1. **Dynamic blending**

## define

|  |
| --- |
| #define TOTAL\_ZONE\_NUM |

[Description]

Quad 영상에서 dynamic blending을 수행하기 위해 사용하는 Zone의 총 개수

VPU에서 지원하는 최대 가용 Zone의 개수는 16개이다.

|  |
| --- |
| #define OD\_ZONE##\_START\_X |

[Description]

사각형 형태의 Zone 영역에서 시작점 X 좌표

##은 Zone의 번호를 가리키며 01~16 까지 존재한다.

|  |
| --- |
| #define OD\_ZONE##\_START\_Y |

[Description]

사각형 형태의 Zone 영역에서 시작점 Y 좌표

##은 Zone의 번호를 가리키며 01~16 까지 존재한다.

|  |
| --- |
| #define OD\_ZONE##\_END\_X |

[Description]

사각형 형태의 Zone 영역에서 끝점 X 좌표

##은 Zone의 번호를 가리키며 01~16 까지 존재한다.

|  |
| --- |
| #define OD\_ZONE##\_END\_Y |

[Description]

사각형 형태의 Zone 영역에서 끝점 Y 좌표

##은 Zone의 번호를 가리키며 01~16 까지 존재한다.

|  |
| --- |
| #define OD\_ZONE##\_MIN\_DIST |

[Description]

개별 Zone 내에서 추출 된 특징점 매칭 쌍 중에서 유효한 매칭 쌍으로 판별하기 위한 최소 거리. 현재 프레임의 좌표와 이전 프레임의 좌표로부터 계산된 벡터의 크기가 MIN\_DIST 보다 커야 유효 특징점 쌍으로 판별. 이 때 특징점 매칭 쌍의 정보(좌표, 개수)는 H/W(VPU)에 의해 구해진다.

##은 Zone의 번호를 가리키며 01~16 까지 존재한다.

|  |
| --- |
| #define OD\_ZONE##\_MAX\_DIST |

[Description]

개별 Zone 내에서 추출 된 특징점 매칭 쌍 중에서 유효한 매칭 쌍으로 판별하기 위한 최대 거리. 현재 프레임의 좌표와 이전 프레임의 좌표로부터 계산된 벡터의 크기가 MAX\_DIST 보다 작아야 유효 특징점 쌍으로 판별. 이 때 특징점 매칭 쌍의 정보(좌표, 개수)는 H/W(VPU)에 의해 구해진다.

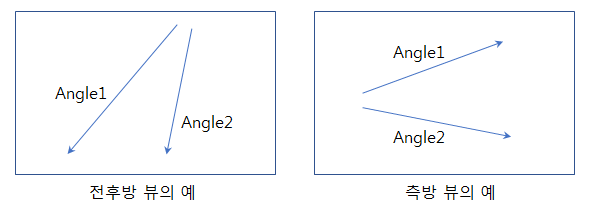
##은 Zone의 번호를 가리키며 01~16 까지 존재한다.

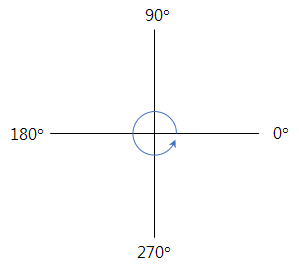
|  |
| --- |
| #define OD\_ZONE##\_ANGLE1  #define OD\_ZONE##\_ANGLE2 |

[Description]

개별 Zone 내에서 추출 된 특징점 매칭 쌍 중에서 접근하는 물체임을 판단하기 위한 기준으로 두 각을 지정하고 그 사이에 존재하는 벡터의 방향을 유효한 특징점 쌍으로 판별. 이 때 기준으로 사용되는 각 중의 하나이다. Angle2 와의 관계에 있어 지정 순서 등에 있어서 제약은 없으며 둘 중 하나를 Angle1 이라 하면 나머지 하나는 Angle2가 된다.

##은 Zone의 번호를 가리키며 01~16 까지 존재한다.





