Virtual Gate

注意

您購買的產品、服務或特性等應受晨星半導體有限公司商業合同和條款的約束，本文檔中描述的全部或部分產品，服務或特性可能不在您的購買或使用範圍之內。除非合同另有約定，本文檔僅作為使用指導，本文檔中的所有陳述，資訊和建議不構成任何明示或暗示的擔保。

晨星半導體股份有限公司

地址：新竹縣竹北市台元街 26 號 4 樓之 1

電話：03-5526006

REVISION HISTORY

Revision No.	Description	Date
1.0	• Created.	11/29/2016
2.0	• Fix parameter range problem	03/21/2017
3.0	• Modify the structure and introduction	04/13/2017
4.0	• Modify the return value	04/26/2017

前言

本文為使用 VG 進行開發的程式師而寫，目的是供您在開發過程中查閱 VG 軟體發展包的各種參考資訊。

本文檔描述 VG 軟體的各個功能，以及相關資料結構和錯誤碼。

讀者對象

本文檔主要適用於以下工程師

- 技術支援工程師。
- 軟體發展工程師。

1 概述

電子圍欄（Virtue Fence）用在獲取的攝像頭視頻中設置圍欄，當有物體穿越設置的圍欄時觸發警報信號，該功能被實際應用在安防監控、視頻儲存紀錄等。

2 API 參考

2.1 函數概述

●MI_VG_Init：初始化 VG 庫。
●MI_VG_Uninit：退出 VG 庫，釋放資源。
●MI_VG_SetScene：設置 VG 應用場景。
●MI_VG_GetScene：獲取設置的 VG 應用場景配置參數。
●MI_VG_SetLineNumber：設置 VG 線段數目。
●MI_VG_GetLineNumber：獲取設置的 VG 線段配置數目。
●MI_VG_SetLineAndDir：設置 VG 線段參數。
●MI_VG_GetLineAndDir：獲取 VG 線段配置參數。
●MI_VG_SetObjSizeThd：設置 VG 檢測物體閥值[百分比值]。
●MI_VG_GetObjSizeThd：獲取 VG 檢測物體閥值[百分比值]。
●MI_VG_Run：運行 VG 庫。
●MI_VG_GetResult：獲取 VG 檢測結果。
●MI_VG_GetDebugInfo：獲取設置 VG 參數。

2.2 函數功能參考

MI_VG_Init

【描述】
初始化 VG 庫。

【語法】
MI_VG_HANDLE MI_VG_Init(MI_VgSet_t* vg_user_info, uint16_t width, uint16_t height);

【參數】

參數描述	說明
vg_user_info	輸入 VgSet 結構體指標
width	輸入源寬
height	輸入源高

【返回值】

參數描述	說明
MI_VG_HANDLE	非 0：成功
NULL	0：失敗

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

1， vg_user_info 結構體要先申請記憶體和配置參數。

2， 減少 VG 演算法的 cpu 消耗，可降低輸入源圖像的解析度，源圖像的寬高推薦值為 320、180。

【舉例】

無。

MI_VG_Uninit

【說明】

退出 VG 庫，釋放資源。

【語法】

void MI_VG_Uninit(MI_VG_HANDLE vg_handle);

【參數】

參數描述	說明
vg_handle	VG 的 handle

【返回值】

None

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

無。

【舉例】

無。

MI_VG_SetScene

【描述】

設置 VgSet 結構體中 indoor 成員值。

【語法】

MI_VG_RET MI_VG_SetScene(MI_VgSet_t * vg_user_info, int8_t scene);

【參數】

參數描述	說明
vg_user_info	輸入 VgSet 結構體指標
scene	輸入場景代表參數

【返回值】

參數描述	說明
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INVALID_USER_INFO_POINTER	使用者資訊的指標錯誤
MI_VG_RET_INVALID_ENVIRONMENT_STATE	參數”環境狀態”錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

scene 的值必須為 0 或 1。

【舉例】

無。

MI_VG_GetScene

【描述】

獲取 VgSet 結構體中 indoor 成員值。

【語法】

MI_VG_RET MI_VG_GetScene(MI_VgSet_t * vg_user_info, int8_t* scene);

