FPCV 스터디 4주차

곽나영

2024.09.24.

Image Processing I

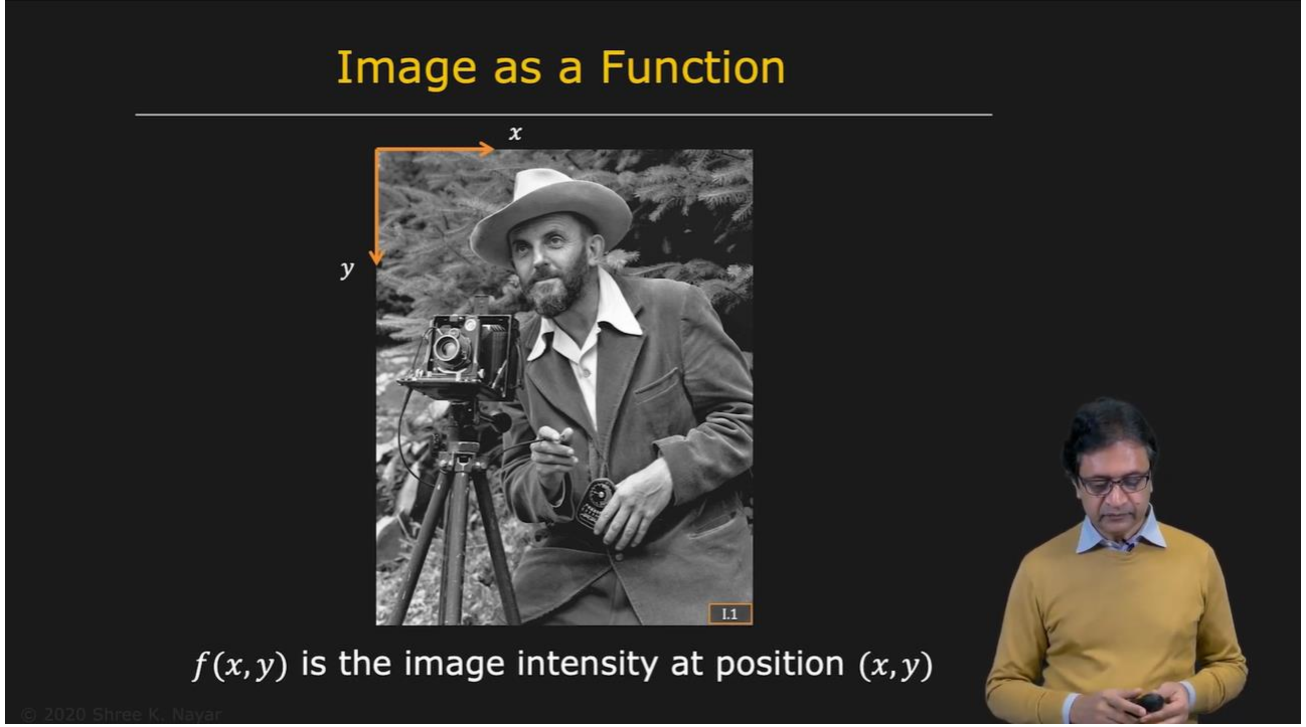
Overview

야간 이미지, 모션 블러나 초점 흐림과 같은 다양한 이미지 노이즈들을 이미지 프로세싱으로 제거할 수 있다.

1. 픽셀 프로세싱

Image Processing I

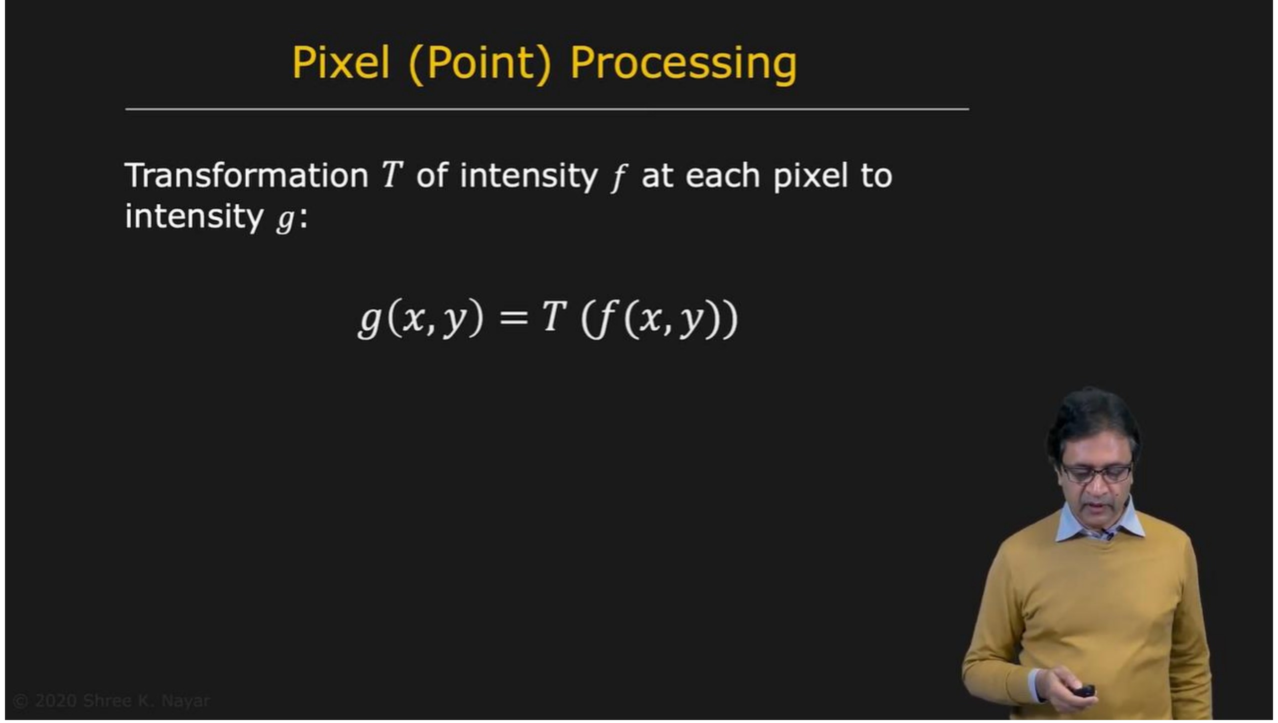
Pixel Processing



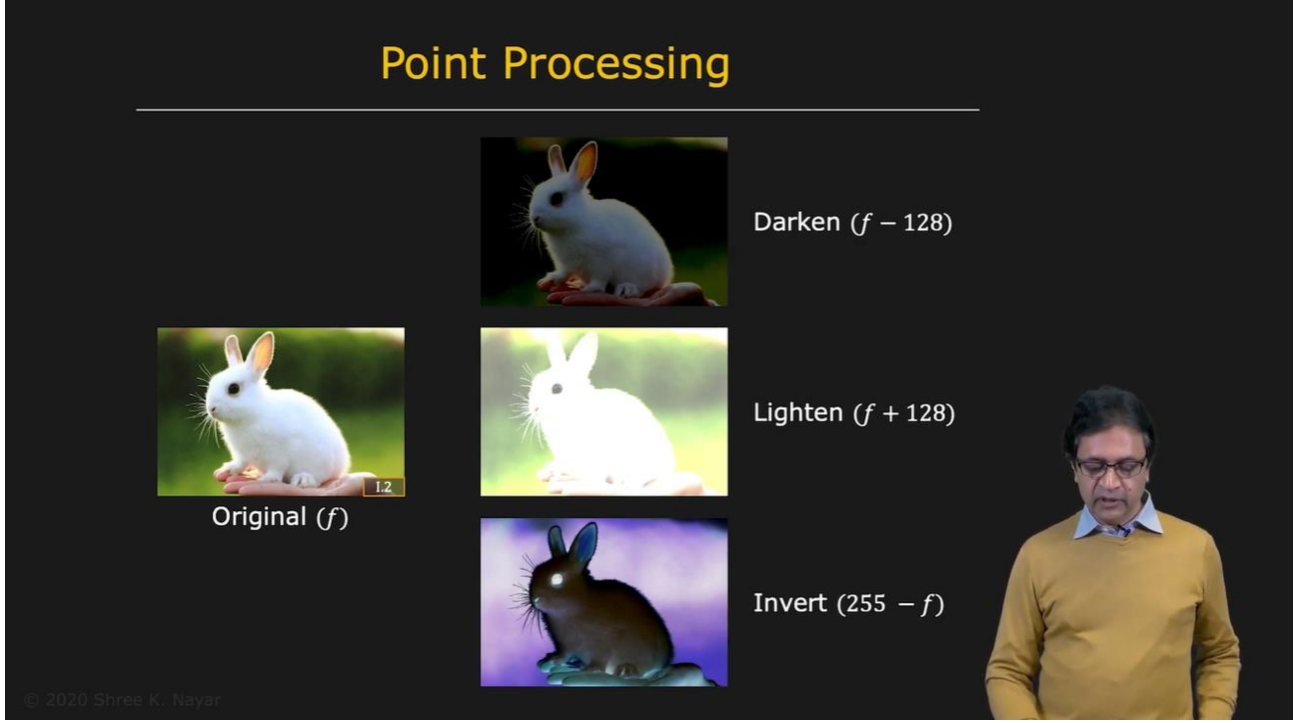
- 이미지를 함수로 정의

- 공간좌표 x, y가 주어지고 밝기 값인 f를 갖는다.

