컴퓨터 그래픽스
4. 2차원 그래픽스의 윈도우와 뷰포트

2020년 2학기

학습 내용

- **윈도우와 뷰포트**
  - 2차원 뷰잉 파이프라인: 윈도우, 뷰포트
  - 클리핑 알고리즘

윈도우와 뷰포트

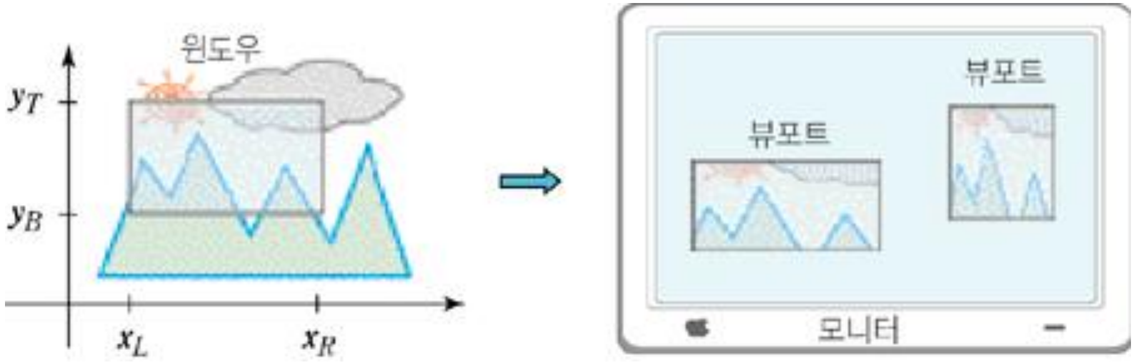
- **2차원 그래픽스의 뷰잉 파이프라인**
  - 모델 좌표계: 개별 객체를 표현하기 위해 사용되는 좌표계
  - 월드 좌표계: 각 모델 좌표계의 통합된 좌표계
  - 뷰잉 좌표계: 출력장치에 출력 위치 및 크기 설정하여 뷰포트에 출력, 뷰포트 좌표계
  - 정규 좌표계: 정규화된 좌표계
  - 장치 좌표계: 출력하려는 장치 좌표계

윈도우와 뷰포트

- Window
  - 출력 장치에 표시하기 위해 선택된 세계 좌표 영역

- Viewport
  - 윈도우가 사상되는 출력 장치의 영역

- **윈도우-뷰포트 변환에 의한 효과**
  - Zooming 효과: (zoom in/zoom out)
  - Panning 효과: 카메라 각도를 돌려가면서 비디오 촬영하는 것과 같은 효과
  - 한번에 여러 개의 화면을 가질 수 있다.

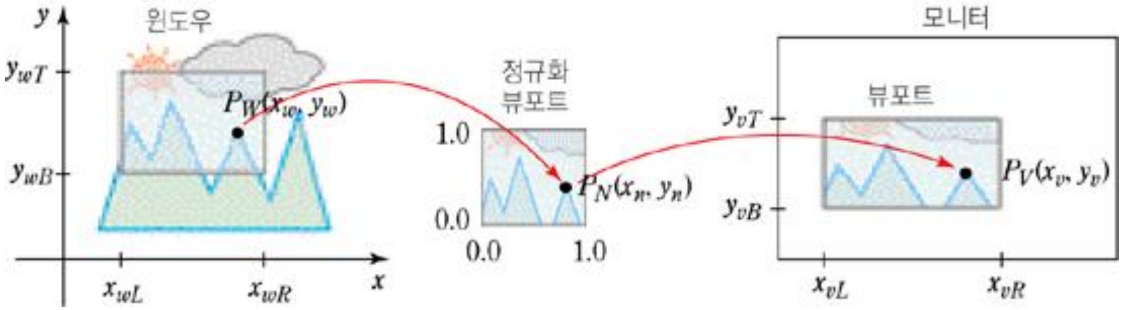


윈도우와 뷰포트

- **윈도우-뷰포트 좌표 변환**
  - (x_w, y_w): 윈도우 내의 점　　　　(x_v, y_v): 뷰포트 안의 점
    - 행렬

    - $x_v = x_{vL} + (x_w - x_{wL})s_x$,　　　$s_x = \dfrac{(x_{vR} - x_{vL})}{(x_{wR} - x_{wL})}$
    - $y_v = y_{vB} + (y_w - y_{wB})s_y$,　　　$s_y = \dfrac{(y_{vT} - y_{vB})}{(y_{wT} - y_{wB})}$