【參數】

參數描述	說明
vg_user_info	輸入 VgSet 結構體指標
scene	Unsigned char 型別 (用來獲得 indoor 參數訊息)

【返回值】

參數描述	說明
MI_VG_RET_SUCCESS	成功

MI_VG_RET_INVALID_USER_INFO_POINTER	使用者資訊的指標錯誤
MI_VG_RET_INVALID_ENVIRONMENT_POINTER	環境指標錯誤
MI_VG_RET_INVALID_ENVIRONMENT_STATE	參數”環境狀態”錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

scene 必須先申請內存

【舉例】

無。

MI_VG_SetLineNumber

【描述】

設置 VgSet 結構體中 line_number 成員值。

【語法】

MI_VG_RET MI_VG_SetLineNumber(MI_VgSet_t * vg_user_info, uint16_t lineno);

【參數】

參數描述	說明
vg_user_info	輸入 VgSet 結構體指標
lineno	輸入 line_number 成員配置值

【返回值】

參數描述	說明
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INVALID_USER_INFO_POINTER	使用者資訊的指標錯誤
MI_VG_RET_INVALID_LINE_NUMBER	參數”線段數目”錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

lineno 的值必須為 1 或 2。

【舉例】

無。

MI_VG_GetLineNumber

【描述】

獲取 VgSet 結構體中 line_number 成員值。

【語法】

MI_VG_RET MI_VG_GetLineNumber(MI_VgSet_t * vg_user_info, uint16_t* lineno);

【參數】

參數描述	說明
vg_user_info	輸入 VgSet 結構體指標
lineno	Unsigned short 型別 (用來獲得 line_number 參數的值)

【返回值】

參數描述	說明
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INVALID_USER_INFO_POINTER	使用者資訊的指標錯誤
MI_VG_RET_INVALID_LINE_POINTER	線段指標錯誤
MI_VG_RET_INVALID_LINE_NUMBER	參數"線段數目"錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

lineno 必須先申請內存。

【舉例】

無。

MI_VG_SetLineAndDir

【描述】

設置 VgSet 結構體中指定 Line 屬性值。

【語法】

MI_VG_RET MI_VG_SetLineAndDir(MI_VgSet_t * vg_user_info, MI_VgLine_t *

line_coordinate, uint16_t lineno);

【參數】

參數描述	說明
vg_user_info	輸入 VgSet 結構體指標
line_coordinate	輸入 Line 的屬性值（1 點座標，2 點座標，1 點方向座標和 2 點方向座標）
lineno	輸入要設置的 Line 的索引值

【返回值】

參數描述	說明
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INVALID_USER_INFO_POINTER	使用者資訊的指標錯誤
MI_VG_RET_INVALID_COORDINATE_POINTER	線段座標的指標錯誤
MI_VG_RET_INVALID_LINE_NUMBER	參數”線段數目”錯誤
MI_VG_RET_INVALID_FIRST_LINE_INFO	第一條虛擬線段資訊錯誤
MI_VG_RET_INVALID_SECOND_LINE_INFO	第二條虛擬線段資訊錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

1， lineno 的取值範圍為 1 or 2；

2， line1 和 line2 不能交叉。

3， 減少 VG 演算法的 cpu 消耗，設置的線段長度建議小於輸入影像寬度的一半。

【舉例】

無。

MI_VG_GetLineAndDir

【描述】

獲取 VgSet 結構體中指定 Line 屬性值。

【語法】

MI_VG_RET MI_VG_GetLineAndDir(MI_VgSet_t * vg_user_info, MI_VgLine_t * line_coordinate, uint16_t lineno);