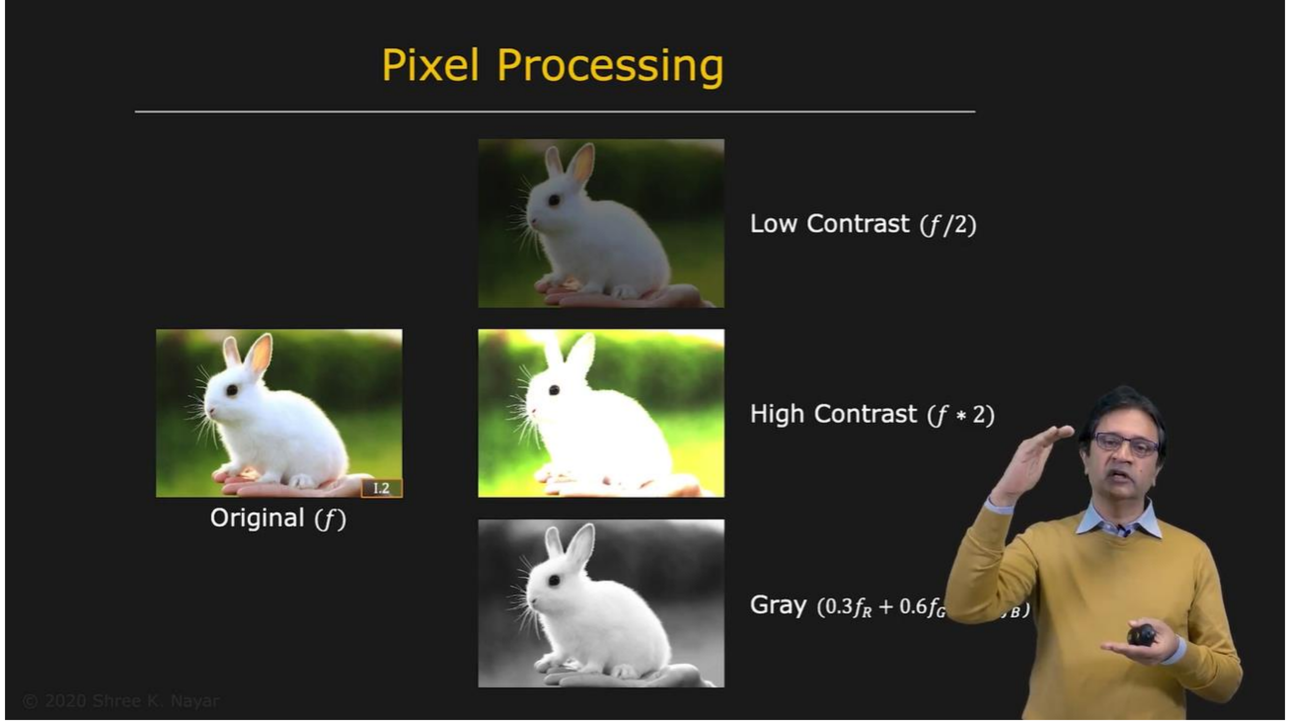
- 컬러 이미지라면 각각 RGB값을 갖는다고 생각!



- 픽셀 프로세싱 이란 픽셀의 밝기 값을 확인해서 어떠한 밝기 값이나 다른 색상으로 매핑하는 것
- 말 그대로 어떠한 작업을 할 때 픽셀값을 사용해서 처리를 하는 것
- 블러 필터: 주변 픽셀의 평균을 사용해 이미지를 부드럽게 한다.
- 색상 조정: 필터 값을 변경해 이미지의 밝기를 조정



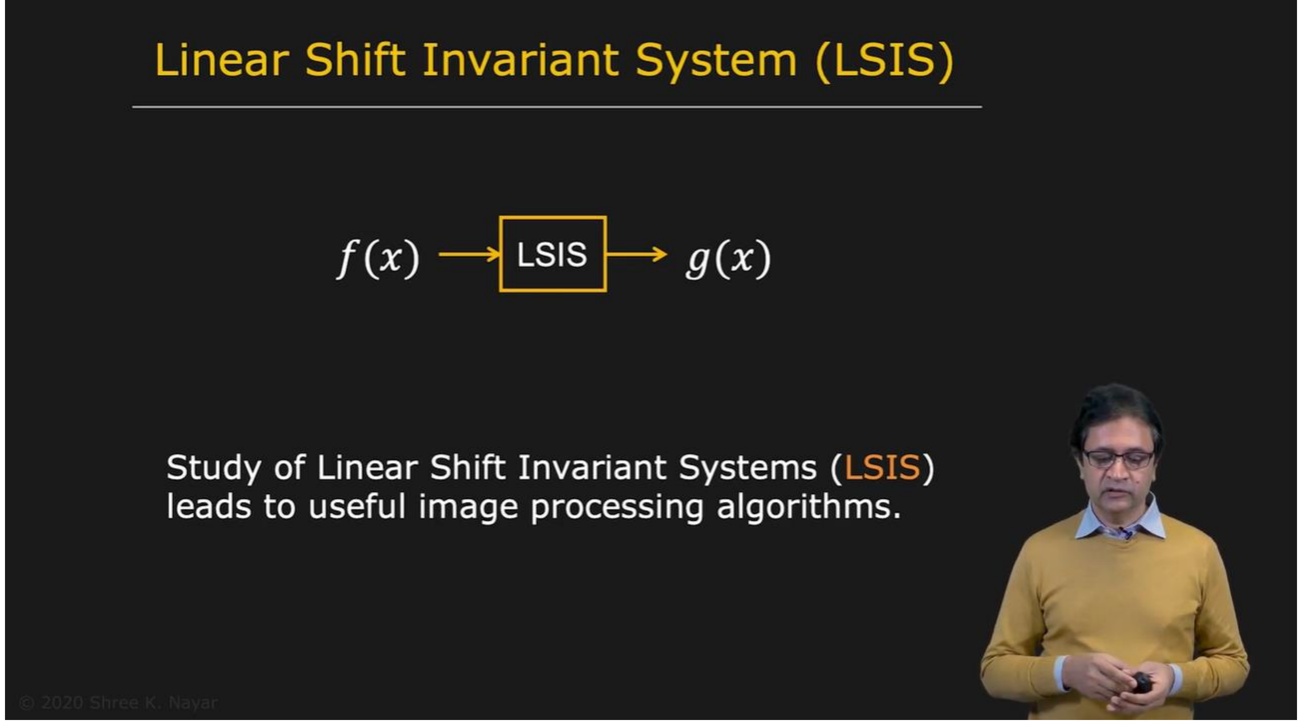
- 이미지의 색상을 조정하는 방법
- 어둡게 하고 싶으면 3채널 각각에서 -128을 한다.
- 밝게 하고 싶으면 +128, 반전을 하고 싶다면 (8비트 이미지의 경우) 255에서 각 픽셀 값들을 뺀다.



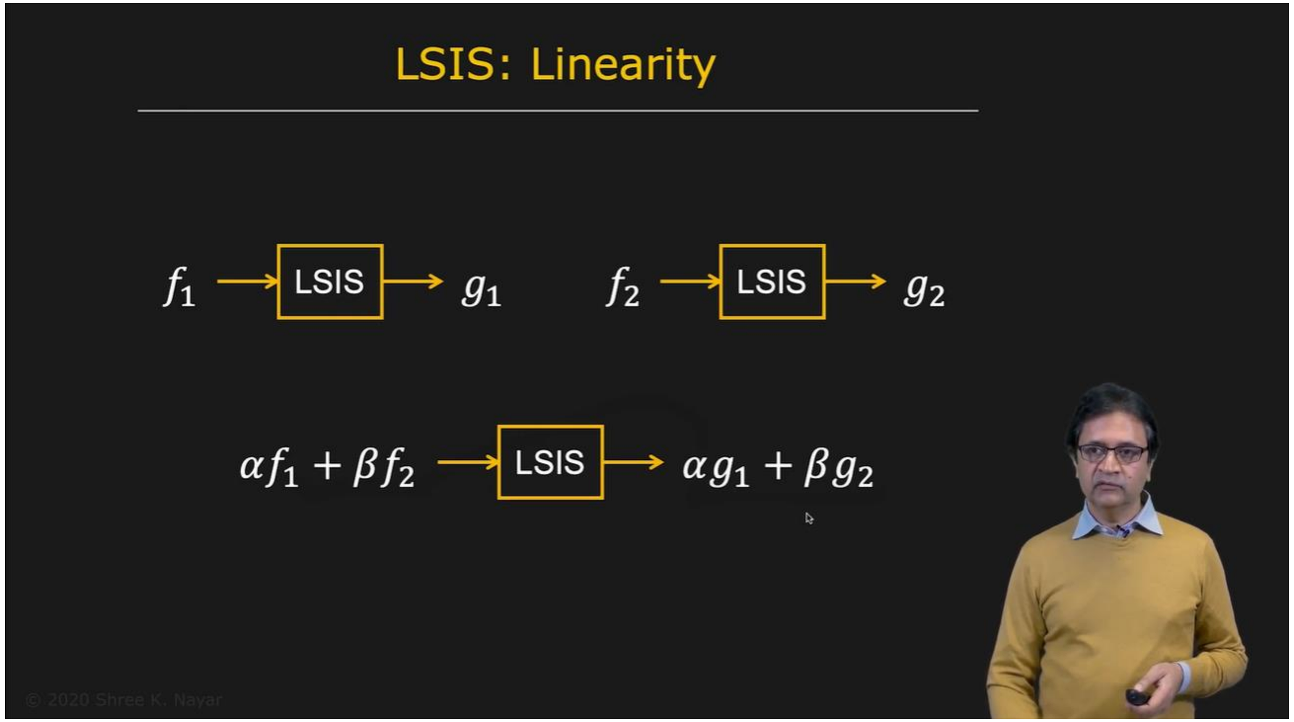
- 이미지의 대비를 변경하는 방법
- 대비를 낮추고 싶다면 각 픽셀 값에 /2를 해준다.
- 대비를 높이고 싶다면 각 픽셀 값에 *2를 해준다. 이때 픽셀 값은 이미지 자체의 동적범위를 넘어설 수 없다.
- 토끼의 하얀색 몸이 **과포화**된 상태!
- 픽셀의 RGB 값을 변경해 그레이스케일 이미지도 만들 수 있다.