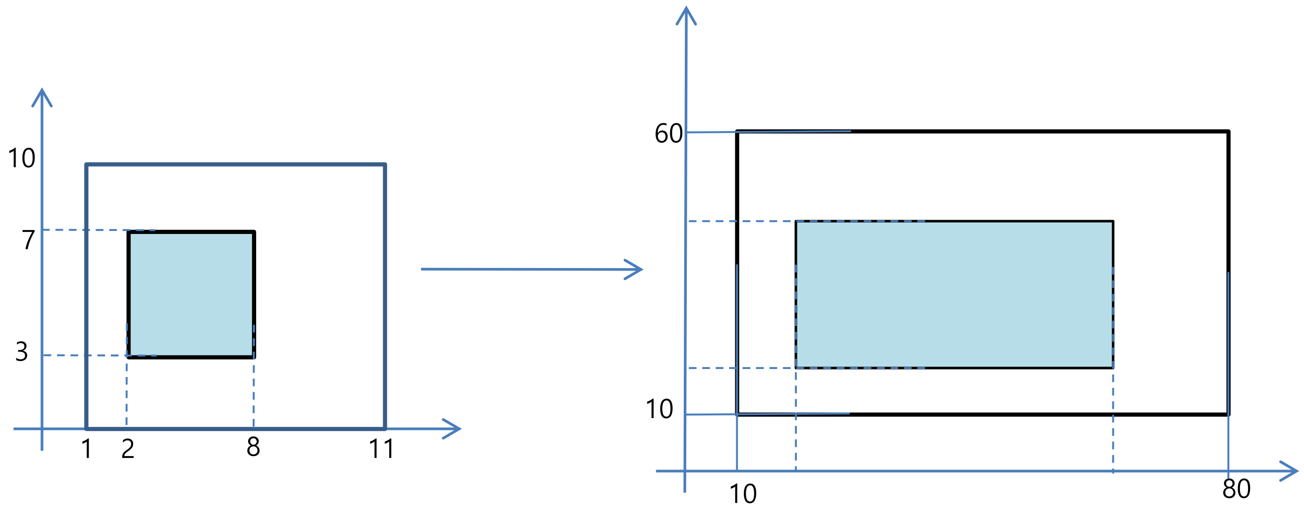
x_{wR}, x_{wL}: 윈도우의 x방향 최대값, 최소값
y_{wT}, y_{wB}: 윈도우의 y방향 최대값, 최소값



윈도우와 뷰포트

- **예) 다음의 도형에 대하여 윈도우-뷰포트 변환이 주어졌을 때 변환 좌표 값**
  - 윈도우 (1, 0) (11,10) → 뷰포트 (10, 10) (80, 60)

    도형 좌표: (2, 3) (8, 7)로 이루어진 사각형이 윈도우-뷰포트 변환 후 좌표값:



Clipping (클리핑)

- **Clipping이란**
  - 윈도우-뷰포트 변환 시, 출력장치에 표시되어서는 안될 그림영역을 제거한 뒤, 나머지 그림영역을 출력화면에 나타내는 것

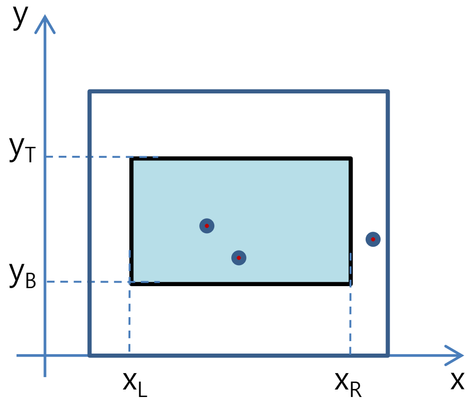
  - 월드 좌표 클리핑:
    - 윈도우를 설정할 때 윈도우 바깥 영역을 제거하여 윈도우 내부 영역만 뷰포트로 매핑시키는 방법
  - 뷰포트 클리핑:
    - 월드 좌표계를 표현된 그림 전부를 뷰포트로 매핑시킨 후 뷰포트 외부에 위치한 객체나 그림의 일부를 제거하는 방법

  - 두 클리핑이 모두 결과는 같다.

