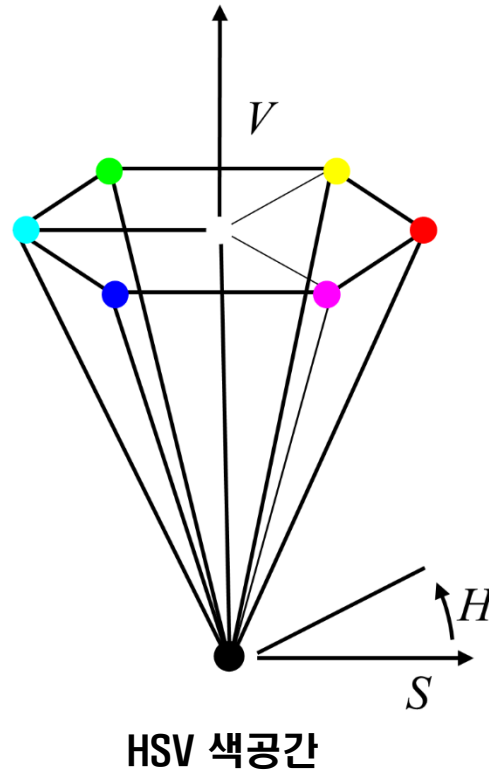
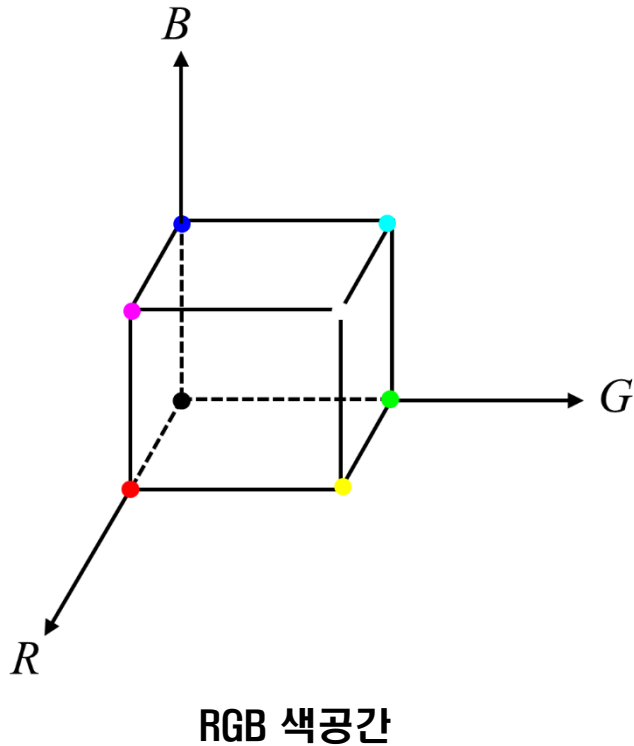
- 특징 분석에 유리한 색 공간 선정
- 최적의 조명 조건 선정



가시화 영상

## 2. 얼음 파괴모드 가시화

- 얼음의 특성 분석
  - 색상 특성 분석



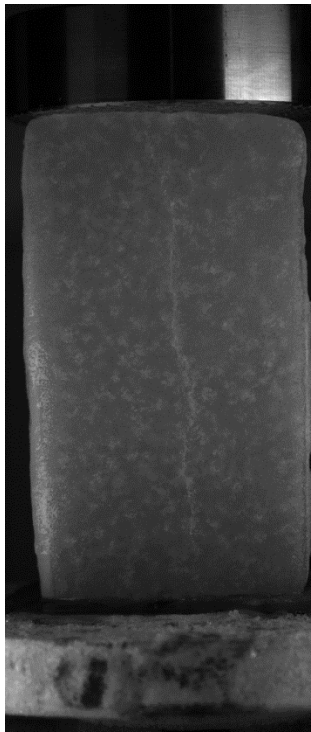
$$H = \cos^{-1} \frac{0.5[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}}$$

$$S = \frac{\text{Max}(R, G, B) - \text{Min}(R, G, B)}{\text{Max}(R, G, B)}$$

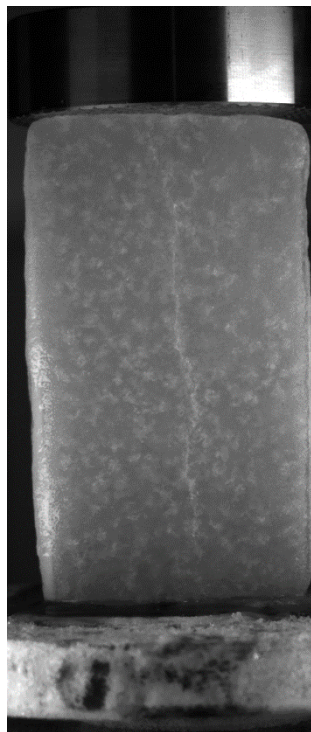
$$V = \frac{\text{Max}(R, G, B)}{255}$$

## 2. 얼음 파괴모드 가시화

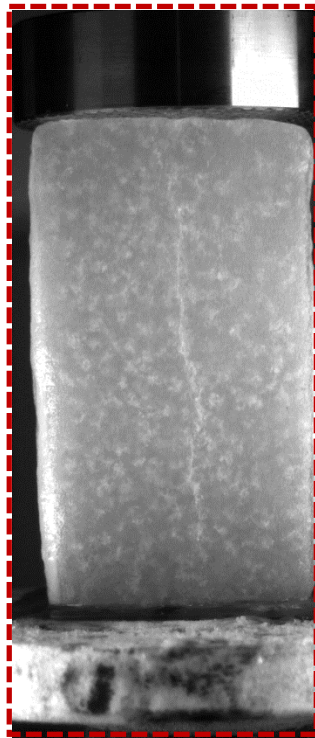
- 얼음의 특성 분석
  - 색상 특성 분석



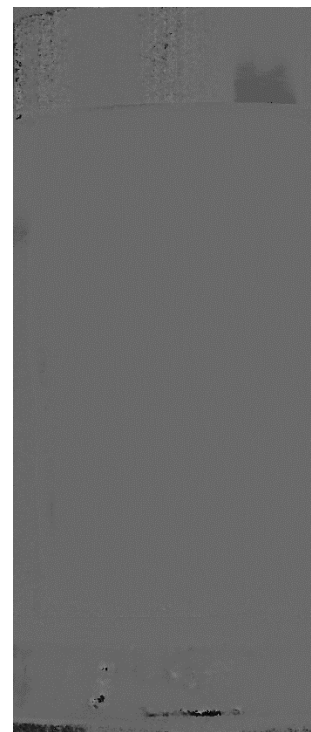
Red



Green



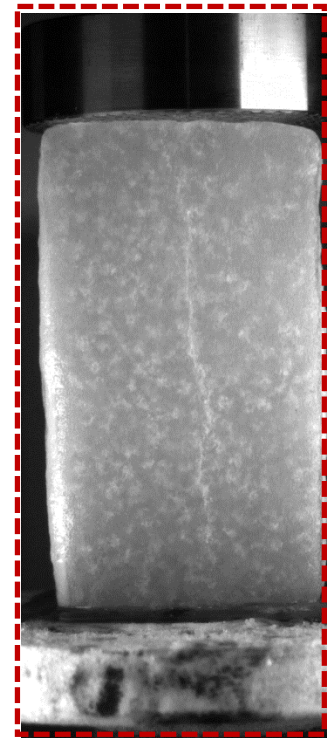
Blue



Hue



Saturation

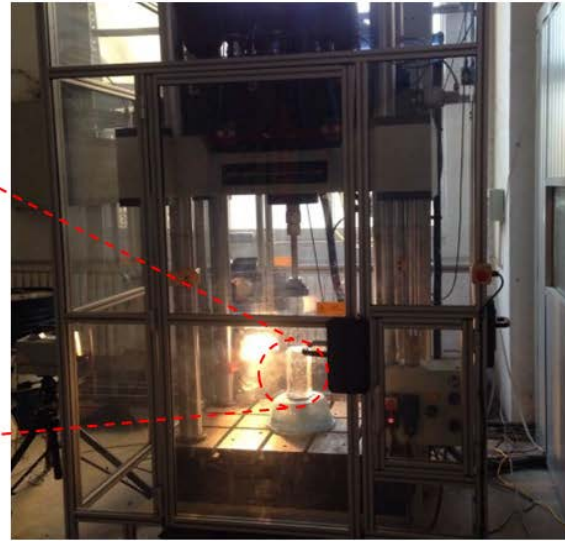
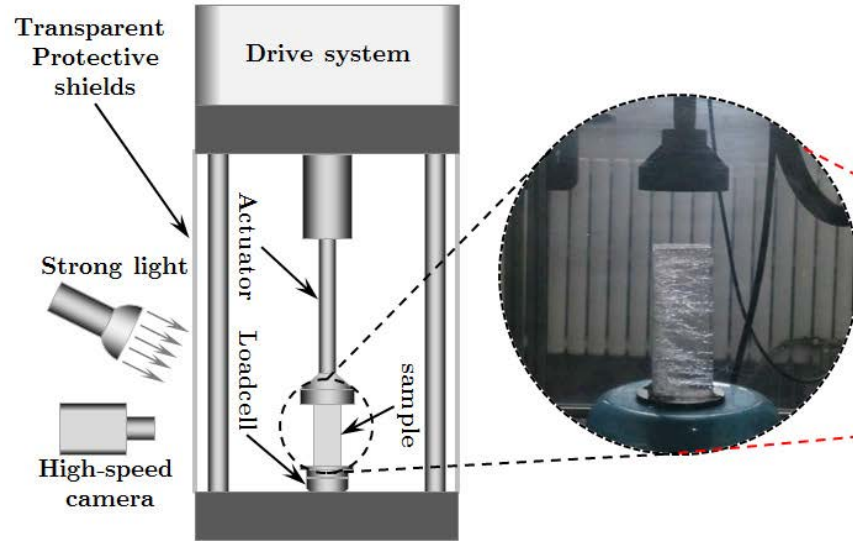


Value



## 2. 얼음 파괴모드 가시화

- 얼음의 특성 분석
  - 조명 위치 분석



기존의 조명 위치\*



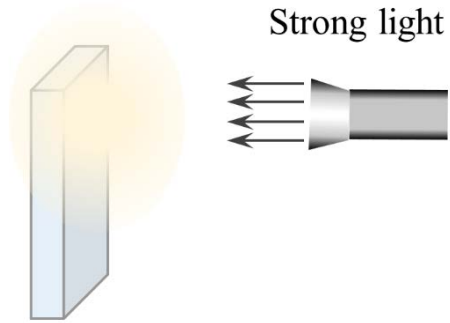
관측 결과

\* Qi, Chunfeng, et al. "Dynamic Compressive Strength and Failure of Natural Lake Ice Under Moderate Strain Rates at Near Melting Point Temperature." *Latin American Journal of Solids and Structures* 14.9 (2017): 1669-1694.