Image Processing I

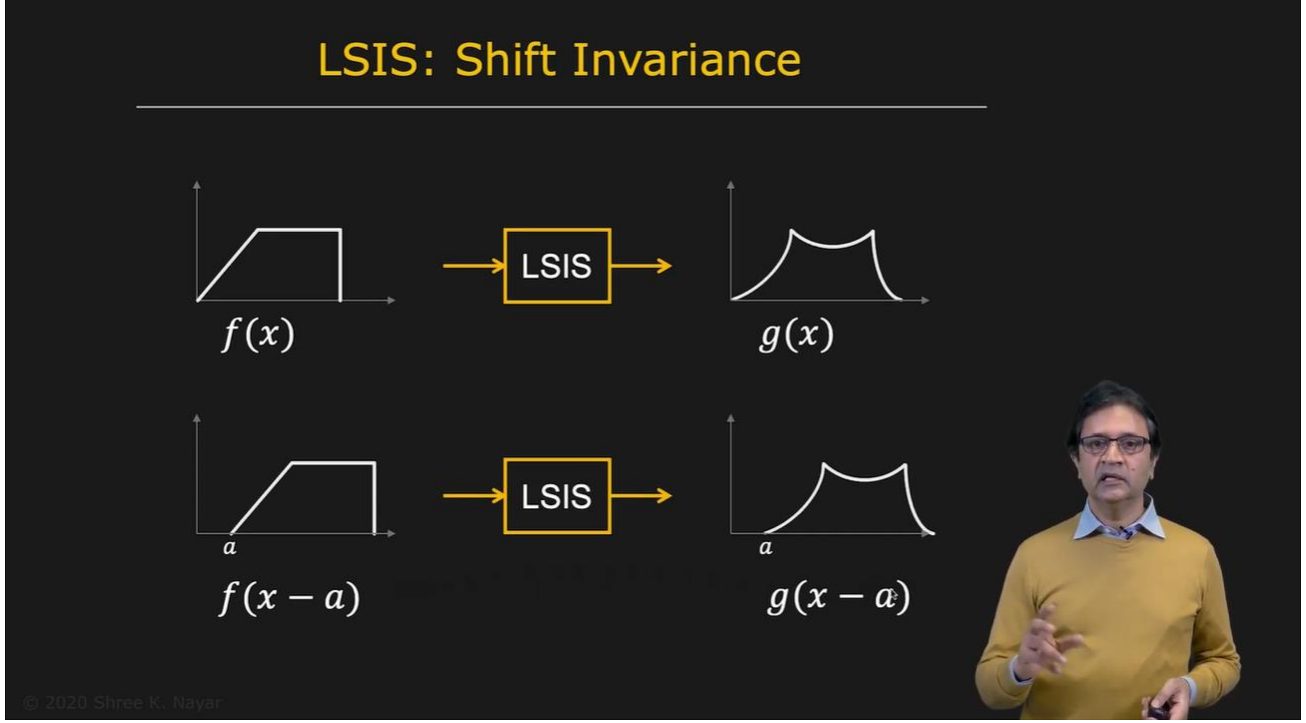
LSIS and Convolution



- 선형 이동 불변 시스템은 이미지 프로세싱 알고리즘에 중요한 역할을 한다.
- 입력 f(x)와 출력 g(x)가 있는 시스템



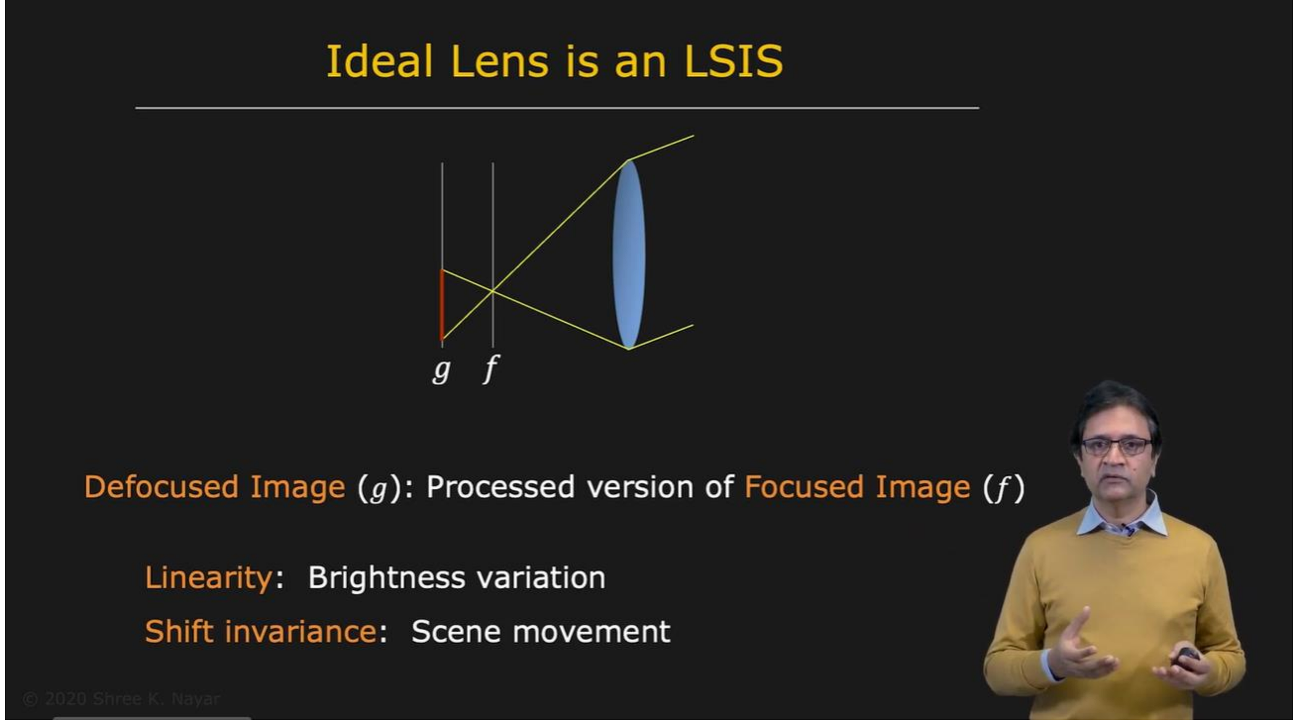
- 1. 선형성(Linearity)
- 입력 f1에 대해 출력 g1, 입력 f2에 대해 g2가 발생한다.
- 선형성을 가진 시스템은 α f1 + β f2 을 입력했을 때 α g1 + β g2를 출력한다.
- 입력 신호에 상수배를 하고 더했을 때, 출력 신호에도 동일하게 상수배와 덧셈 연산이 적용되는 성질



- 2. 이동 불변성

- 입력 f(x)을 a만큼 움직이면 출력 g(x)도 a만큼 움직인다.

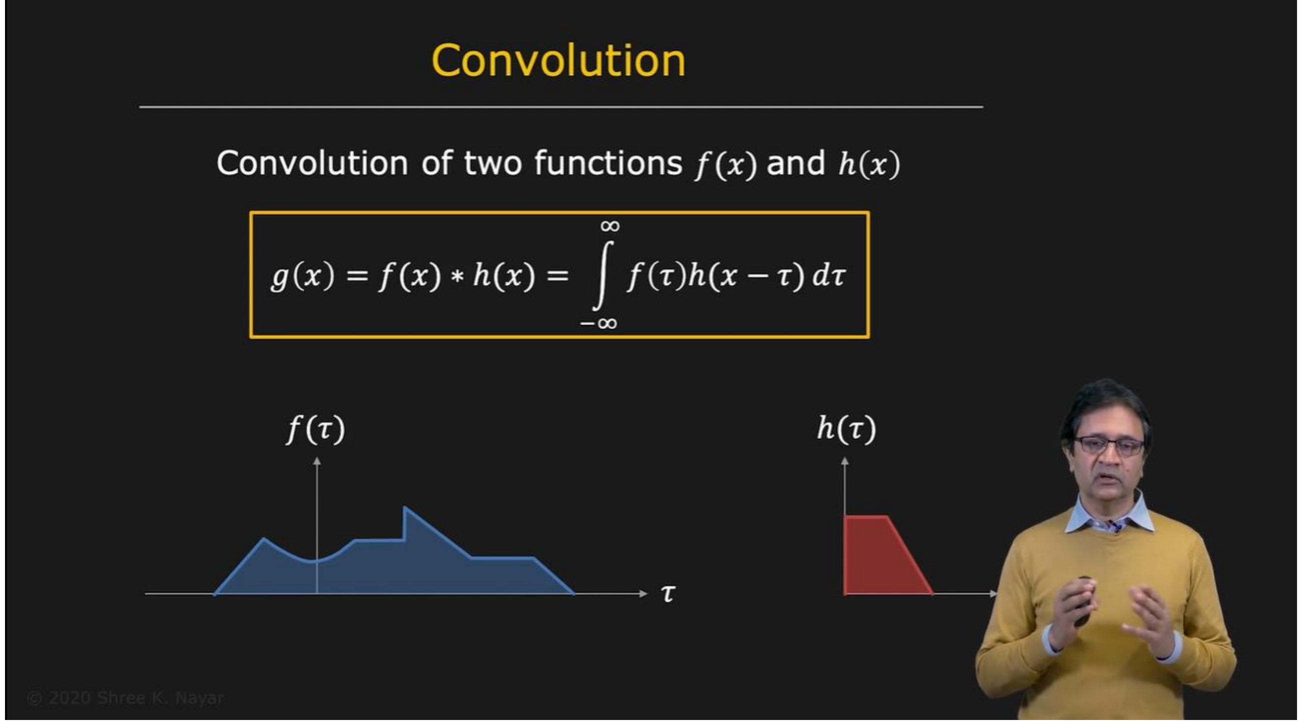
- 선형성과 이동 불변성을 모두 만족하는 시스템을 LSIS라고 한다.