Clipping (클리핑): 점

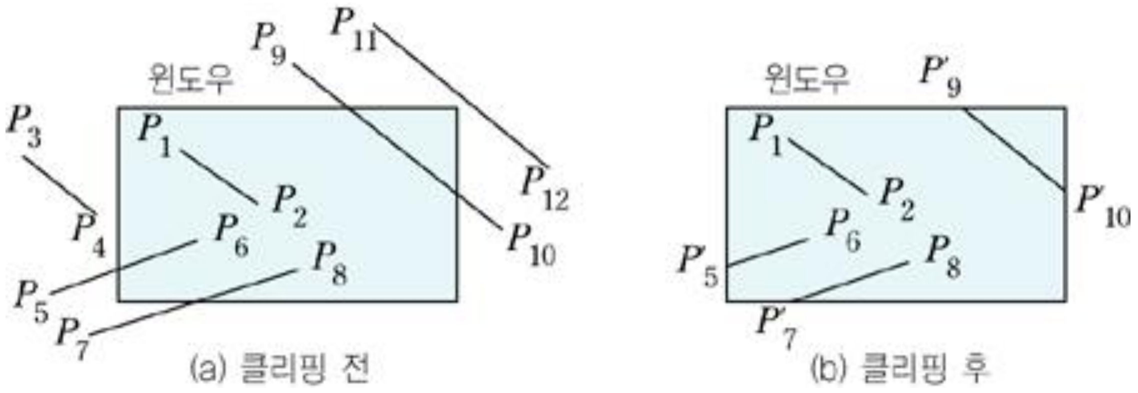
- **점 클리핑**
  - 클리핑 되는 객체가 점
  - 한 점 P(x, y)는
    - $x_L \le x \le x_R$, $y_B \le y \le y_T$

    이면 그려진다.



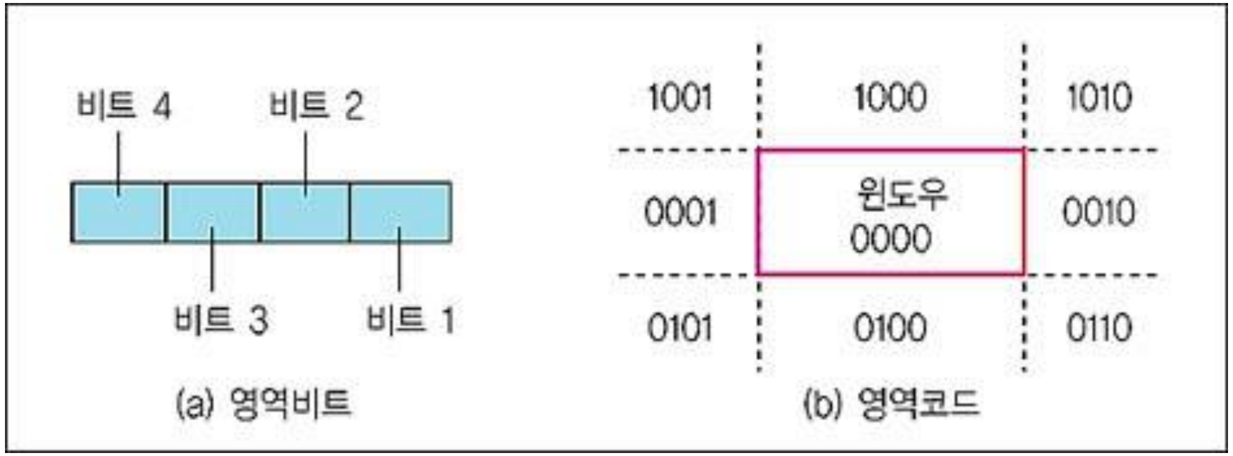
Clipping (클리핑): 선

- **선 클리핑**
  - 클리핑 되는 객체가 선분
  - 선분이 클리핑 영역의 내부 또는 외부에 완전히 포함되는가/포함되지 않는가
  - 부분적으로 속하는가
  - 속한다면 교차점은 어떻게 구하는가