## 2. 얼음 파괴모드 가시화

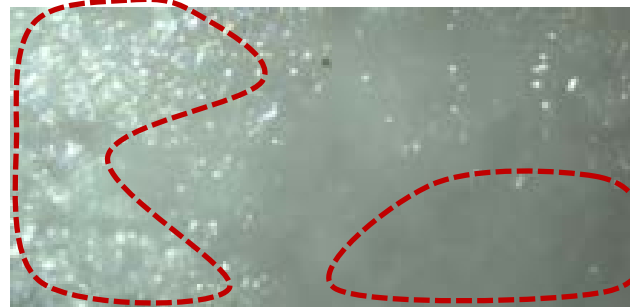
- 얼음의 특성 분석

- 조명 위치 분석

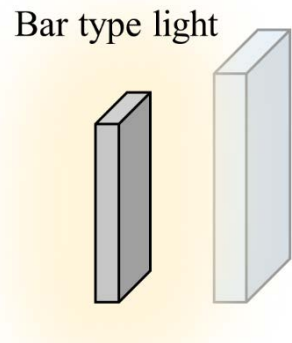


기존의 조명 위치

: 조명을 **직접적**으로 배치하여 영상의 열화가 발생함



동일한 얼음 표면에서 다양한 밝기값을 가짐.



제안하는 조명 위치

: 조명을 **간접적**으로 배치하여 영상의 열화를 줄임.

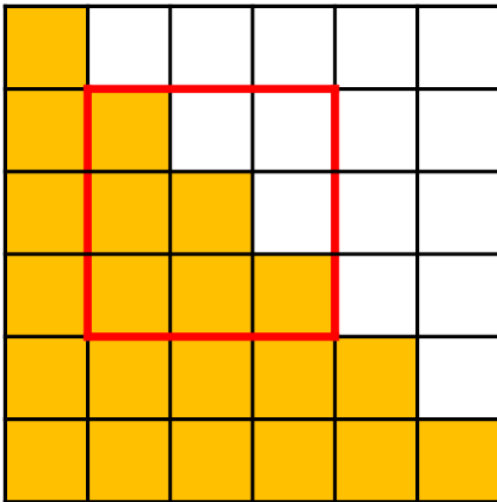


동일한 얼음 표면에서 비슷한 수준의 밝기값을 가짐.

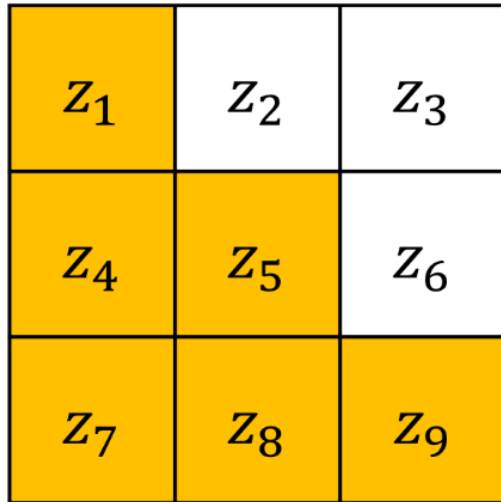
# 2. 얼음 파괴모드 가시화

- 얼음 크랙 전파 알고리즘

- 단순한 경계 검출기를 이용한 얼음 크랙 검출(소벨 연산자)



경계 영상



각 픽셀 위치

[그래디언트 정의]

$$g_x = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

$$g_y = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

[소벨 연산자] : 중심에 가중치를 둔 그래디언트

-1	0	+1
-2	0	+2
-1	0	+1

X 방향

-1	-2	-1
0	0	0
+1	+2	+1

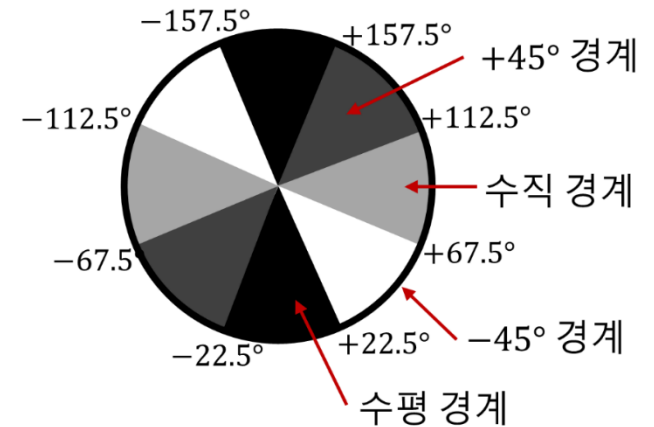
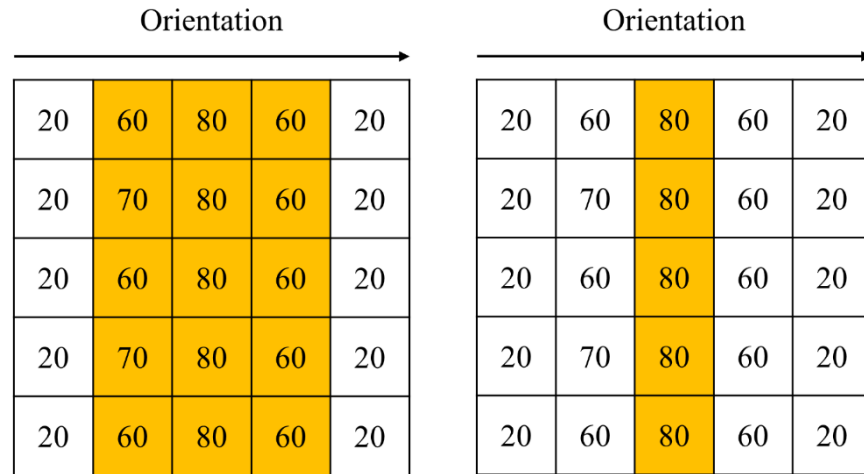
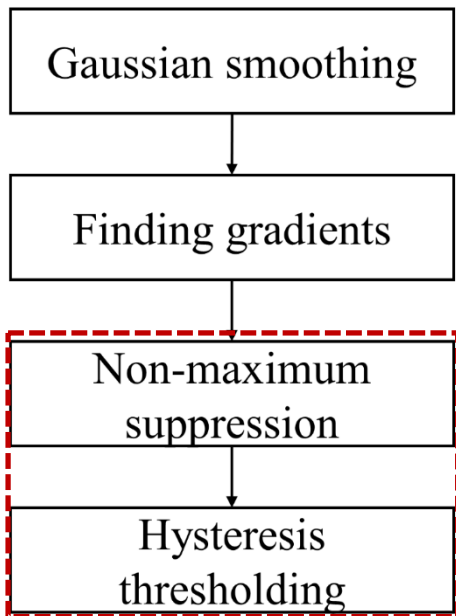
Y 방향

# 2. 얼음 파괴모드 가시화

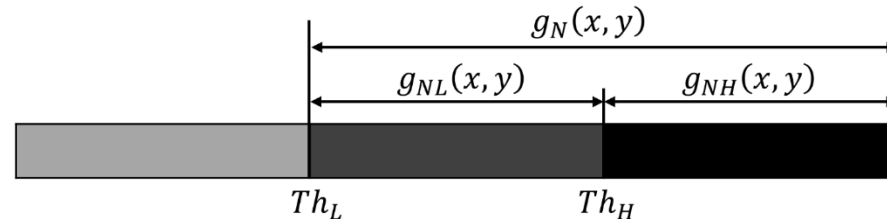
## ● 얼음 크랙 전파 알고리즘

- 단순한 경계 검출기를 이용한 얼음 크랙 검출(케니 엣지)

Non-maximum suppression : 주변과 비교하여 최대가 아닌 값은 억제함.

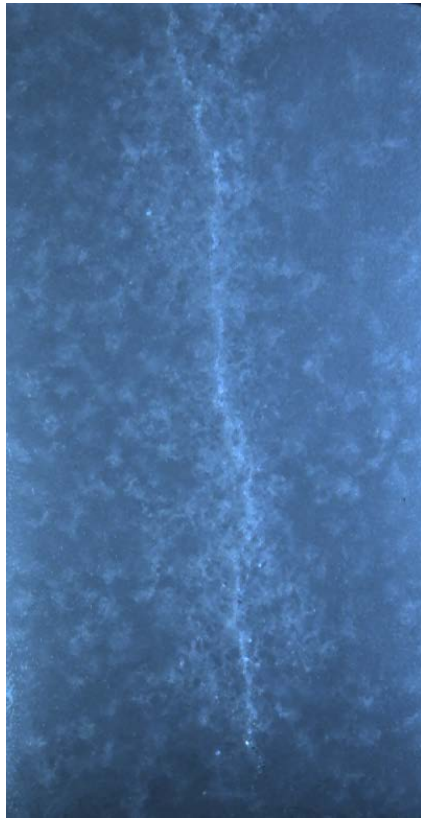


Hysteresis thresholding : 강한 경계만 살림. (단, 강한 경계와 연결되어 있는 약한 경계는 살림.)