## Structure

### Point\_t

[Syntax]

|  |
| --- |
| typedef struct Point\_t\_  {  uint16\_t x;  uint16\_t y;  }\_Point\_t |

[Description]

점의 좌표를 위한 structure

[Member]

|  |  |
| --- | --- |
| Member | Description |
| x | Point X 좌표 |
| y | Point Y 좌표 |

### OptFlowPairPos\_t

[Syntax]

|  |
| --- |
| typedef struct OptFlowPairPos\_t\_  {  Point\_t first;  Point\_t second;  }\_ OptFlowPairPos\_t |

[Description]

Optical Flow 에서 매칭되는 특징점 쌍의 좌표를 위한 structure

[Member]

|  |  |
| --- | --- |
| Member | Description |
| first | 첫 번째 프레임의 특징점 좌표(X,Y) |
| second | 두 번째 프레임의 특징점 좌표(X,Y) |

### ObjectDetectParam\_t

[Syntax]

|  |
| --- |
| typedef struct ObjectDetectParam\_t\_  {  uint16\_t roiStartX;  uint16\_t roiStartY;  uint16\_t roiWidth;  uint16\_t roiHeight;  uint16\_t minDist;  uint16\_t maxDist;  uint16\_t angle1;  uint16\_t angle2;  }\_ObjectDetectParam\_t |

[Description]

Object Detection 수행에 필요한 parameter 정보를 위한 structure

[Member]

|  |  |
| --- | --- |
| Member | Description |
| roiStartX | Zone의 시작 X 좌표 |
| roiStartY | Zone의 시작 Y 좌표 |
| roiWidth | Zone의 Width |
| roiHeight | Zone의 Height |
| minDist | Zone 내에서 유효한 optical flow 쌍으로 판단하기 위한 최소 거리 |
| maxDist | Zone 내에서 유효한 optical flow 쌍으로 판단하기 위한 최대 거리 |
| angle1 | Zone 내에서 유효한 optical flow 쌍으로 판단하기 위한 기준각 1 |
| angle2 | Zone 내에서 유효한 optical flow 쌍으로 판단하기 위한 기준각 2 |

### BoundBoxRect\_t\_

[Syntax]

|  |
| --- |
| typedef struct BoundBoxRect\_t\_  {  Point\_t start;  Point\_t end;  }\_BoundBoxRect\_t |

[Description]

Bounding box (Rectagle) 의 시작점과 끝점을 위한 structure

[Member]

|  |  |
| --- | --- |
| Member | Description |
| start | Bounding box의 시작 좌표(X,Y) |
| end | Bounding box의 끝 좌표(X,Y) |

## API

### ObjectDetector\_InitParam

[Syntax]

|  |
| --- |
| void ObjectDetector\_InitParam(ObjectDetectParam\_t \* odParam); |

[Description]

Object detection을 수행하기 위한 parameter를 설정한다. Main 에서 초기에 한 번 실행한다. 16개의 Zone에 대해 define 된 값으로 zone 영역, MIN/MAX distance, Angle1/2이 설정된다.

[Parameter]

|  |  |
| --- | --- |
| Member | Description |
| odParam | ObjectDetectParam\_t 구조체로 Zone 개수만큼 값을 저장한다. 변수 선언 시 Zone 개수만큼 배열로 선언  ex) ObjectDetectParam\_t *variable* [TOTAL\_ZONE\_NUM] |

### ObjectDetector

[Syntax]

|  |
| --- |
| void ObjectDetector (OptFlowPairPos\_t \*\* inMotionPair, uint16\_t \* numMotion, OptFlowPairPos\_t (\*prevMotionPair)[100], uint16\_t \* prevNumMotion, ObjectDetectParam\_t \* odParam, uint8\_t \* statusNear, uint8\_t \* statusAway); |

[Description]

VPU의 Hamming Accelerator 결과 (특징점 매칭쌍의 좌표, 개수)를 인자로 받아들이고 ObjectDetector\_InitParam에 의해 설정된 파라미터들을 받아서 각 Zone의 Approaching or Leaving object의 상태를 내보낸다. 이 때 시간축 상의 매칭 쌍 filtering을 위해 함수 내에서의 처리 결과를 다음 프레임 처리 시 사용하기 위해 prevMotionPair와 prevNumMotion으로 저장한다.