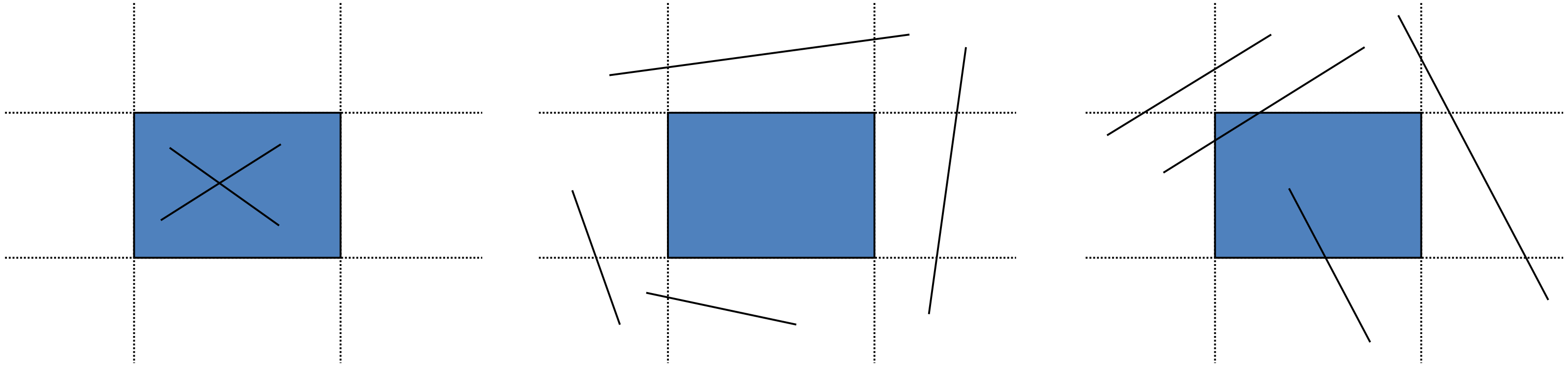
Clipping (클리핑): Cohen-Sutherland 알고리즘

- Cohen-Sutherland 알고리즘
  - 윈도우를 중심으로 전체 그림 영역을 9개 영역으로 구분
  - 각 영역에 4비트를 사용하여 영역코드를 부여한다.
    - 비트 1: 윈도우의 왼쪽에 있으면 1
    - 비트 2: 윈도우의 오른쪽에 있으면 1
    - 비트 3: 윈도우의 아래쪽에 있으면 1
    - 비트 4: 윈도우의 위쪽에 있으면 1



Clipping (클리핑): Cohen-Sutherland 알고리즘

– 알고리즘 수행 과정
  - 양 끝점의 코드가 모두 0000이면 →
  - 양 끝점의 코드 중 한 쪽 코드는 0이고 다른 쪽 코드는 0이 아니면 →
  - 양 끝점 코드가 모두 0이 아니고, 양 끝점 코드간 AND 연산이 0이 아니면 →
  - 양 끝점 코드가 모두 0이 아니고, 양 끝점 코드간 AND 연산이 0이면 →



Clipping (클리핑): Cohen-Sutherland 알고리즘

- **주어진 선분에 대한 교차점 구하기**
  - 수직 경계: $x = x_L$ 또는 $x = x_R$

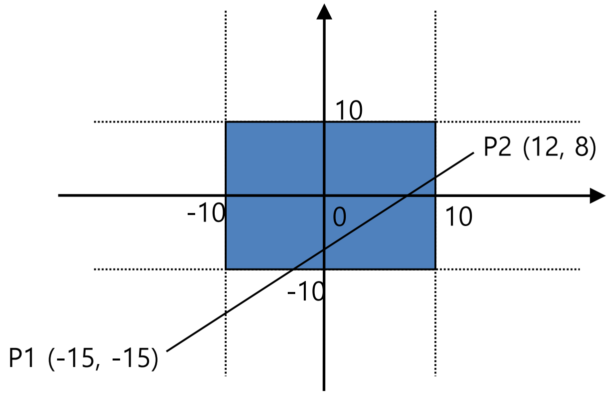
    $y = y1 + m(x - x1)$, $m = \frac{y2-y1}{x2-x1}$

  - 수평 경계: $y = y_B$ 또는 $y = y_T$

    $x = x1 + \frac{(y-y1)}{m}$, $m = \frac{y2-y1}{x2-x1}$

x1, y1: 선분의 끝 점

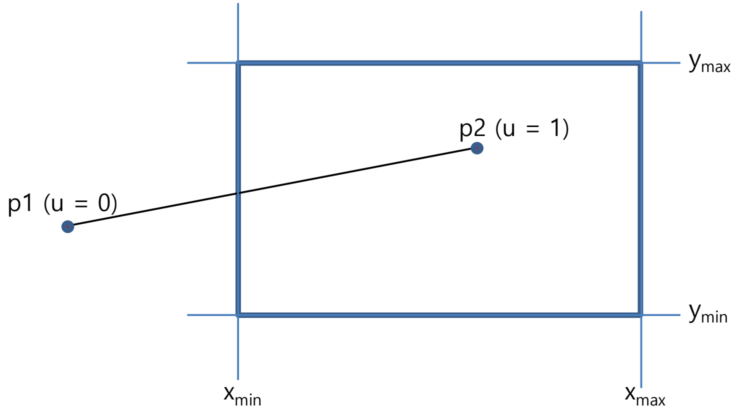
x_L, x_R , y_B, y_T : 윈도우의 경계



Clipping (클리핑): Liang-barskey 알고리즘

- **매개변수 방정식을 이용하여 선분을 윈도우 경계에 대하여 자르는 알고리즘**
  - 선을 나타내는 매개변수 방정식은
    - $p(u) = (1- u)p_1 + up_2$
      $= p_1 + u(p_2 - p_1)$ $0 \le u \le 1$