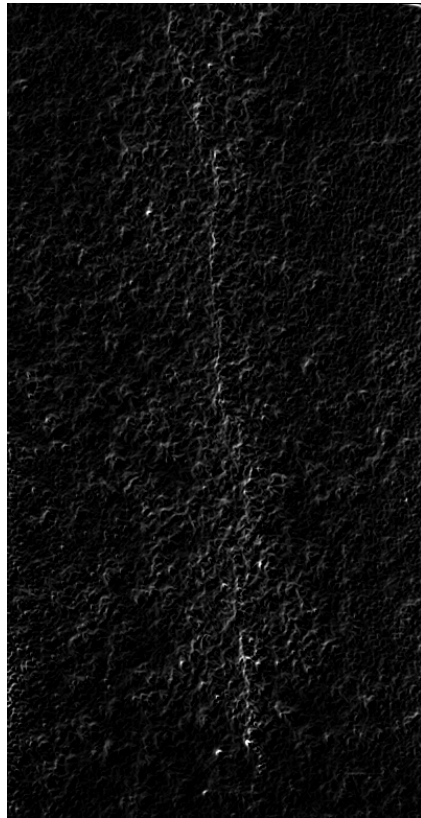


## 2. 얼음 파괴모드 가시화

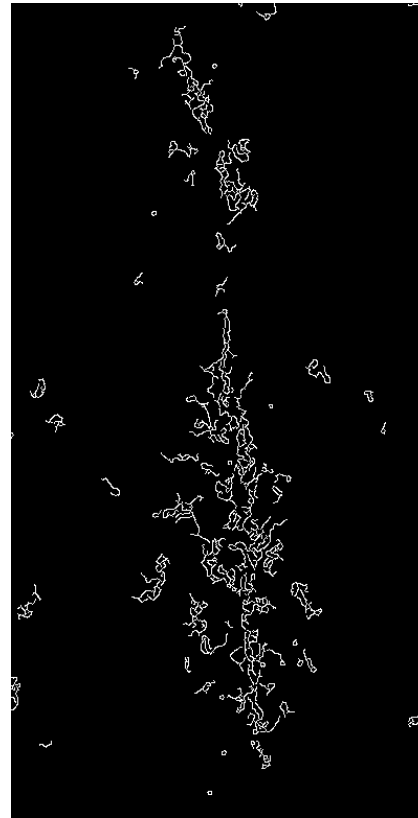
- 얼음 크랙 전파 알고리즘
  - 단순한 경계 검출기를 이용한 얼음 크랙 검출



원본 영상



소벨 결과

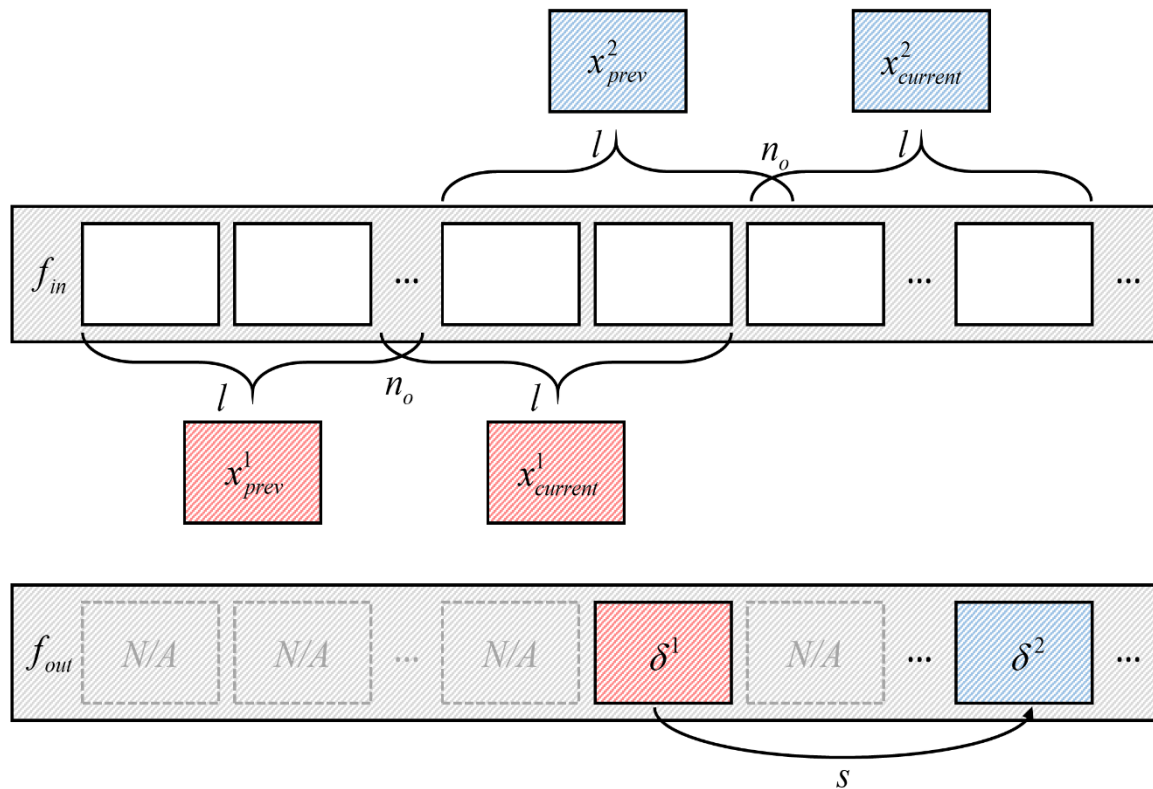


케니 엣지 결과

## 2. 얼음 파괴모드 가시화

- 얼음 크랙 전파 알고리즘

- 프레임 누적과 케니 엣지를 이용한 얼음 크랙 검출



제안하는 프레임 누적 기법

$f$  : 영상열

$x$  : 누적한 프레임으로 생성된 영상

$l$  : 누적하는 프레임 수

$n_o$  : 겹치는 프레임 수

$\delta$  : 차분 영상

$s$  : 처리하는 프레임 간격

$$FPS_{out} = \frac{FPS_{in}}{s}$$

□ : 사용자 정의 변수

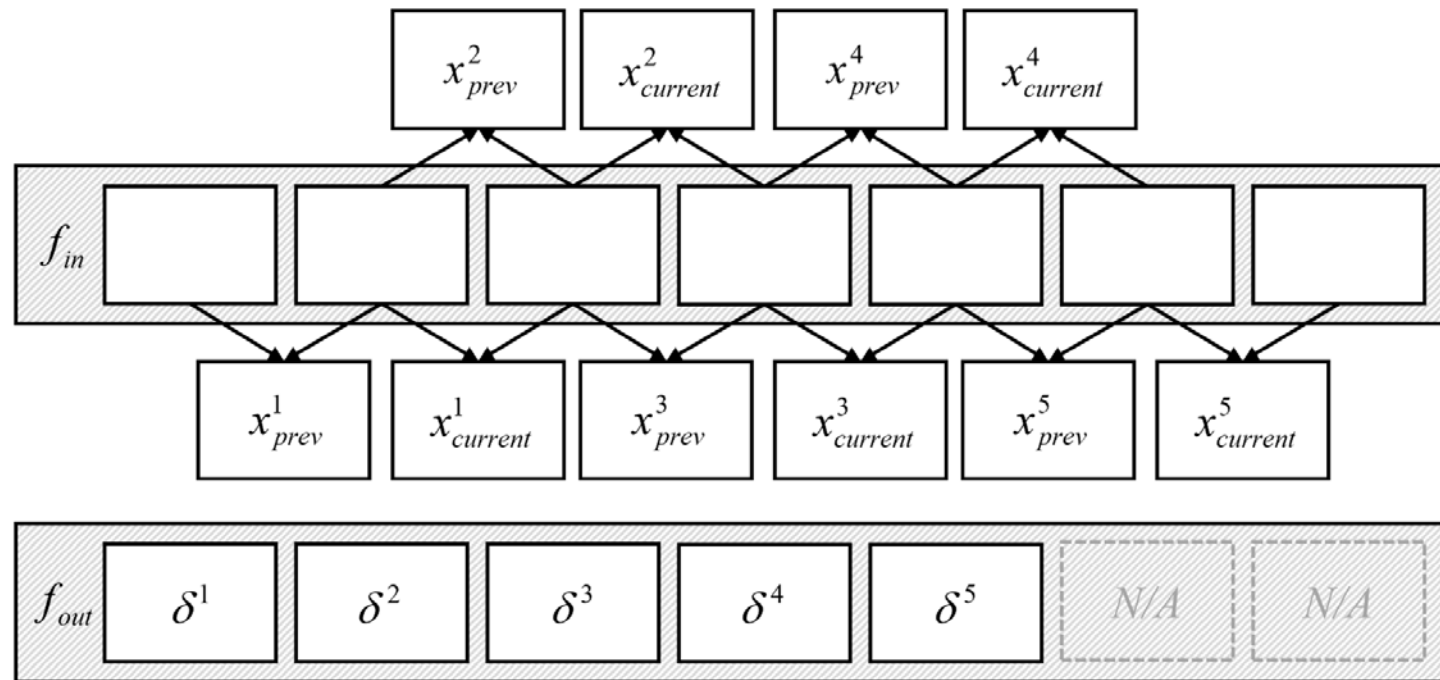


## 2. 얼음 파괴모드 가시화

### ● 얼음 크랙 전파 알고리즘

- 프레임 누적과 케니 엣지를 이용한 얼음 크랙 검출

$l = 2, n_o = 1, s = 1$  일 때



제안하는 프레임 누적 기법 예시

## 2. 얼음 파괴모드 가시화

### ● 얼음 크랙 전파 알고리즘

- 프레임 누적과 케니 엣지를 이용한 얼음 크랙 검출

---

Algorithm 1: ICP(Ice Crack Propagation), our proposed algorithm. Experiment in the paper using the default values  $s = 1, l = 5, n_o = 0, \Delta_{new} = 100, \Delta_{old} = 20, C = 2.5$

---

1 function ICP ( $f, s, l, n_o$ );

  Input : The input video sequence  $f$ , the number of frames in progress  $s$ , level of cumulative sum  $l$  (*i.e the number of frames to accumulate.*), and the number of overlapping frames between the previous frame and the current frame  $n_o$

  Output: A crack detection image.

```

2 for  $n \leq 2 \times l - n_o$  do
3   if  $n < 2 \times l - n_o$  then
4     if  $n < l$  then
5       |  $x_{prev} += f_n$ 
6     end
7     if  $n \geq l - n_o$  then
8       |  $x_{current} += f_n$ 
9     end

```

```

10   else
11     |  $x_{diff} = x_{current} - x_{prev}$ 
12     |  $x_{canny}(i, j) = Canny(x_{diff})$ 
13     if  $x_{canny}(i, j) > 0$  then
14       |  $Map(i, j) += \Delta_{new}$ 
15     else if  $x_{canny}(i, j) == 0$  and  $Map(i, j) > 0$  then
16       |  $Map(i, j) -= \Delta_{old}$ 
17     else
18       |  $Map(i, j) = 0$ 
19     end
20     if  $Map(i, j) > 0$  then
21       |  $x_{out}^R(i, j) = C \times Map(i, j)$ 
22       |  $x_{out}^G(i, j) = 0$ 
23       |  $x_{out}^B(i, j) = C \times (\Delta_{new} - Map(i, j))$ 
24     end
25   end
26 endfor