[Parameter]

|  |  |
| --- | --- |
| Member | Description |
| inMotionPair | Zone 내에서의 특징점 매칭 좌표값.  이전 프레임의 특징점 좌표(X,Y), 현재 프레임의 특징점 좌표(X,Y)  Zone의 개수와 각 Zone에서의 특징점 매칭 개수만큼 필요  ex) OptFlowPairPos\_t \* *variable* [TOTAL\_ZONE\_NUM] |
| numMotion | Zone 내에서의 특징점 매칭 개수  Zone 의 개수만큼 필요  ex) uint16\_t *variable* [TOTAL\_ZONE\_NUM] |
| prevMotionPair | 입력된 특징점 매칭 좌표(inMotionPair)로부터 유효한 정보로 판단되어 걸러진 매칭 좌표 값. 다음 프레임 입력 시 사용하기 위해 저장됨  Zone 의 개수만큼 필요하고 Zone 별 100개로 제한  ex) OptFlowPairPos\_t *variable* [TOTAL\_ZONE\_NUM][100] |
| prevNumMotion | 입력된 특징점 매칭 개수(numMotion)로부터 유효한 정보로 판단되어 걸러진 매칭 좌표 개수. 다음 프레임 입력 시 사용하기 위해 저장됨  Zone의 개수만큼 필요  ex) uint16\_t *variable* [TOTAL\_ZONE\_NUM] |
| odParam | ObjectDetector\_InitParam 함수에서 전달된 parameter 값 |
| statusNear | Zone에서의 approaching object detect 상태  0 : No detection  1 : Weak detection  2 : Strong detection  Zone의 개수만큼 필요  ex) uint8\_t *variable* [TOTAL\_ZONE\_NUM] |
| statusAway | Zone에서의 leaving object detect 상태  0 : No detection  1 : Weak detection  2 : Strong detection  Zone의 개수만큼 필요  ex) uint8\_t *variable* [TOTAL\_ZONE\_NUM] |

### odBlendCorner

[Syntax]

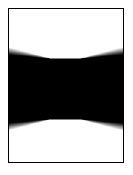
|  |
| --- |
| void odBlendCorner (uint8\_t dynBlendEn, uint8\_t \* statusNear, uint8\_t \* statusAway, uint8\_t cntCornerNear, uint8\_t \* cntCornerAway, uint8\_t \* blendCorner); |

[Description]

각 Zone의 approaching or leaving object의 detection 상태를 전달받아서 최종적으로 Top View의 네 코너(Top/Left, Top/Right, Bottom/Left, Bottom/Right)의 alpha mode를 출력한다. 이 때 시간 축 상으로 detection 상태를 누적하여 관리하기 위해 approaching과 leaving 상태 각각에 대해 counter를 두어 저장한다.

[Parameter]

|  |  |
| --- | --- |
| Member | Description |
| dynBlendEn | Dynamic blend enable/disable  0: Disable, 1: Enable |
| statusNear | ObjectDetector 에서 전달된 approaching object detection 상태  Zone 의 개수만큼 필요  ex) uint16\_t *variable* [TOTAL\_ZONE\_NUM] |
| statusAway | ObjectDetector 에서 전달된 leaving object detection 상태  Zone 의 개수만큼 필요  ex) uint16\_t *variable* [TOTAL\_ZONE\_NUM] |
| cntCornerNear | statusNear의 상태값에 따라 counter 값이 증감한다.  Dynamic blending 이 동작하는 동안 계속 누적되어 동작한다.  Zone의 개수만큼 필요  ex) uint8\_t *variable* [TOTAL\_ZONE\_NUM] |
| cntCornerAway | statusAway의 상태값에 따라 counter 값이 증감한다.  Dynamic blending 이 동작하는 동안 계속 누적되어 동작한다.  Zone의 개수만큼 필요  ex) uint8\_t *variable* [TOTAL\_ZONE\_NUM] |
| blendCorner | Top View 네 코너에서의 alpha mode  0 : Vertical Major  255 : Horizontal Major  네 코너의 개수만큼 필요  ex) uint8\_t *variable* [4] |

< blendCorner = 255 > < blendCorner = 0 >