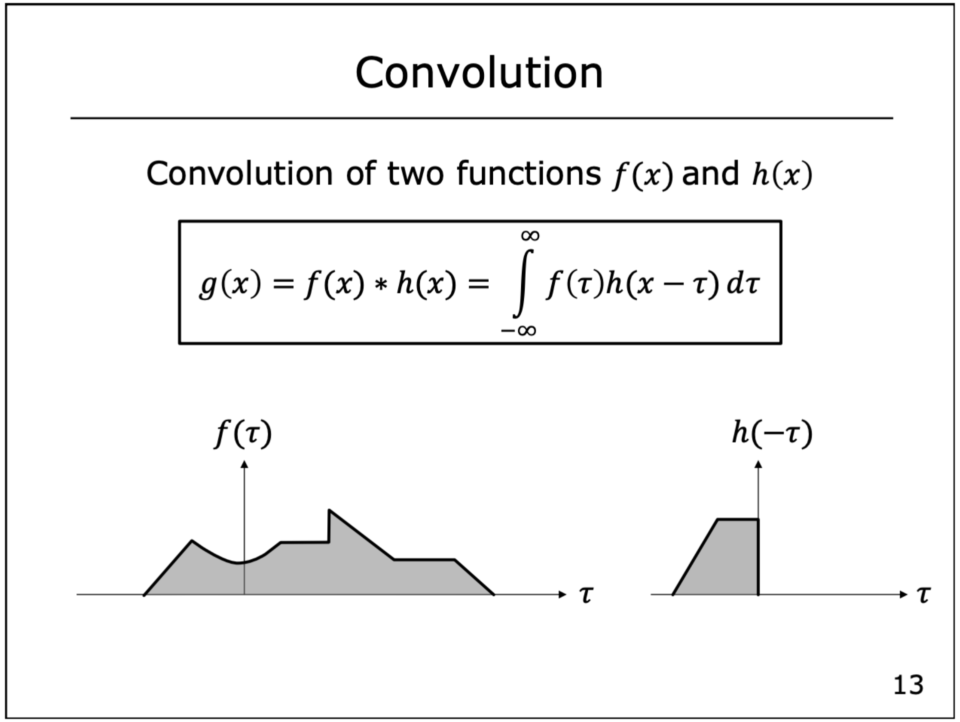
- 초점이 맞춰진 f와 초점이 흐린 g가 있다.

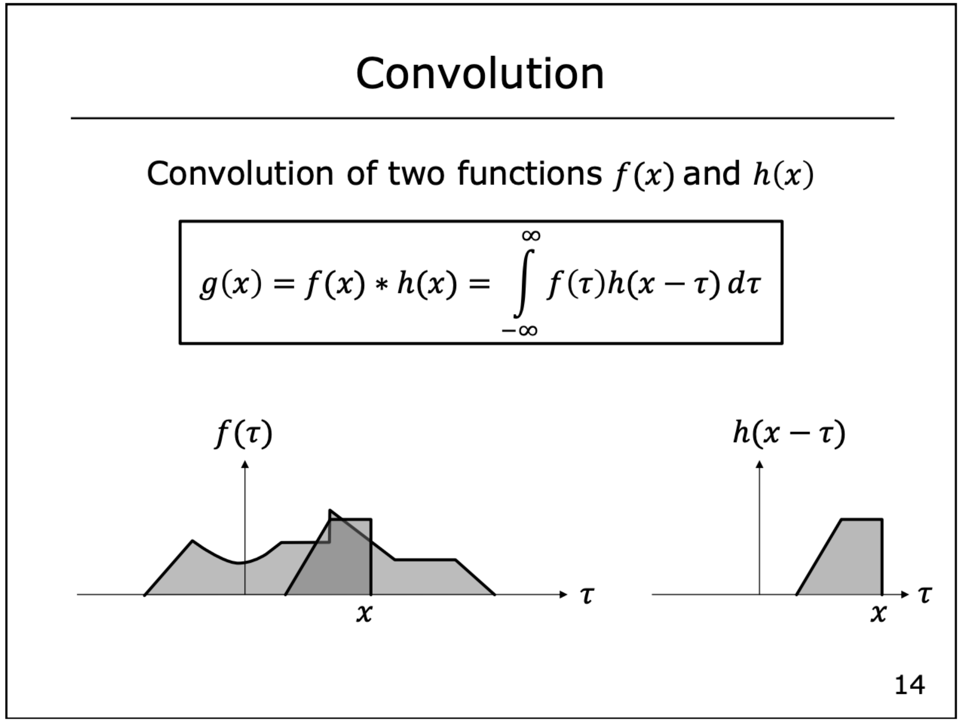
- 3차원 장면의 밝기가 증가하면 f와 g 모두 밝기가 올라간다. -> 선형성

- 3차원 장면 내 물체를 이동시키면 f와 g에서도 개체가 똑같이 이동한다. -> 이동 불변



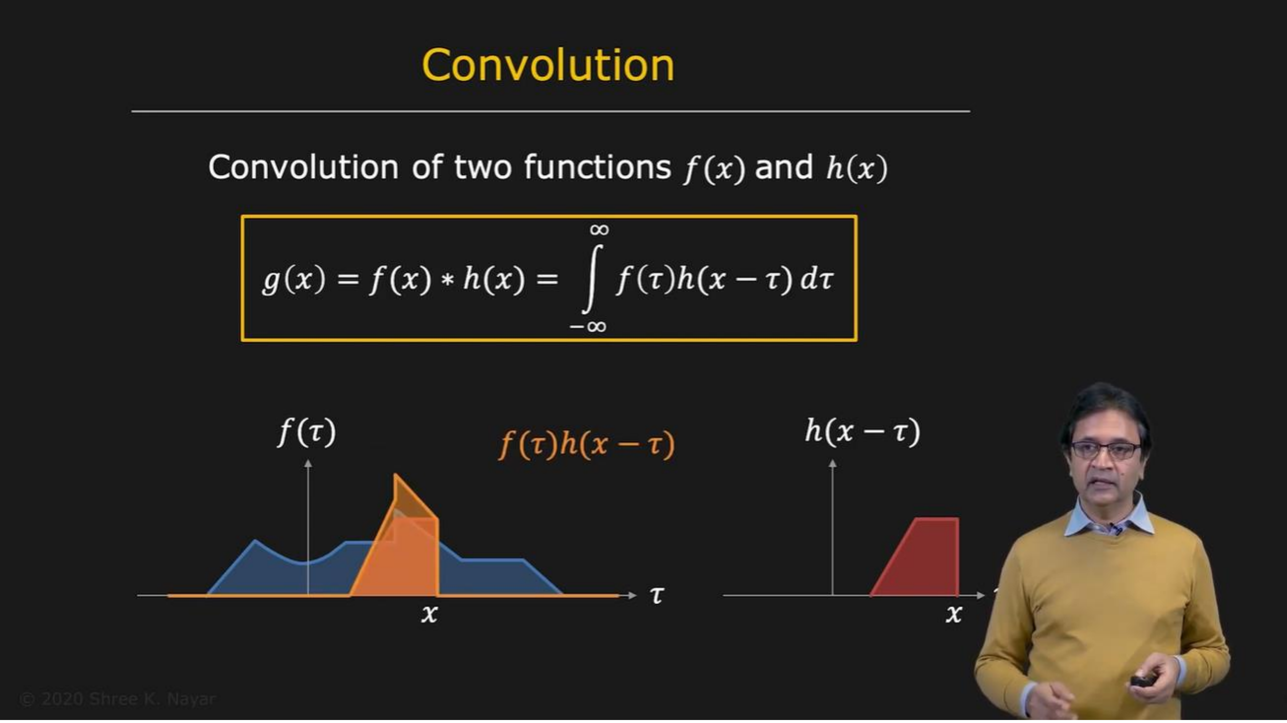
- 컨볼루션이란 두 함수나 신호를 결합하여 새로운 출력을 생성하는 방식으로 기호는 * 이다.
- 원본 신호에 대해 특정 패턴을 강조하거나 특정 노이즈를 제거하는 데 사용
- 시간에 따라 적분을 하는데 두 함수가 각 순간에 얼마나 겹치는 지 평가하는 것이다.
- 음성신호라면 1차원의 시간이 되겠지만 이미지에서는 2차원의 픽셀이 얼마나 일치하는 지를 보는 것.

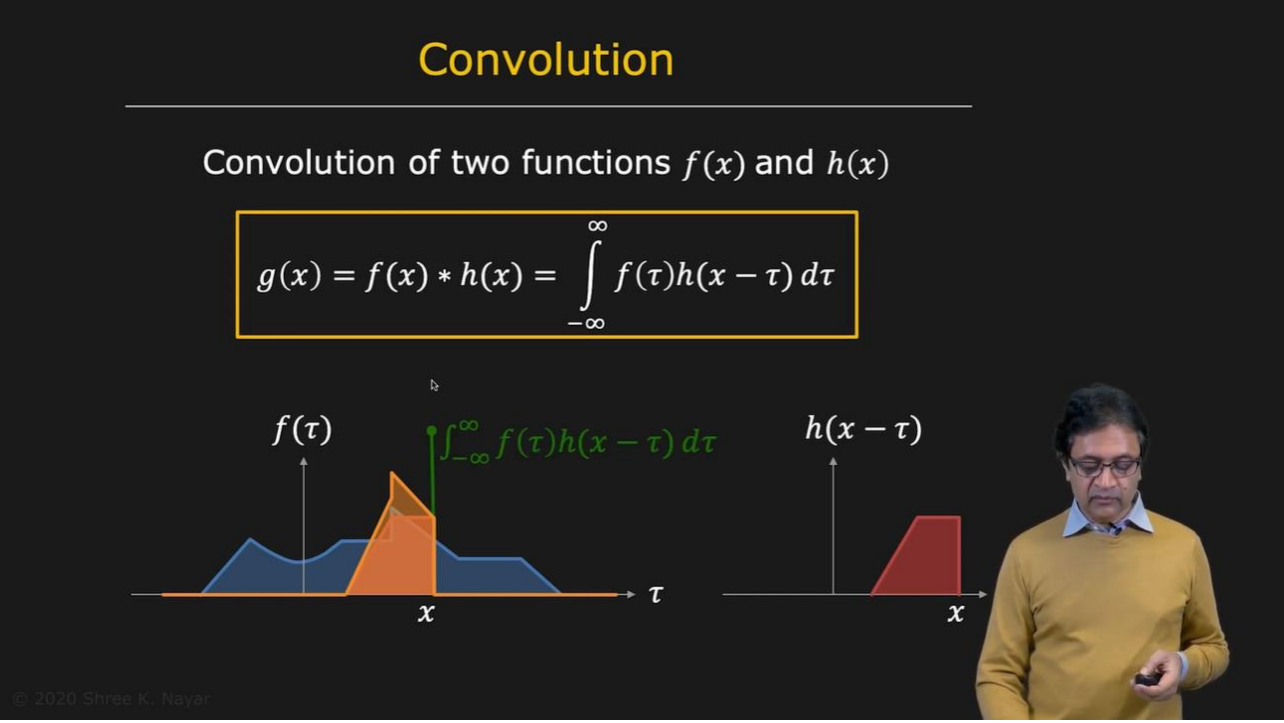




- 1. h(τ)를 수직 축에 대해 뒤집어 h(- τ)을 구하고 이를 x만큼 이동시켜 h(x - τ)를 만든다.