    즉, $x = x_1 + u(x_2 - x_1)$
    $y = y_1 + u(y_2 - y_1)$



- **끝점이 P1 = (x1, y1) P2 = (x2, y2) 인 선분일 때**
  - 매개변수 방정식 사용하여 임의의 점 P (x, y) 을 표시
    - $x = x1 + (x2 - x1)u$ $0 \le u \le 1$ ($x2 - x1 \rightarrow d_x$)
    - $y = y1 + (y2 - y1)u$ $0 \le u \le 1$ ($y2 - y1 \rightarrow d_y$)
      - $u = 0$ $\rightarrow x = x1, y = y1$
      - $u = 1$ $\rightarrow x = x2, y = y2$

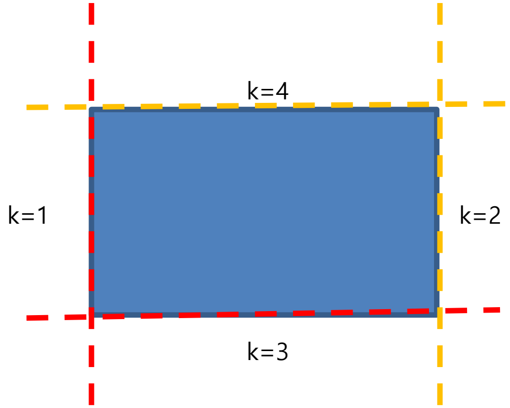
    - 선분위에 있는 점들은 아래의 조건을 만족
      - $x_{min} \le x \le x_{max}$, $y_{min} \le y \le y_{max}$

Clipping (클리핑): Liang-barskey 알고리즘

- 매개 변수 방정식으로 다시 작성하면
  - $x_{min} \leq x1 + d_x u \leq x_{max}$ --- ① $(d_x = x2 - x1)$
  - $y_{min} \leq y1 + d_y u \leq y_{max}$ --- ② $(d_y = y2 - y1)$

  - ① 은 다음과 같다
    - » 왼쪽 가장자리에 대하여 $-d_x u < x1 - x_{min}$
    - » 오른쪽 가장자리에 대하여 $d_x u < x_{max} - x1$
  - ② 도 같은 방식으로 바꿀 수 있다.
    - » 아래쪽 가장자리에 대하여 $-d_y u < y1 - y_{min}$
    - » 위쪽 가장자리에 대하여 $d_y u < y_{max} - y1$

- 위의 식은, $up_k < q_k$ $k = 1, 2, 3, 4$
  - $p_1 = -d_x$ $q_1 = x1 - x_{min}$ ➔ left
  - $p_2 = d_x$ $q_2 = x_{max} - x1$ ➔ right
  - $p_3 = -d_y$ $q_3 = y1 - y_{min}$ ➔ bottom
  - $p_4 = d_y$ $q_4 = y_{max} - y1$ ➔ top



Clipping (클리핑): Liang-barskey 알고리즘

1. p_k == 0 일 때,
   - p_k가 0 이면, k번째 가장자리와 평행
     - q_k < 0이면, 그 가장자리 영역 밖에 있다.

       즉, $p_1 = -dx = -(x2 - x1) = 0$ ➔ x2 == x1

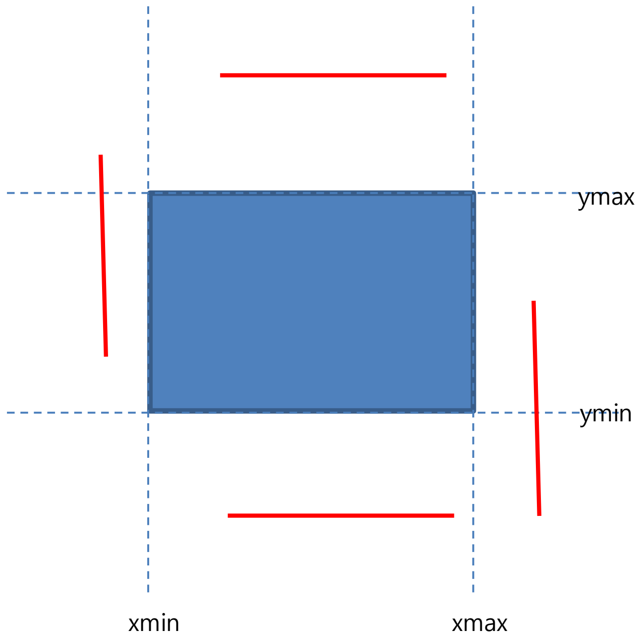
       a. $q_1 < 0$ ➔ $q_1 = (x1 - xmin) < 0$ ➔ x1 < xmin ➔ 영역 밖에 있다.