```

---

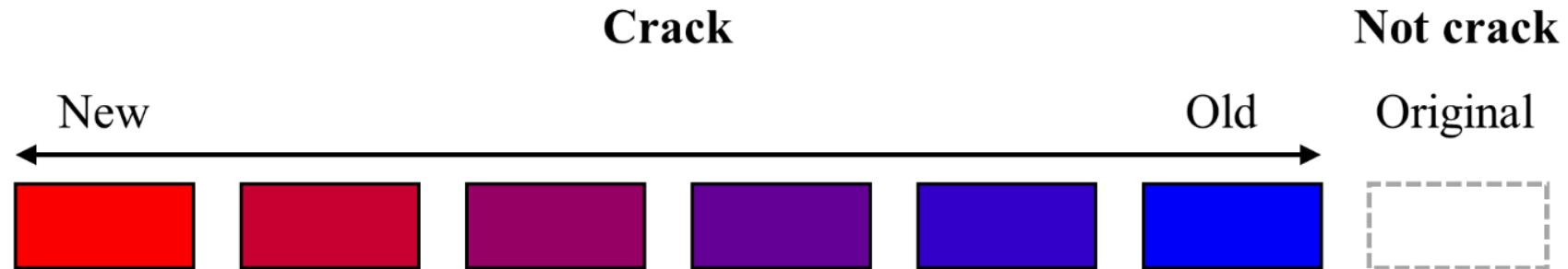


## 2. 얼음 파괴모드 가시화

- 얼음 크랙 전파 알고리즘

- 프레임 누적과 케니 엣지를 이용한 얼음 크랙 검출

$\Delta_{new} = 100, \Delta_{old} = 20, C = 2.5$  일 때



## 2. 얼음 파괴모드 가시화

### ● 얼음 크랙 전파 알고리즘

- 프레임 누적과 LoG를 이용한 얼음 크랙 검출

---

**Algorithm 2:** ICP2(Ice Crack Propagation using LoG), our proposed algorithm. Experiment in the paper using the default values  $s = 1, l = 5, n_o = 0, \Delta_{new} = 100, \Delta_{old} = 20, C = 2.5$

---

```

1 function ICP2 ( $f, s, l, n_o$ );
  Input  : The input video sequence  $f$ , the number of frames in progress
            $s$ , level of cumulative sum  $l$  (i.e the number of frames to
           accumulate.), and the number of overlapping frames between
           the previous frame and the current frame  $n_o$ 

  Output: A crack detection image.
2 for  $n \leq 2 \times l - n_o$  do
3   if  $n < 2 \times l - n_o$  then
4     if  $n < l$  then
5        $x_{prev+} = f_n$ 
6     end
7     if  $n \geq l - n_o$  then
8        $x_{current+} = f_n$ 
9     end

```

```

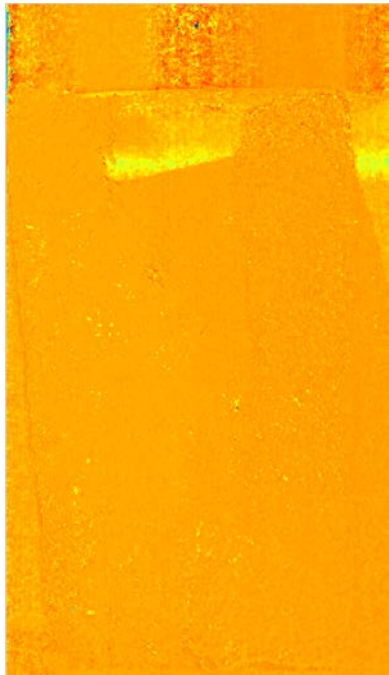
10   else
11      $x_{current}^{LoG} = LoG(x_{current})$  ▷ Different point 1
12      $x_{prev}^{LoG} = LoG(x_{prev})$  ▷ Different point 2
13      $x_{diff}(i, j) = x_{current}^{LoG} - x_{prev}^{LoG}$  ▷ Different point 3
14     if  $x_{canny}(i, j) > 0$  then
15        $Map(i, j) + = \Delta_{new}$ 
16     else if  $x_{canny}(i, j) == 0$  and  $Map(i, j) > 0$  then
17        $Map(i, j) - = \Delta_{old}$ 
18     else
19        $Map(i, j) = 0$ 
20     end
21     if  $Map(i, j) > 0$  then
22        $x_{out}^R(i, j) = C \times Map(i, j)$ 
23        $x_{out}^G(i, j) = 0$ 
24        $x_{out}^B(i, j) = C \times (\Delta_{new} - Map(i, j))$ 
25     end
26   end
27 endfor

```

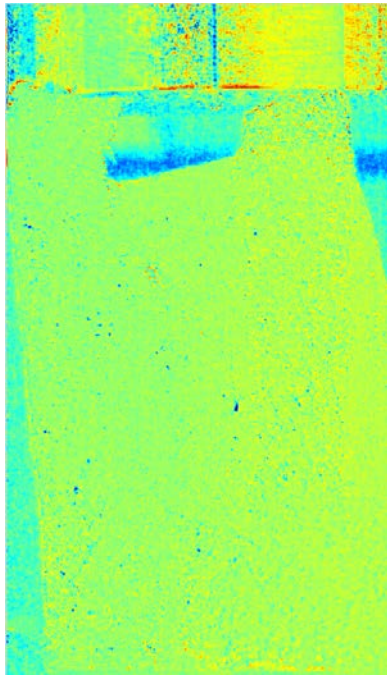
## 2. 얼음 파괴모드 가시화

- 얼음 영역 식별을 위한 세그멘테이션 알고리즘
  - 통계적 특성 기반의 세그멘테이션

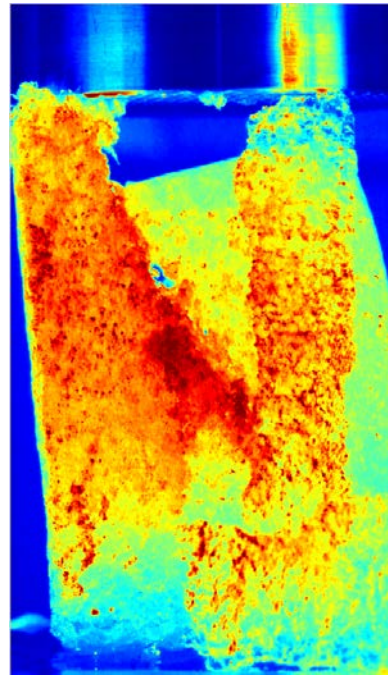
[파괴 전]



Hue

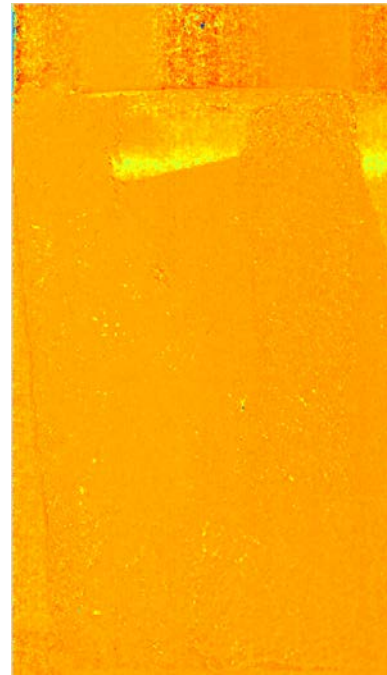


Saturation

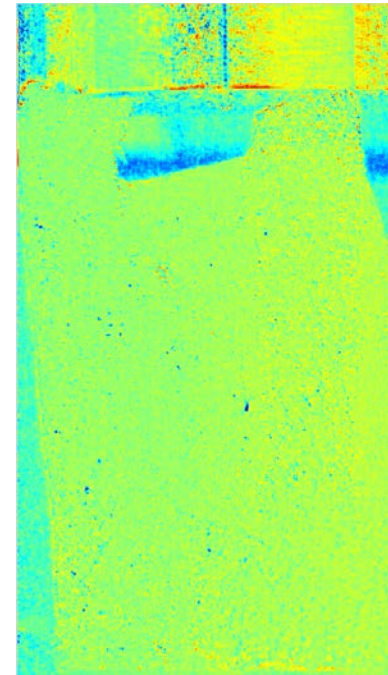


Value

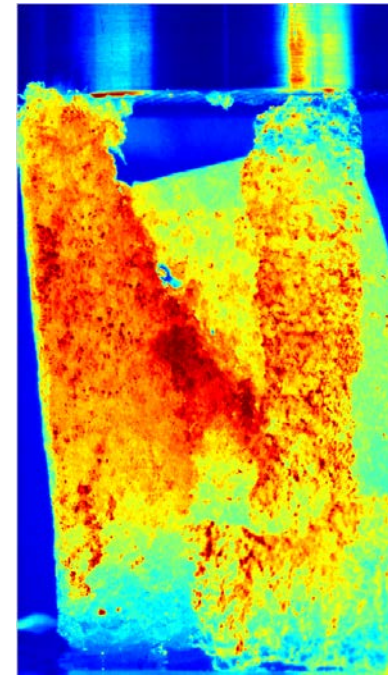
[파괴 후]