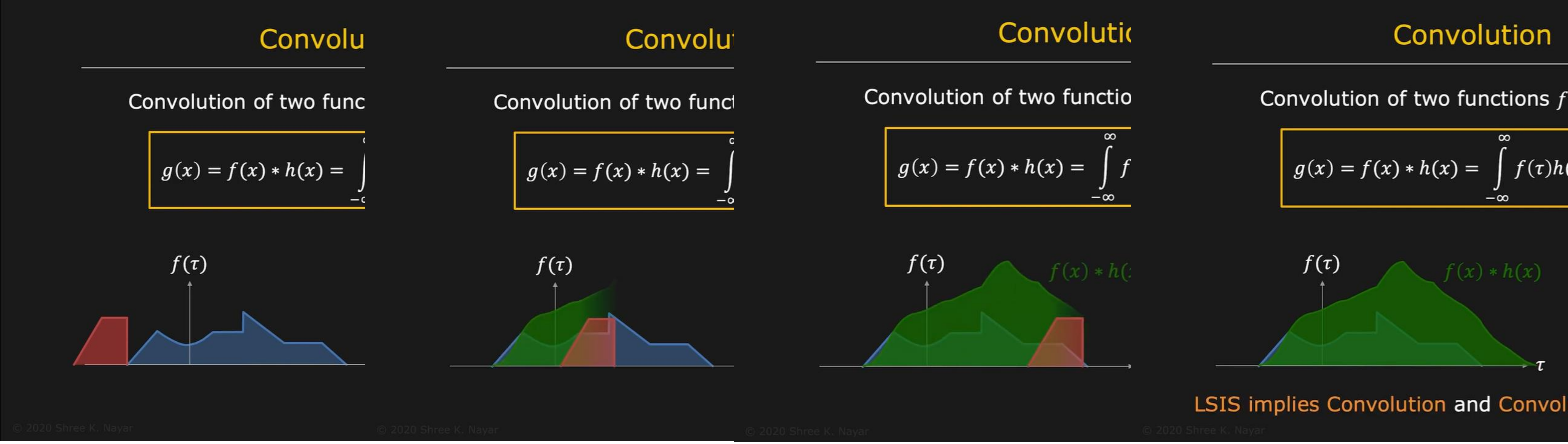
- 2. h(x - τ)를 f(τ) 위에 겹쳐둔다.



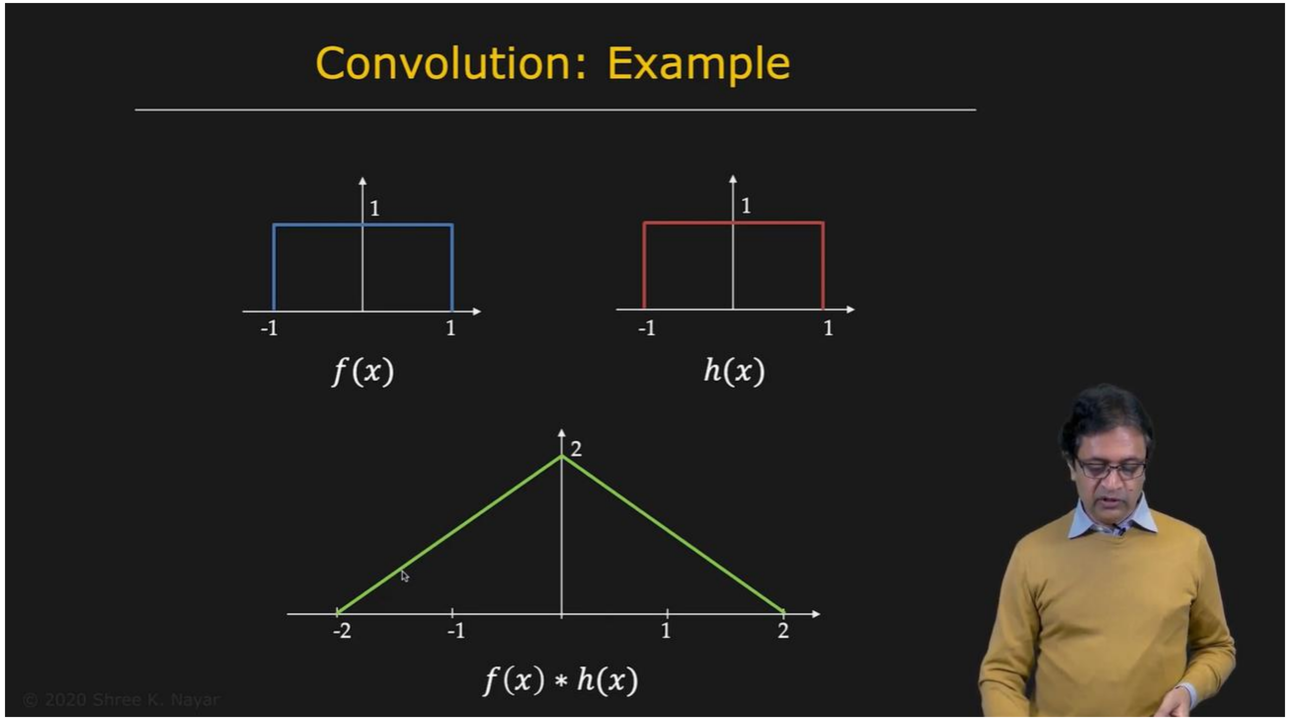


- 3. 겹쳐진 두 함수 f(τ)와 h(x - τ)의 곱을 구하고 적분을 구한다.

- 4. 해당 스칼라 값이 특정 x에서의 g 값이다.



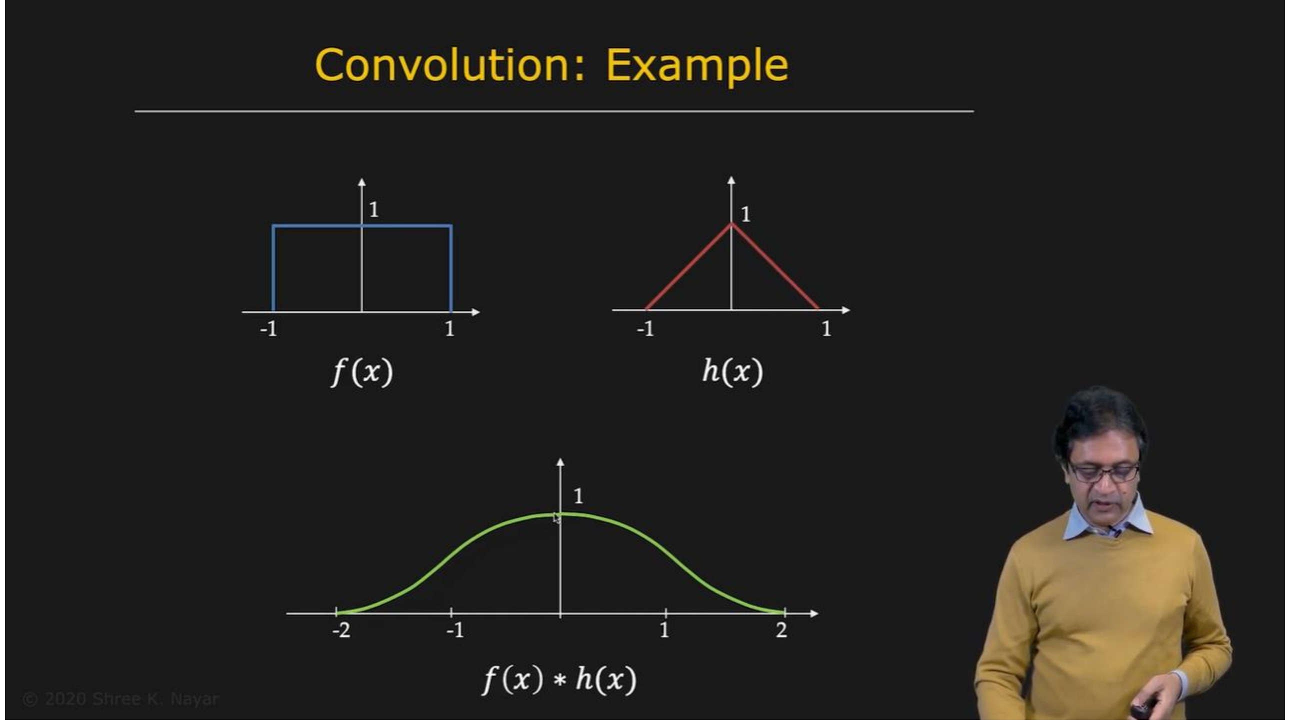
- 전체 함수 g(x)를 찾기 위해서는 h(- τ)을 만들고 $-\infty$에서부터 ∞까지 슬라이딩을 시킨다.
- 그리고 이를 f(x)와 h(- τ)을 곱하고 적분하면 g(x)가 나온다.



- 직사각형 h(x)함수를 뒤집어 h(-x)를 만들고 (같은 모양) $-\infty$부터 ∞까지 슬라이드 한다.

- 그러면 두 함수가 만나기 시작하는 -2부터 겹치는 영역이 증가해 정확히 겹쳐지는 원점까지 계속 증가한다.