       b. $q_2 < 0$ ➔ $q_2 = (xmax - x1) < 0$ ➔ xmax < x1 ➔ 영역 밖에 있다.

       c. $q_3 < 0$ ➔ $q_3 = (y1 - ymin) < 0$ ➔ y1 < ymin ➔ 영역 밖에 있다.

       d. $q_4 < 0$ ➔ $q_4 = (ymax - y1) < 0$ ➔ ymax < y1 ➔ 영역 밖에 있다.

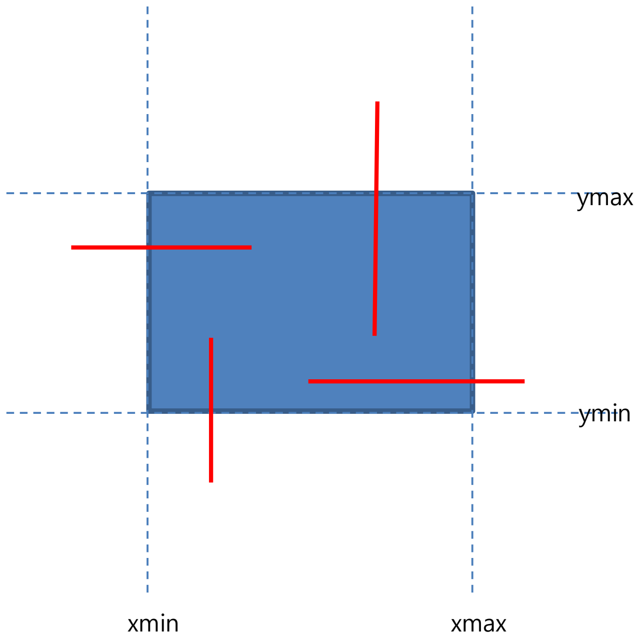
     - p_k == 0 이고 q_k < 0 이면, 영역 밖에 있으므로 안 그린다.



Clipping (클리핑): Liang-barskey 알고리즘

1. p_k == 0 일 때,
   - p_k = 0 이고, q_k > 0 일 때
     a. $q_1 > 0 \rightarrow q_1 = (x1 - xmin) > 0 \rightarrow x1 > xmin$
     b. $q_2 > 0 \rightarrow q_2 = (xmax - x1) > 0 \rightarrow xmax > x1$
     c. $q_3 > 0 \rightarrow q_3 = (y1 - ymin) > 0 \rightarrow y1 > ymin$
     d. $q_4 > 0 \rightarrow q_4 = (ymax - y1) > 0 \rightarrow ymax > y1$

- 평행한 가장자리외의 가장자리와 만날 수 있다.



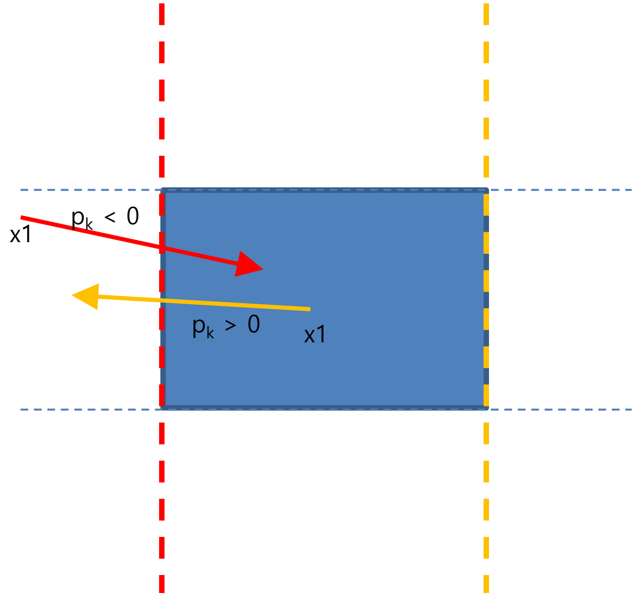
Clipping (클리핑): Liang-barskey 알고리즘

2. p_k != 0 일 때,

- p_k != 0이면, 선분이 경계선 중 하나와 평행하지 않다.

  -> 그 선분의 무한한 연장선은 윈도우의 네 개의 경계선과 어디에선가 교차한다.