Hue



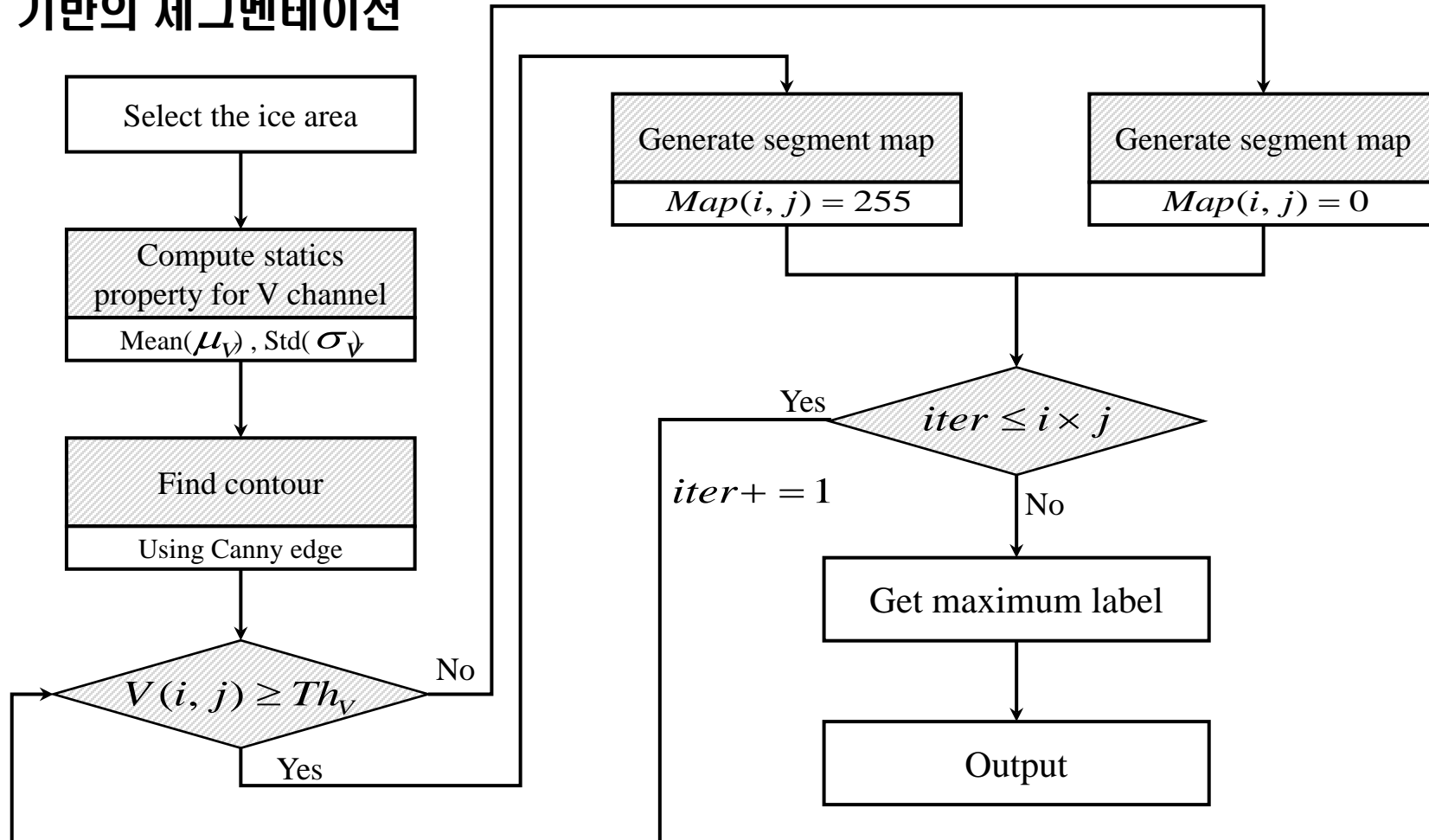
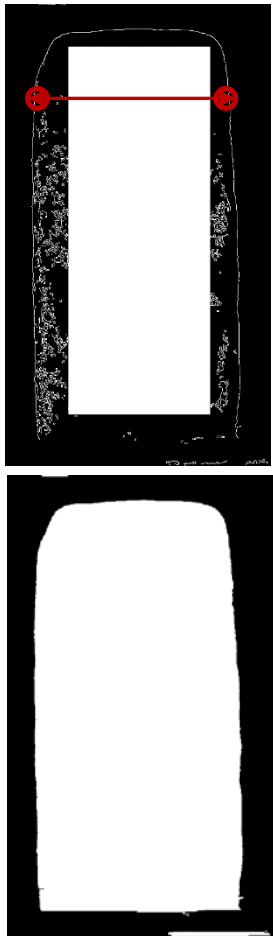
Saturation



Value

# 2. 얼음 파괴모드 가시화

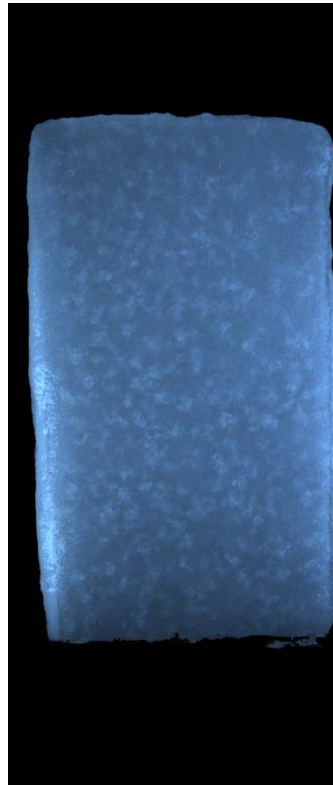
- 얼음 영역 식별을 위한 세그멘테이션 알고리즘
  - 통계적 특성 기반의 세그멘테이션



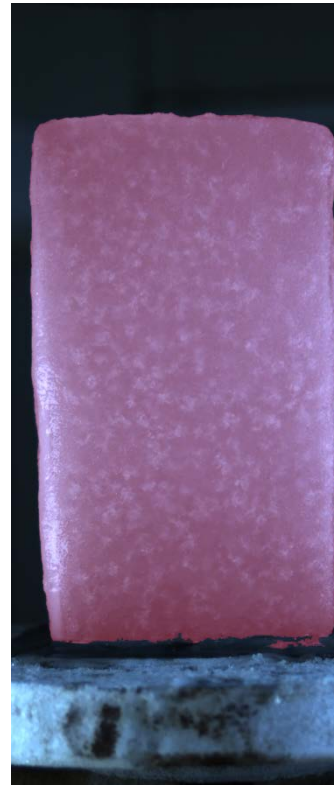
## 2. 얼음 파괴모드 가시화

- 얼음 영역 식별을 위한 세그멘테이션 알고리즘
  - 영역 오검출 제거 기법

[파괴 전]

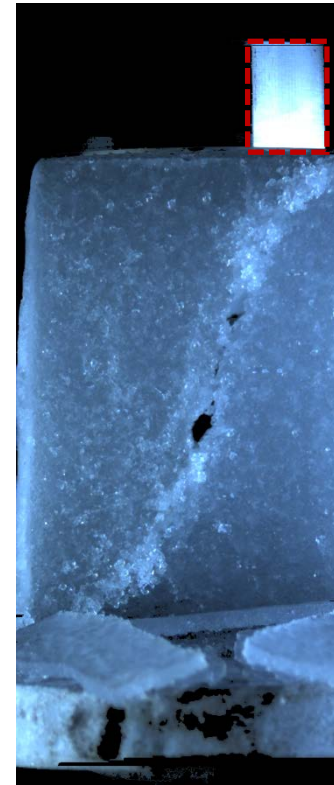


세그멘테이션

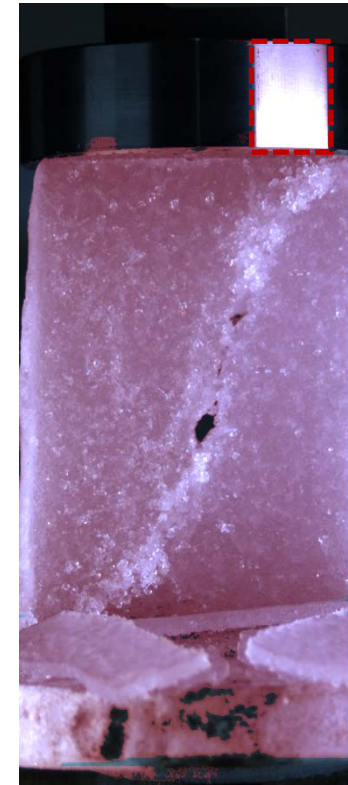


세그멘테이션(오버레이)

[파괴 후]



세그멘테이션

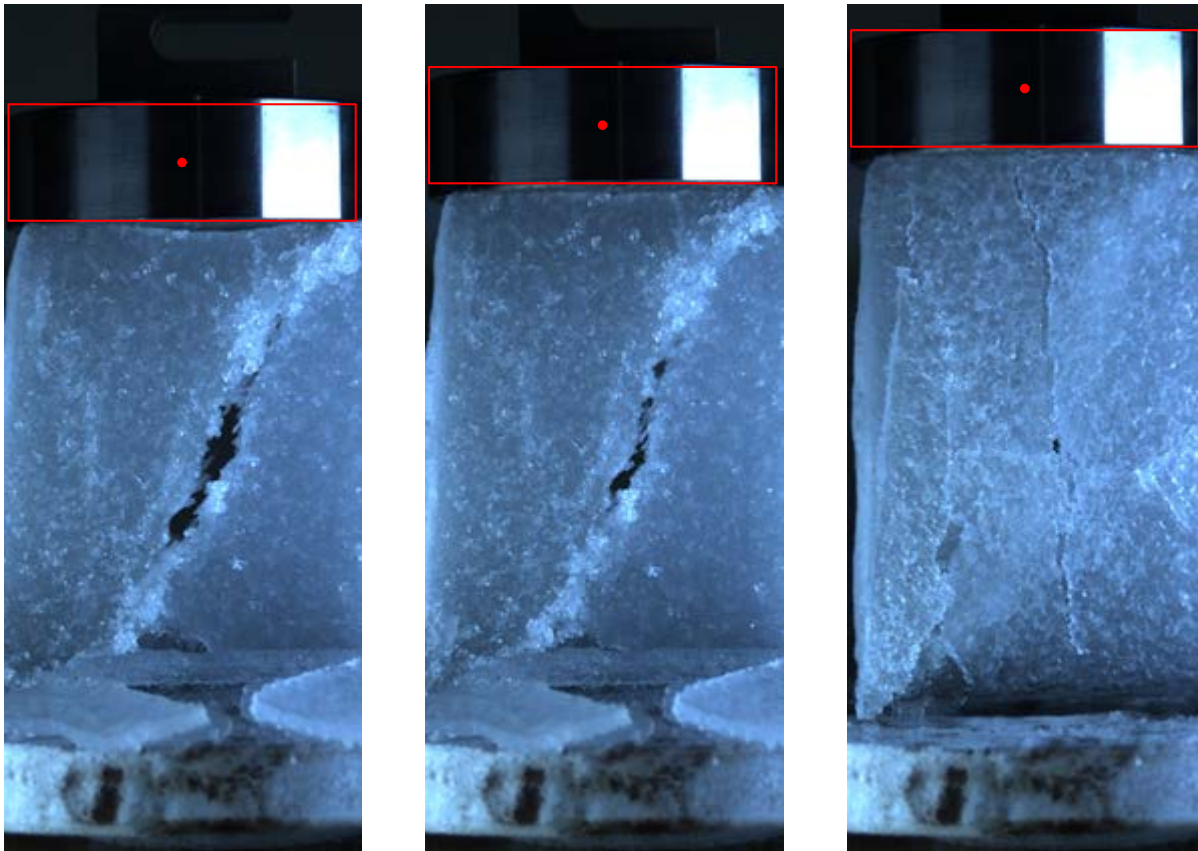


세그멘테이션(오버레이)