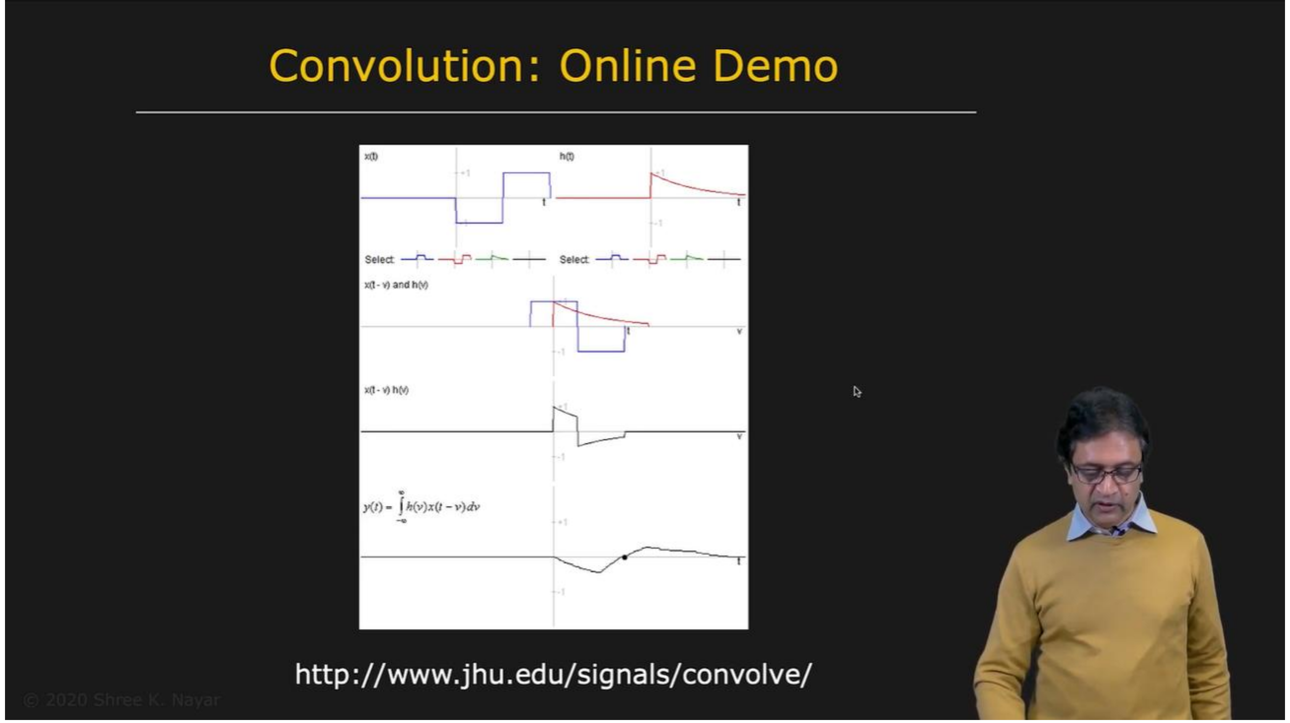
- 그때 겹치는 영역은 2가 된다.그리고 서서히 줄어들고 x=2이가 되면 겹치는 영역이 0이 된다.



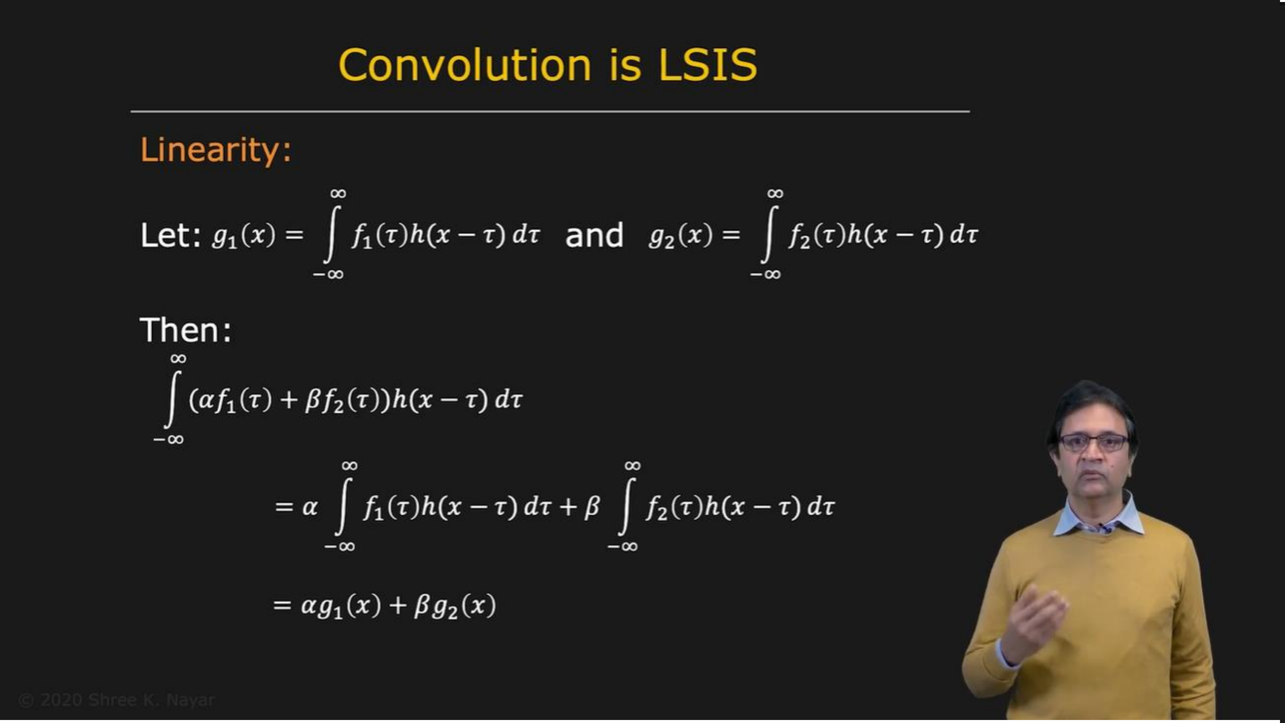
- 삼각형 h(x)함수를 뒤집어 h(-x)를 만들고 (같은 모양) $-\infty$부터 ∞까지 슬라이드 한다.

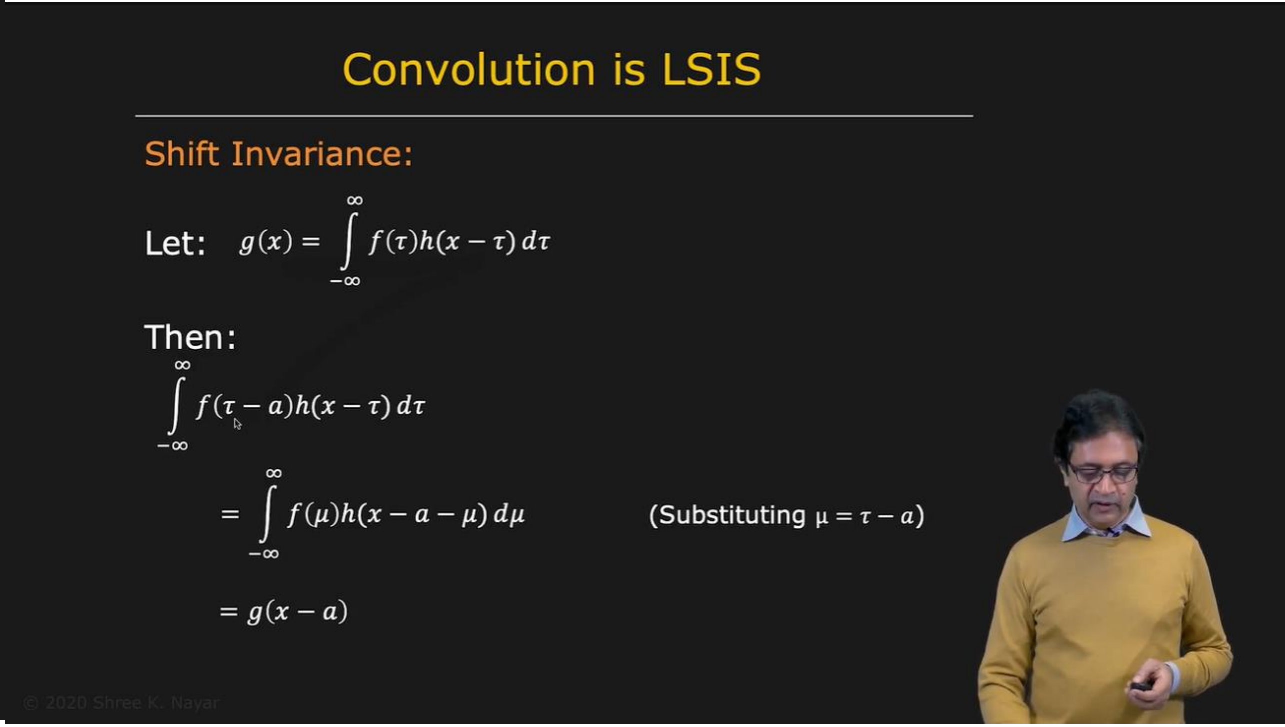
- 그러면 두 함수가 만나기 시작하는 -2부터 겹치는 영역이 증가해 정확히 겹쳐지는 원점까지 계속 증가한다.