- $p_k < 0$:
  - $p_1 < 0$ ➔ $p_1 = -dx = -(x2 - x1) < 0$ ➔ $x2 - x1 > 0$ ➔ $x2 > x1$
    - » $p_1 < 0$이면 $p_2 = dx$ ➔ $p_2 > 0$
  - $p_3 < 0$ ➔ $p_3 = -dy = -(y2 - y1) < 0$ ➔ $y2 - y1 > 0$ ➔ $y2 > y1$
    - » $p_3 < 0$ 이면 $p_4 = dy$ ➔ $p_4 > 0$

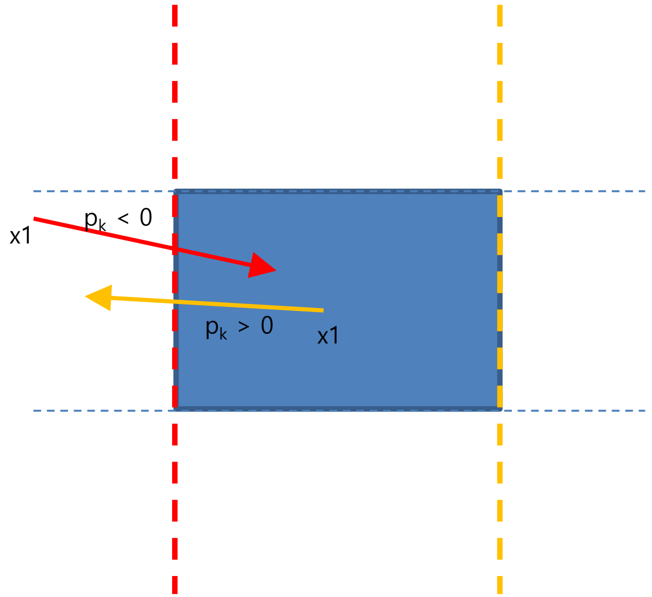


- $p_k > 0$:
  - $p_1 > 0$ ➔ $p_1 = -dx = -(x2 - x1) > 0$ ➔ $x2 - x1 < 0$ ➔ $x2 < x1$
    - » $p_1 > 0$ 이면 $p_2 = dx$ ➔ $p_2 < 0$
  - $p_3 > 0$ ➔ $p_3 = -dy = -(y2 - y1) > 0$ ➔ $y2 - y1 < 0$ ➔ $y2 < y1$
    - » $p_3 > 0$ 이면 $p_4 = dy$ ➔ $p_4 < 0$

Clipping (클리핑): Liang-barskey 알고리즘

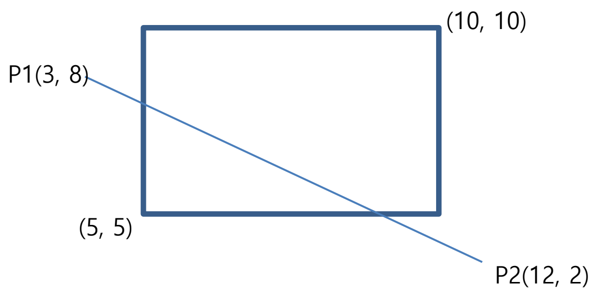
- 만약 $p_k < 0$이면, 직선은 밖➔ 안으로 진행
- 만약 $p_k > 0$이면, 직선은 안 ➔ 밖으로 진행

- 0이 아닌 p_k에 대하여, 매개변수 u의 값으로 가장자리와의 교차점을 찾을 수 있다. 즉,
  - $u_k = q_k / p_k$ ($k = 1, 2, 3, 4$)
    - k 에 대한 u의 값에 대하여,
    - $p_k < 0$ 이면, $u_{start} = \max(u_k, 0)$
    - $p_k > 0$ 이면, $u_{end} = \min(u_k, 1)$
  - 가장자리와의 교차점인 새로운 매개변수 u_{start} 와 u_{end} 에 대해서,
    - if $u_{start} > u_{end}$ ➔ reject
    - if $u_{start} < u_{end}$ ➔ 새로운 좌표값
      - » $new\_x1 = x1 + u_{start} * dx$, $new\_y1 = y1 + u_{start} * dy$
      - » $new\_x2 = x1 + u_{end} * dx$, $new\_y2 = y1 + u_{end} * dy$