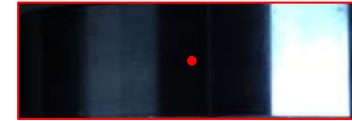


## 2. 얼음 파괴모드 가시화

- 얼음 영역 식별을 위한 세그멘테이션 알고리즘
  - 영역 오검출 제거 기법



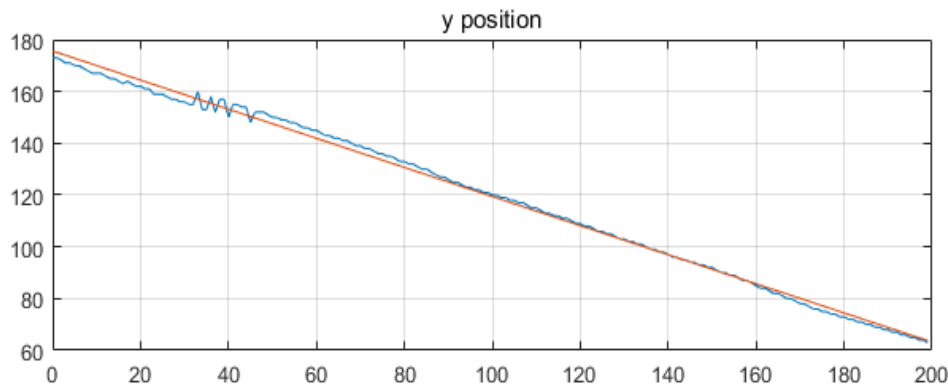
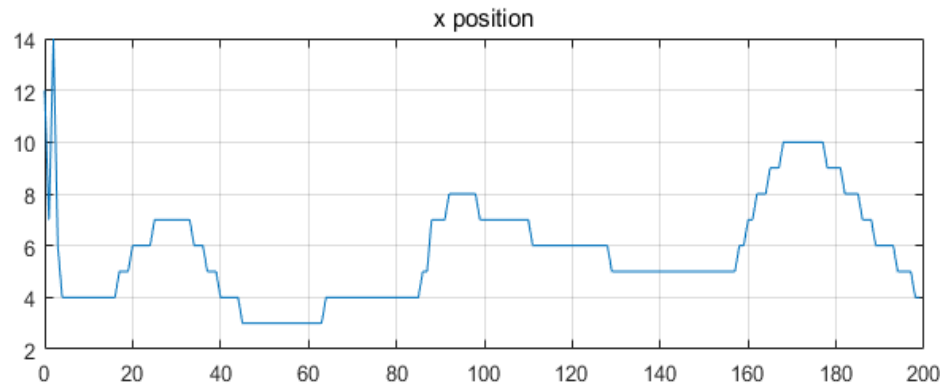
: 영상 역재생을 통한 금속 구조물 위치 추정



$$NCC = \frac{1}{N} \sum_{i,j} \frac{(I_1(i, j) - m_1)(I_2(i, j) - m_2)}{\sigma_1 \sigma_2}$$

# 2. 얼음 파괴모드 가시화

- 얼음 영역 식별을 위한 세그멘테이션 알고리즘
  - 영역 오검출 제거 기법



: LMS를 통한 최적의 금속 구조물 위치 변화량 파악

금속 구조물의 위치 추정

# Contents

1. 서론

2. 얼음 파괴모드 가시화

**3. 실험 결과**

4. 결론

## **3. 실험 결과**

실험 환경 구성도

크랙 가시화 결과

세그멘테이션 결과



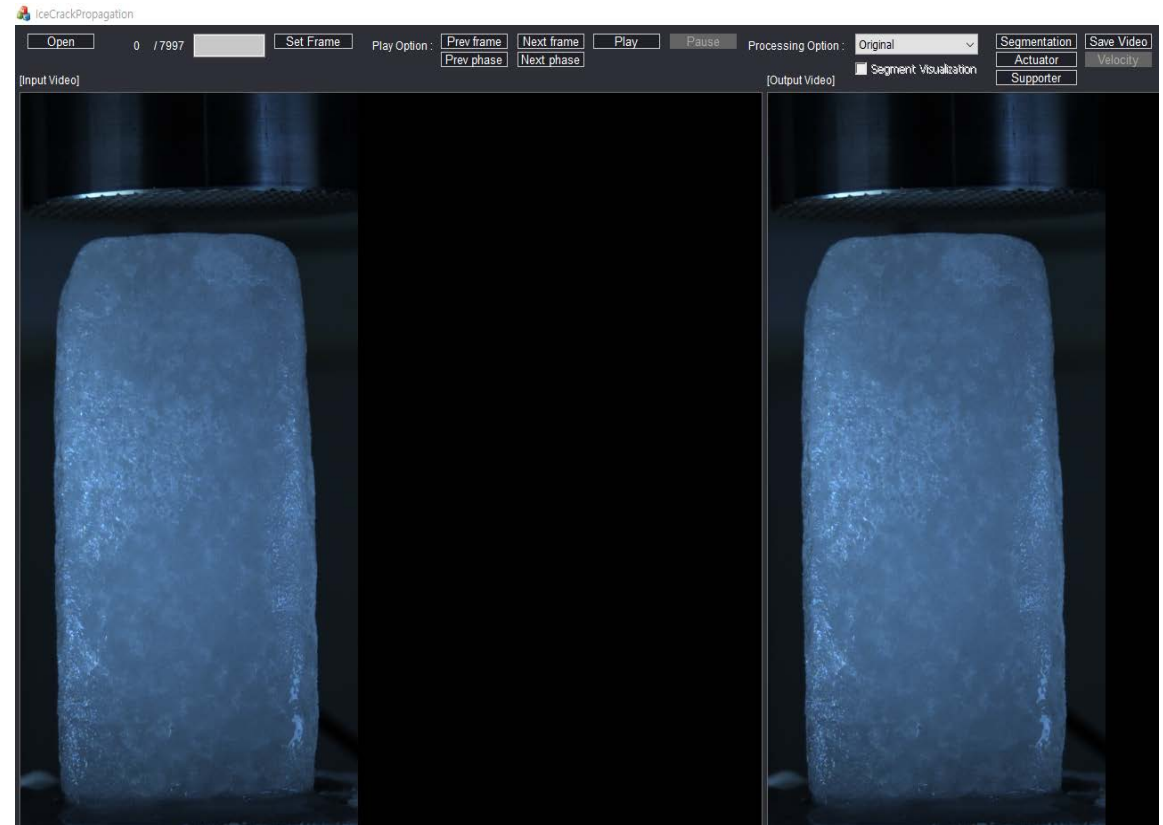
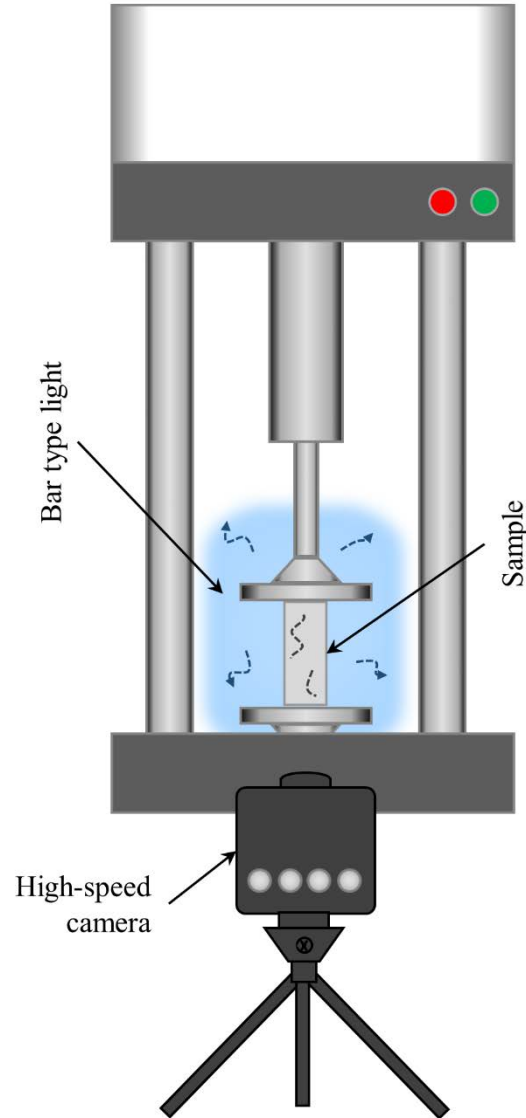
# 3. 실험 결과