- 그때 겹치는 영역은 삼각형 너비인 1이 된다.그리고 서서히 줄어들고 x=2이가 되면 겹치는 영역이 0이 된다.



https://pages.jh.edu/signals/convolve/





- 1. 선형성

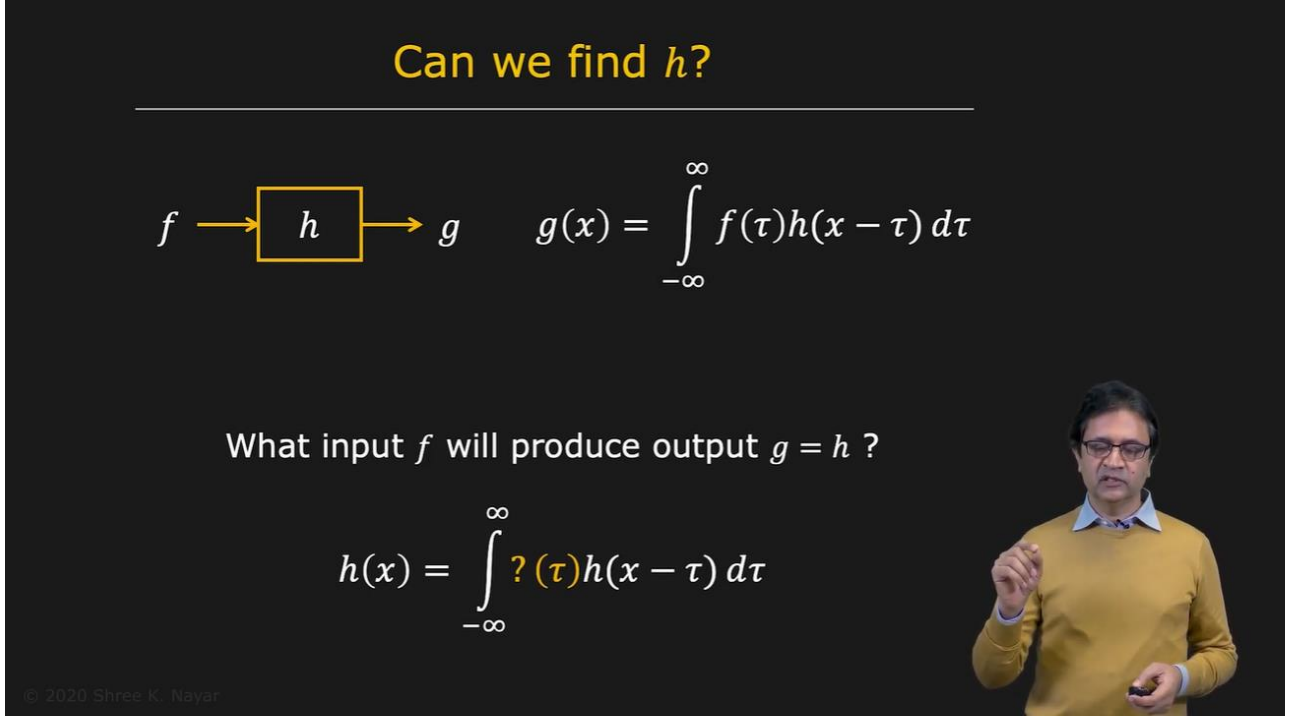
  f1과 h의 컨볼루션 결과를 g1, f2와 h의 컨볼루션 결과를 g2라고 가정

  두 입력 f1과 f2에 상수배를 하고 더하면 그대로 출력에도 똑같이 상수배를 하고 더한 αg1(x) + βg2(x)가 된다.

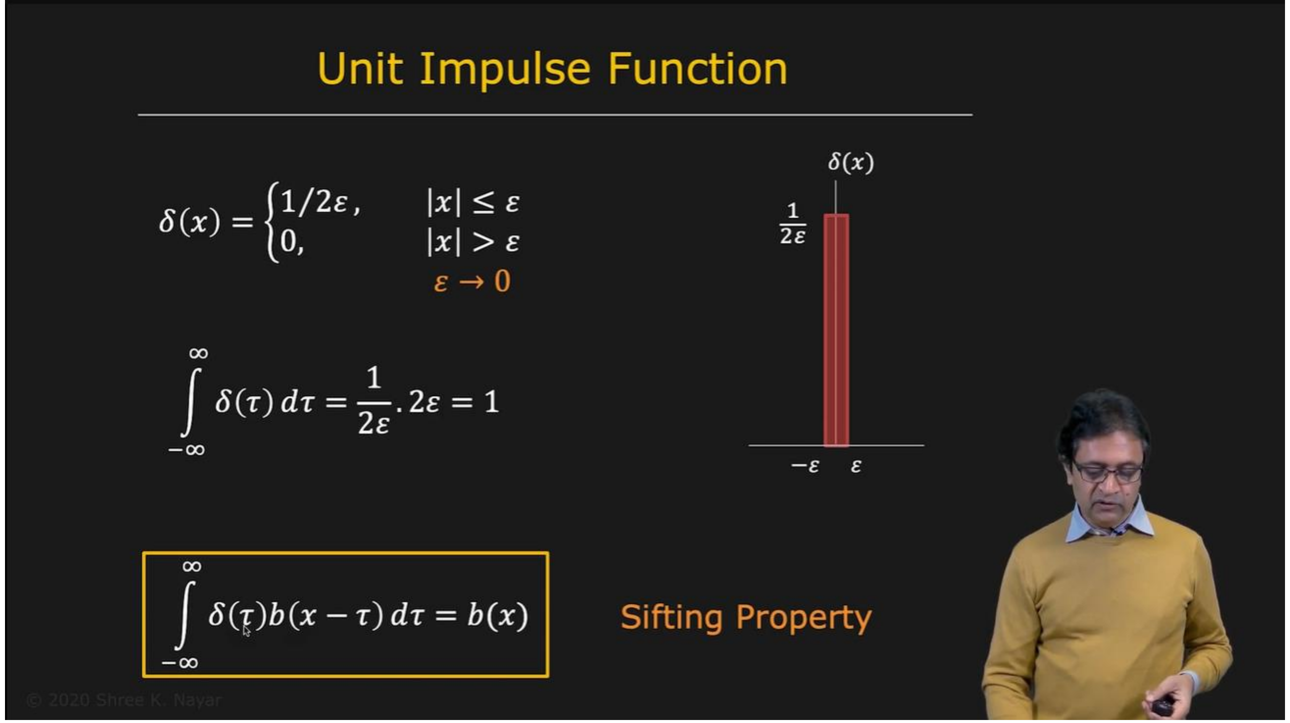
- 2. 이동 불변

  입력 f(τ) 대신 a만큼 이동시켜 f(τ - a)을 만들고 $\mu = \tau -$ a로 치환해 식을 정리하면 출력도 a만큼 이동하게 된

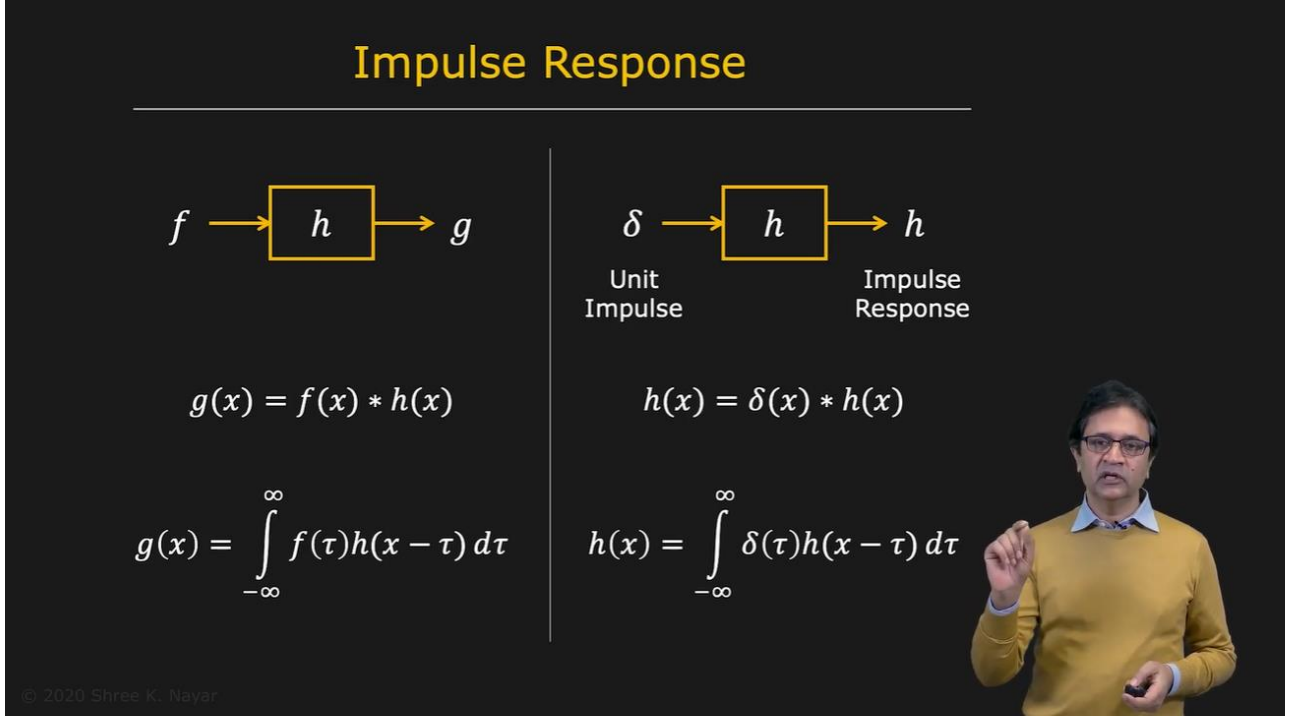
두 가지를 모두 만족한 컨볼루션은 선형 이동 불변 하다.