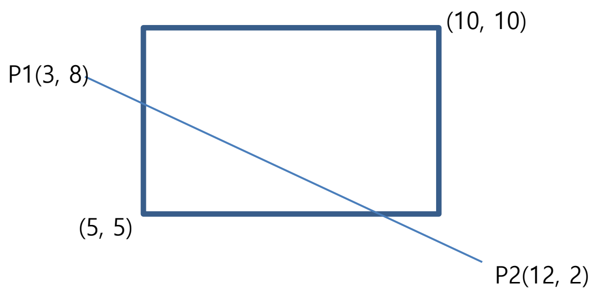
Clipping (클리핑): Liang-barskey 알고리즘

- 예) 윈도우 영역 (5, 5) (10, 10), p1=(3, 8) p2=(12, 2)



Clipping (클리핑): Liang-barskey 알고리즘

- 예) 윈도우 영역 (5, 5) (10, 10), p1=(3, 8) p2=(12, 2)   (계속)



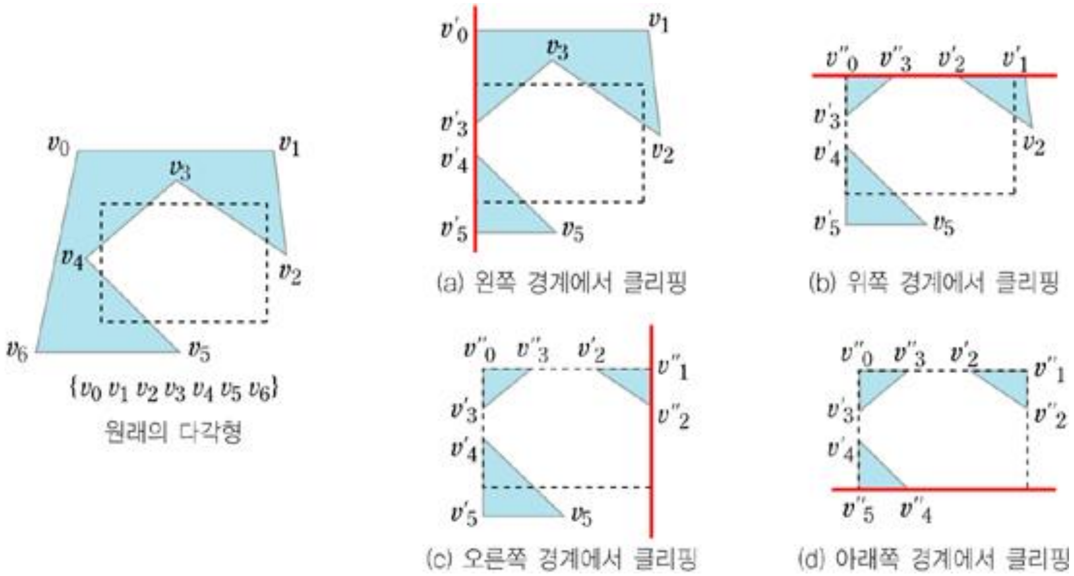
Clipping (클리핑): 다각형

- **속이 빈 다각형(Hollow polygon) :**
  - 선 클리핑 알고리즘 적용

- **속이 찬 다각형 :**
  - 몇 개의 Closed filled polygon 생성

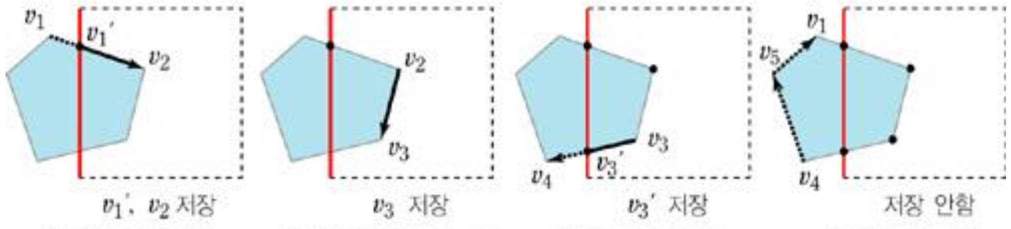
Clipping (클리핑): 다각형

- Sutherland-Hodgeman 알고리즘
  - 다각형의 모든 꼭지점이 윈도우의 내부 또는 외부에 완전히 포함되는지를 결정하여 다각형 전체를 제거하거나 선택하고 그 외의 경우에는 다음 알고리즘을 적용하여 다각형을 클리핑
  - 한 경계변을 기준하여 이 변이 윈도 바깥쪽 영역에 속하는 다각형 부분은 클리핑 소거



Clipping (클리핑): 다각형

- 각 윈도우 경계 (상하좌우)에 대하여 다음 알고리즘을 적용
- 4가지 경우로 구분하여 다각형 꼭지점을 재구성



(a) 외부에서 내부로 (b) 내부에 완전히 포함 (c) 내부에서 외부로 (d) 외부에 포함