## ● 실험 환경 구성도



Photron FASTCAM Mini WX50

압력 : 150kPa  
셔터 스피드 : 1/1000초

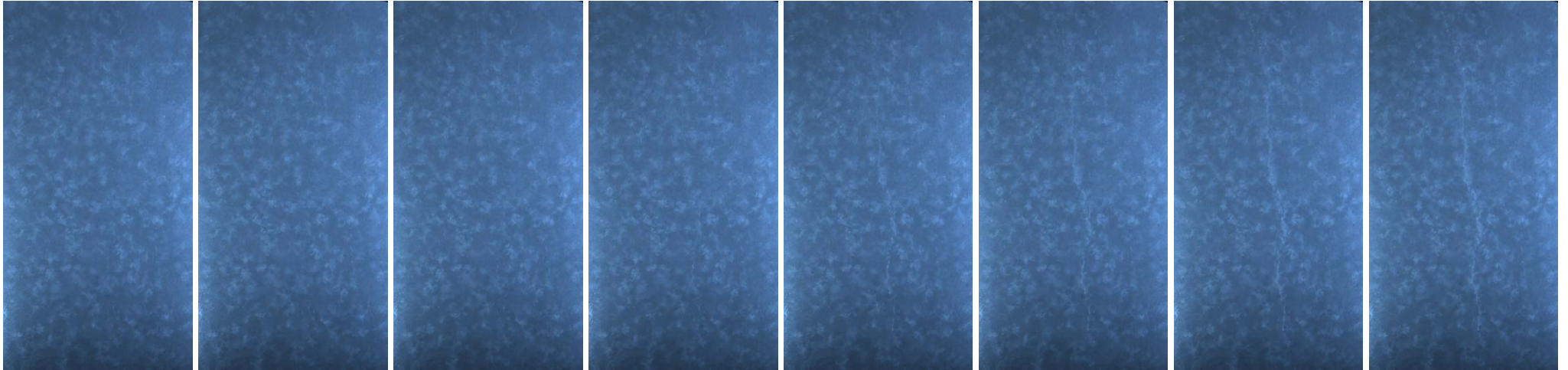


영상 해석 소프트웨어

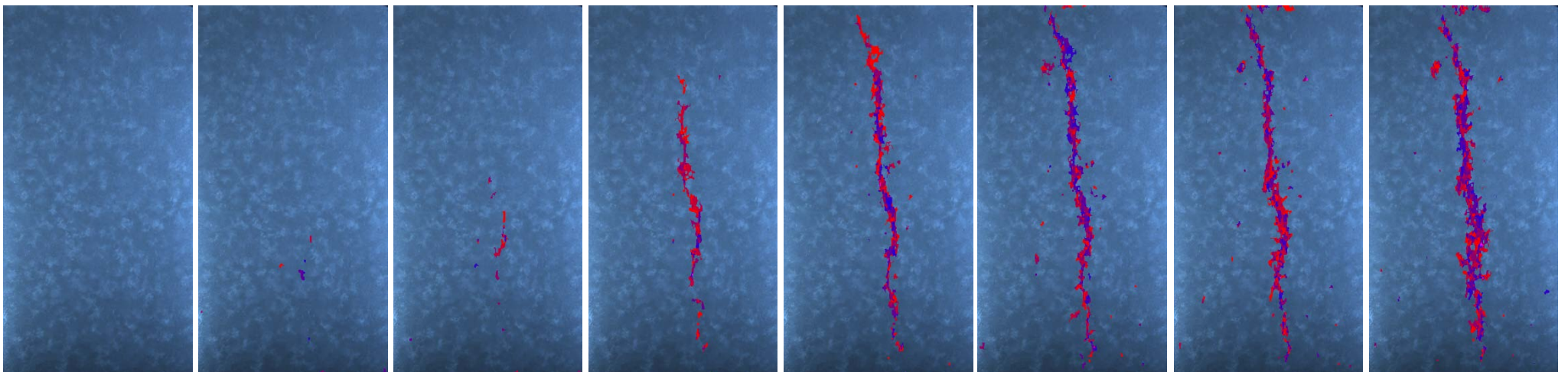
# 3. 실험 결과

## ● 크랙 가시화 결과

[원본 영상]



[프레임 누적 + 케니엣지]

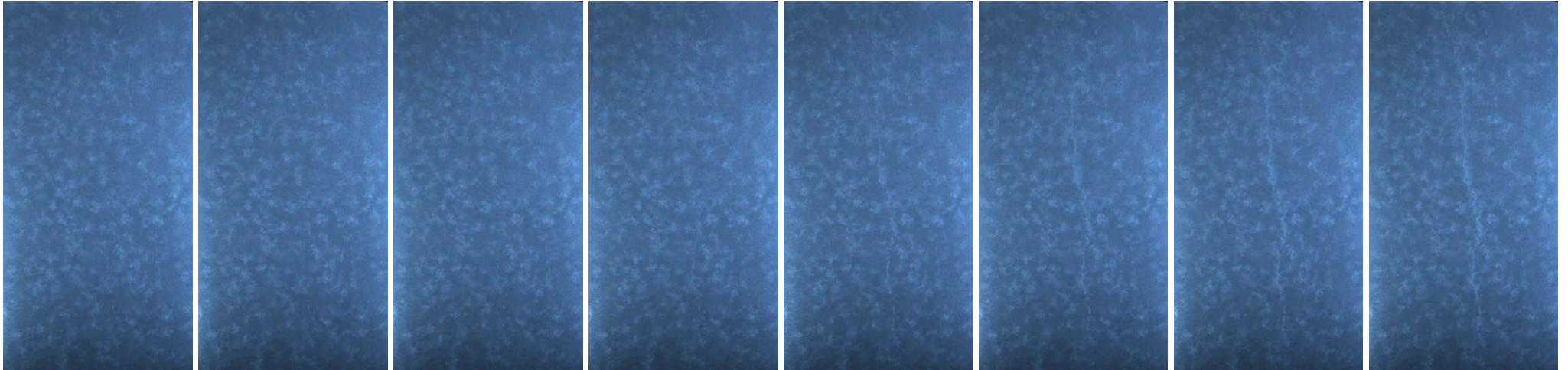




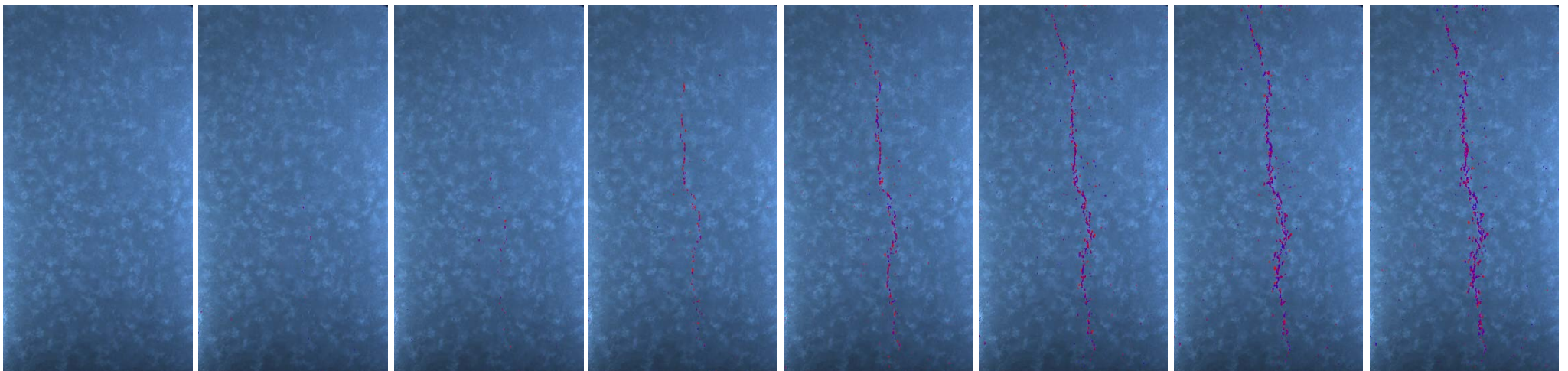
# 3. 실험 결과

## ● 크랙 가시화 결과

[원본 영상]



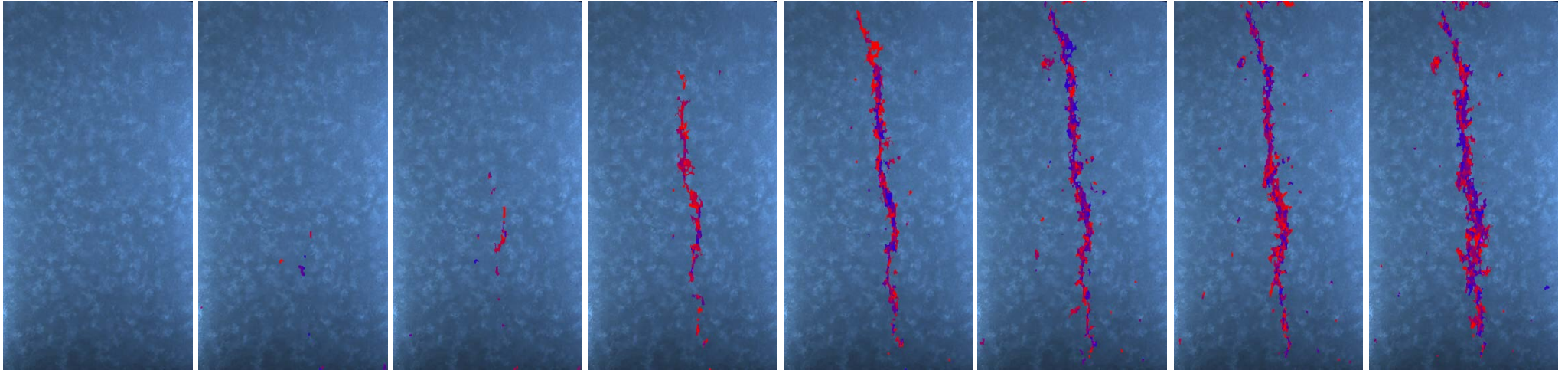
[프레임 누적 + LoG]



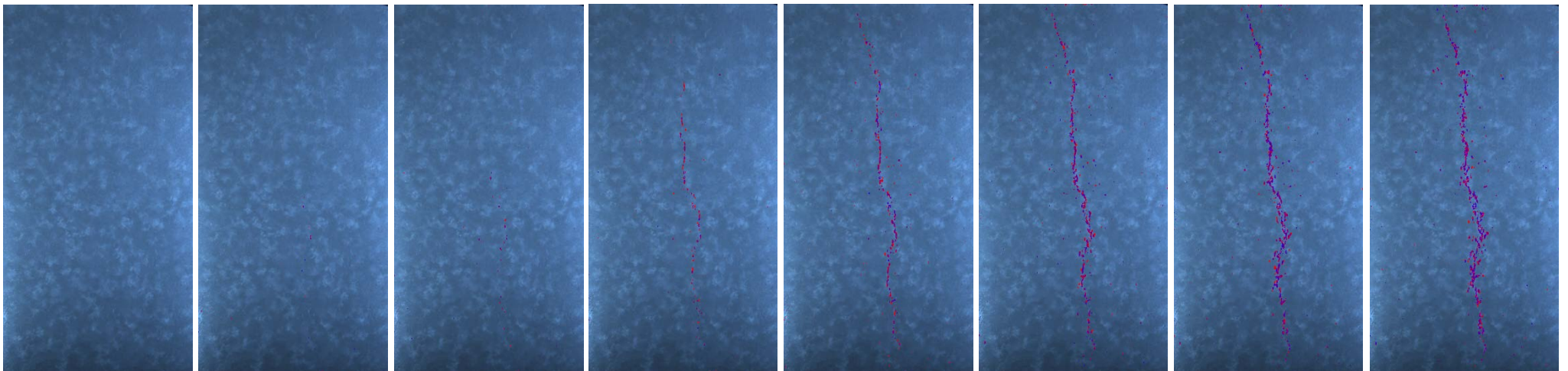
# 3. 실험 결과

## ● 크랙 가시화 결과

[프레임 누적 + 케니엣지]