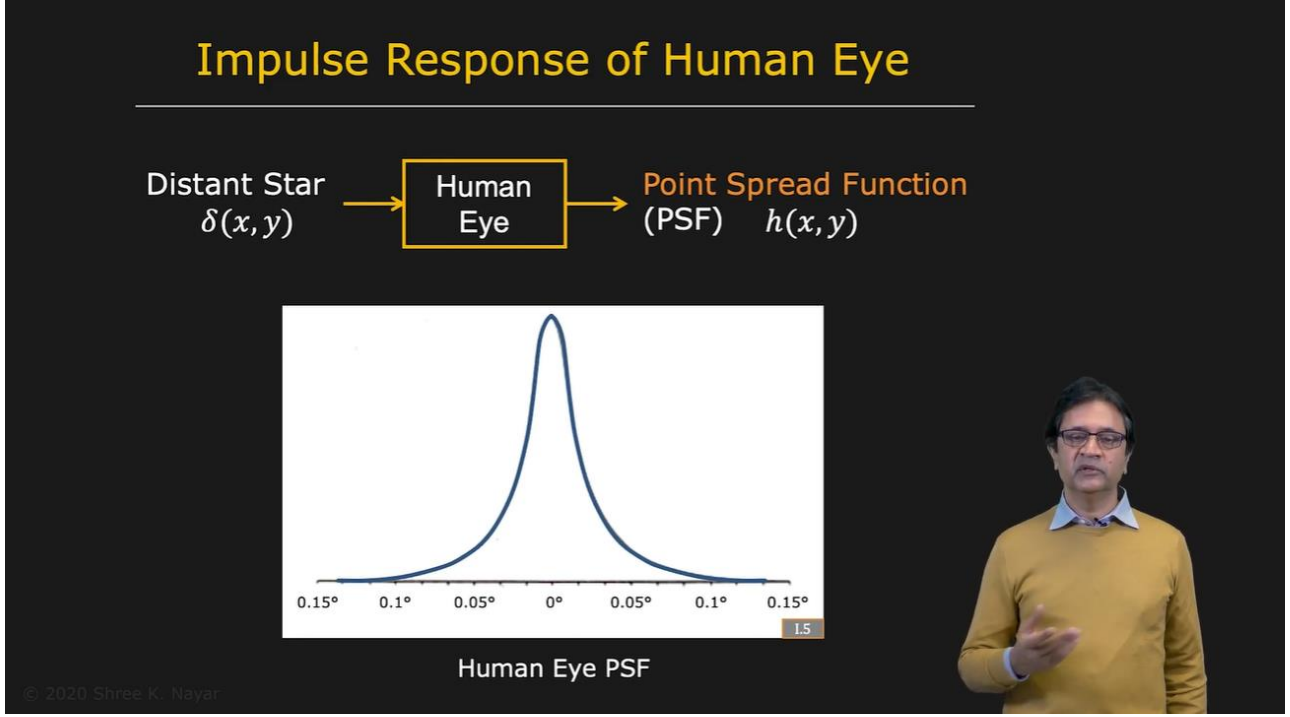
- 우리는 컨볼루션이 선형 이동 불변하다는 것을 알지만 입력이 어떤 함수와 연산되는 지는 모른다. **블랙박스 시스템**
- h(x)를 구하는 방법은 따로 없을까?



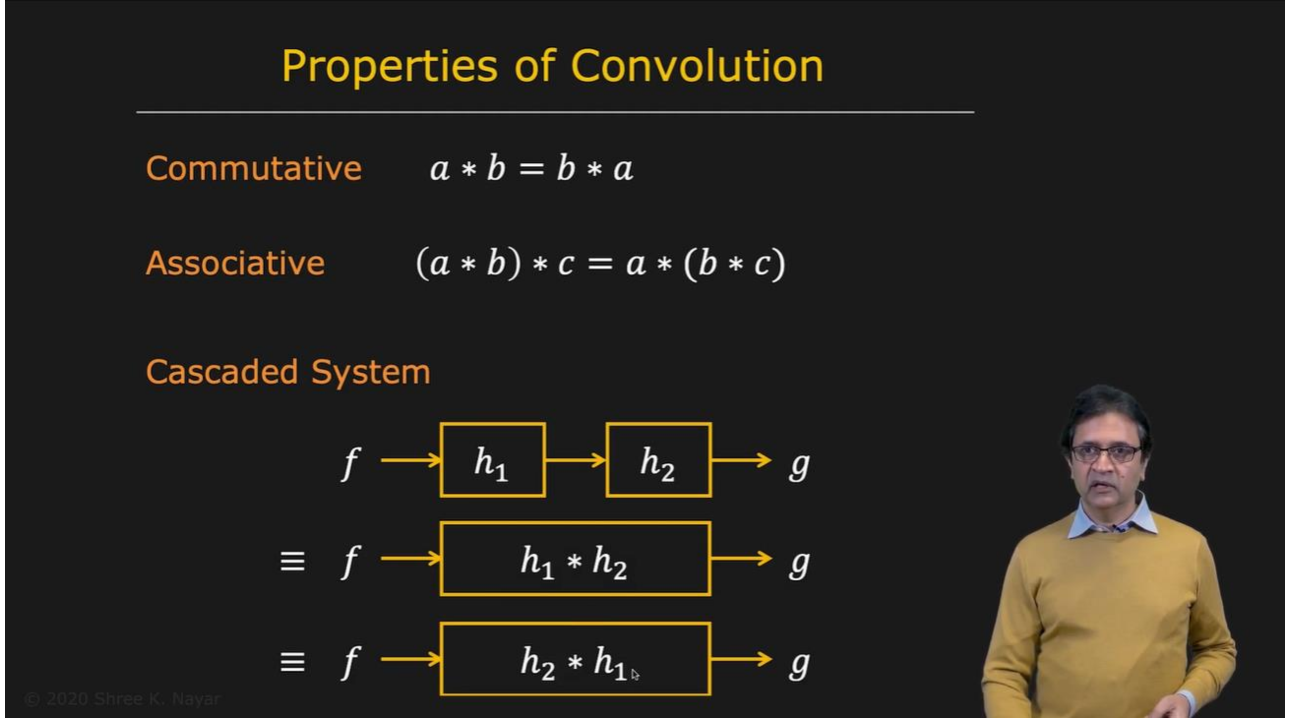
- 유닛 임펄스 함수(델타 함수)의 면적은 1이다.
- 이 함수를 입력으로 넣어서 h(x)와 컨볼루션을 하면 δ(x) · h(x)는 δ(x) 면적이 1이므로 h(x)의 값을 얻게 된다.
- 따라서 단위 임펄스 함수와 컨볼루션한 모든 함수는 원래 함수 자체가 된다.



- 유닛 임펄스 함수와 컨볼루션한 결과 h 함수를 임펄스 응답이라고도 한다.
- 임펄스 응답은 블랙박스에 어떤 함수가 있는 지를 설명해 줄 수 있다.

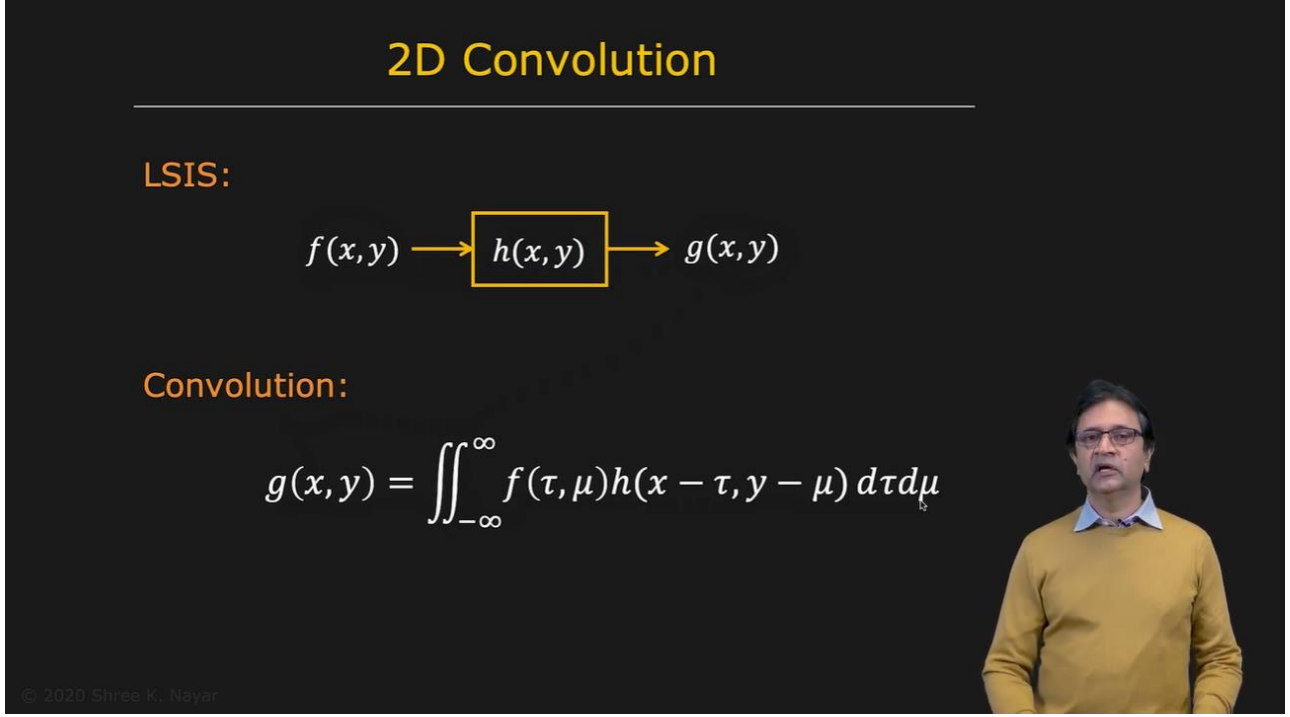


- 사람의 눈의 임펄스 응답을 어떻게 관찰 할 수 있을까?
- 눈에 2차원 임펄스 함수를 주면 눈의 임펄스 응답을 측정할 수 있다.
- 아주 멀리 있는 빛을 임펄스 함수로 주고 망막에 형성하는 이미지는 눈의 임펄스 응답



- 컨볼루션 연산은 교환 법칙과 결합 법칙이 성립한다.

- 따라서 두 번의 컨볼루션 연산을 하나의 컨볼루션 연산으로 결합하는 것이 가능하다.



- 2차원의 이미지의 LSIS의 시스템에서는 입력도, 임펄스 응답도, 출력도 모두 2차원이 된다.
- 컨볼루션 연산에서는 h 함수가 x, y 각각 두 번 뒤집히게 되고 이중 적분을 한다.