[프레임 누적 + LoG]

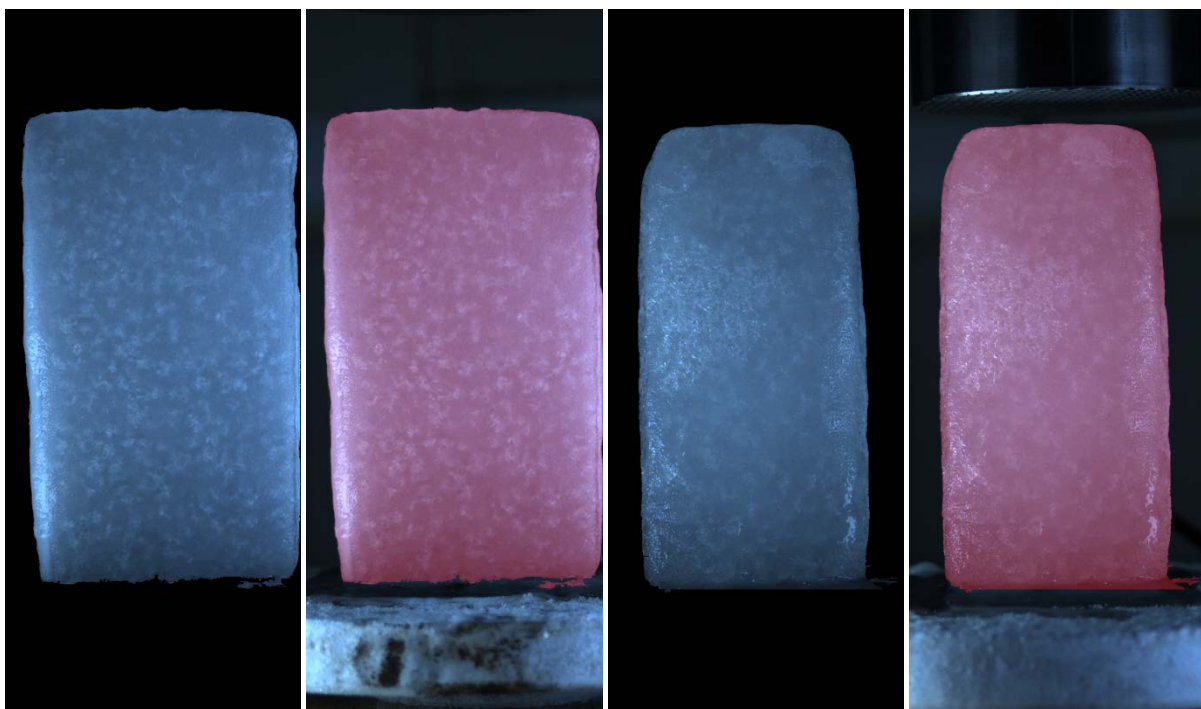




# 3. 실험 결과

## ● 세그멘테이션 결과

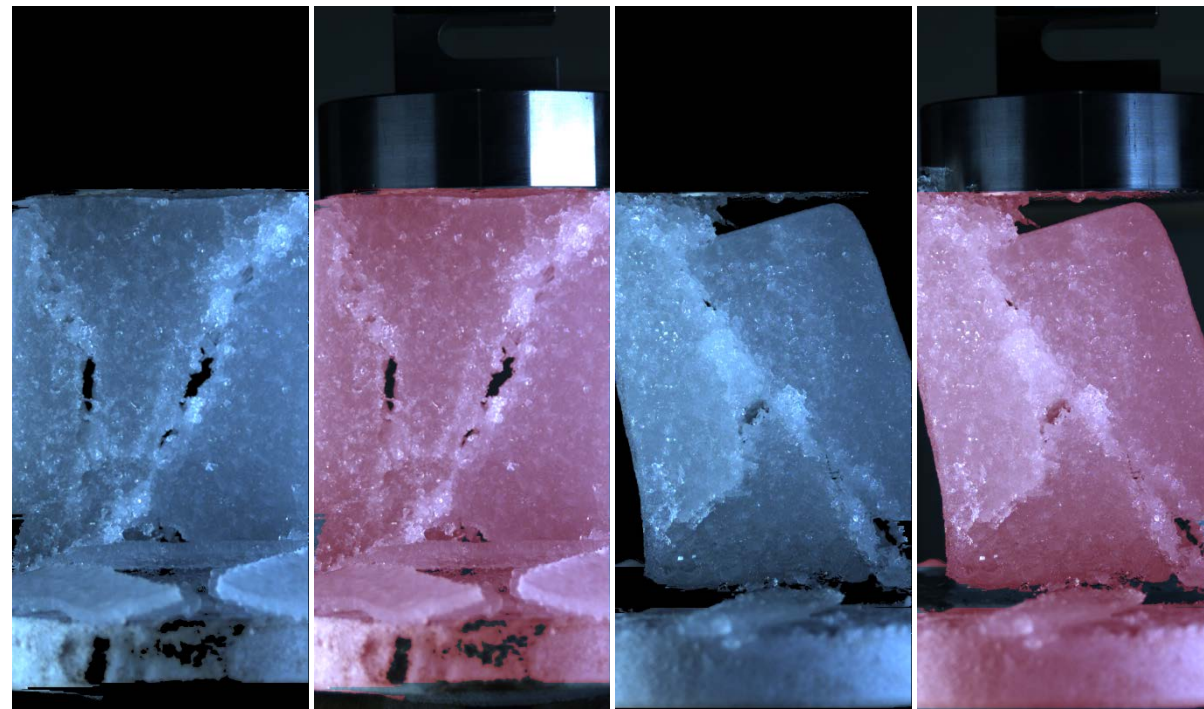
[파괴 전]



1번 영상

2번 영상

[파괴 후]



1번 영상

2번 영상

# Contents

1. 서론

2. 얼음 파괴모드 가시화

3. 실험 결과

4. 결론

4. 결론

## 4. 결론

- ▶▶ 얼음 파괴모드 분석을 위한 얼음 내 발생하는 크랙에 대한 가시화 및 세그멘테이션 기법을 개발하고 이를 해석 가능한 소프트웨어 형태로 구현함.
- ▶▶ 원본 영상과 제안하는 방법의 결과들을 비교하여, 크랙에 대한 시인성이 좋아진 것을 확인함.
- ▶▶ 원본 영상에 세그멘테이션 영역을 오버레이 시켜 얼음 영역이 적절히 검출되는 것을 확인함.
- ▶▶ 추후 얼음 파괴모드 분석에 직접적으로 적용되어 크랙 가시화 이외에 다양한 얼음 파괴모드 분석을 위해 활용될 수 있을 것으로 예상함.

# Future Work

## GAN Research



- ~~Vanilla GAN~~
- ~~DCGAN~~
- ~~InfoGAN~~
- LS GAN
- BEGAN
- Pix2Pix
- Cycle GAN
- Novel GAN(about depth)

## Tools



- ~~Document~~
- ~~Programming~~
- ~~PyTorch~~
- Python executable & UI

## I Know What You Did Last Faculty



- ~~C++ Coding Standard~~
- Mathematical theory
- LSM applications

## Other Research



- ~~Level Processor~~
- ~~Ice Propagation~~



Q

&

A

*Thank you for your attention*

# Appendix

- Non-maximum suppression

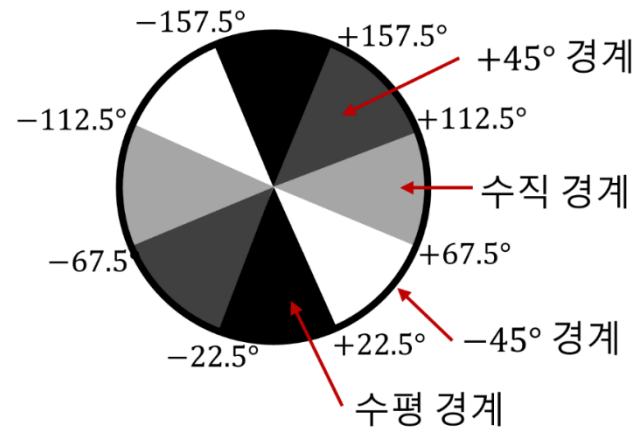
- i)  $\alpha(x, y)$ 에 가장 가까운 방향에서  $d_k$  를 찾음.
- ii) 기준이 되는  $M(x, y)$ 의 값과  $d_k$  방향에 주변 두 이웃들의 값과 비교함.
- iii) 두 이웃들의 값들 중 적어도 하나보다 작으면,  $g_N(x, y) = 0$ 로 함.
- iv) 아닌 경우에는  $g_N(x, y) = M(x, y)$ 로 함.

$$\alpha(x, y) = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

$$M(x, y) = \sqrt{g_x^2 + g_y^2}$$

Orientation →				
20	60	80	60	20
20	70	80	60	20
20	60	80	60	20
20	70	80	60	20
20	60	80	60	20

Orientation →				
20	60	80	60	20
20	70	80	60	20
20	60	80	60	20
20	70	80	60	20
20	60	80	60	20



# Appendix

- Hysteresis thresholding

- i)  $g_{NH}(x, y)$  의 조건을 만족하는 픽셀  $p(x, y)$  를 찾음.
- ii)  $p(x, y)$  의 8 이웃으로 연결된  $g_{NL}(x, y)$  의 모든 경계를 유효 경계로 선정함.
- iii)  $g_{NH}(x, y)$  의 조건을 만족하는 모든 픽셀을 방문하였는지 확인함.
- iv) iii)의 조건을 만족하지 않은 i)로 돌아감.
- v) iii)의 조건을 만족하면 유효 경계로 설정되지 않은  $g_{NL}(x, y)$  의 모든 경계를 0으로 함.
- vi) 최종적인 출력은  $g_{NH}(x, y)$  과  $g_{NL}(x, y)$  를 합쳐서 출력함.