【參數】

參數描述	說明

vg_user_info	輸入 VgSet 結構體指標
line_coordinate	輸入用於保存獲取 Line 的屬性值
lineno	輸入要獲取的 Line 的值

【返回值】

參數描述	說明
MI_RET_SUCESS	0：成功
MI_VG_RET_INVALID_USER_INFO_POINTER	使用者資訊的指標錯誤
MI_VG_RET_INVALID_COORDINATE_POINTER	線段座標的指標錯誤
MI_VG_RET_INVALID_LINE_NUMBER	參數”線段數目”錯誤
MI_VG_RET_INVALID_FIRST_LINE_INFO	第一條虛擬線段資訊錯誤
MI_VG_RET_INVALID_SECOND_LINE_INFO	第二條虛擬線段資訊錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

lineno 必須先申請內存。

【舉例】

無。

MI_VG_SetObjSizeThd

【描述】

設置 VgSet 結構體中 line_number 成員值。

【語法】

MI_VG_RET MI_VG_SetObjSizeThd(MI_VgSet_t * vg_user_info, uint16_t size_thd);

【參數】

參數描述	說明
vg_user_info	輸入 VgSet 結構體指標
size_thd	輸入 size_thd 成員配置值

【返回值】

參數描述	說明
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INVALID_USER_INFO_POINTER	使用者資訊的指標錯誤
MI_VG_RET_INVALID_THRESHOLD	閥值參數錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

size_thd 的值必須在 0~100 範圍內。

【舉例】

無。

MI_VG_GetObjSizeThd

【描述】

獲取 VgSet 結構體中 object_size_thd 成員值。

【語法】

MI_VG_RET MI_VG_GetObjSizeThd(MI_VgSet_t * vg_user_info, uint16_t* size_thd);

【參數】

參數描述	說明
vg_user_info	輸入 VgSet 結構體指標
size_thd	Unsigned short 型別 (用來獲得 size_thd 參數資訊)

【返回值】

參數描述	說明
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INVALID_USER_INFO_POINTER	使用者資訊的指標錯誤
MI_VG_RET_INVALID_THRESHOLD_POINTER	閥值指標錯誤
MI_VG_RET_INVALID_THRESHOLD	閥值參數錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

size_thd 必須先申請內存

【舉例】

無。

MI_VG_Run

【說明】

運行 VG 庫。

【語法】

MI_VG_RET MI_VG_Run(MI_VG_HANDLE vg_handle, uint8_t* _ucMask);

【參數】

參數描述	說明
vg_handle	輸入 VG 的 handle 控制碼
ucMask	輸入 Yuv 幀 y 分量的 buffer 位址

【返回值】

返回值	說明
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INVALID_HANDLE	VG handle 錯誤
MI_VG_RET_INVALID_INPUT_POINTER	輸入影像的指標錯誤
MI_VG_RET_OPERATE_ERROR	VG 執行錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

減少 VG 演算法的 cpu 使用，可降低輸入 VG 演算法的 YUV 幀率，幀率推薦值為 10~15

【舉例】

無。

MI_VG_GetResult

【說明】

獲取 VG 運行結果。

【語法】

MI_VG_RET MI_VG_GetResult(MI_VG_HANDLE vg_handle, VgResult *cross_alarm);

【參數】

參數描述	說明

vg_handle	輸入 VG 的 handle 控制碼
cross_alarm	輸入 VgResult 結構體指標

【返回值】

返回值	說明
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INVALID_HANDLE	VG handle 錯誤
MI_VG_RET_INVALID_ALARM_POINTER	警報指標錯誤
MI_VG_RET_OPERATE_ERROR	VG 執行錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

cross_alarm 指標要先分配記憶體。

【舉例】

無。

MI_VG_GetDebugInfo

【說明】

獲取 VG 中設置 VgSet 資料結構資訊。

【語法】

MI_VG_RET MI_VG_GetDebugInfo(MI_VG_HANDLE vg_handle, VgDebug *debug_info);

【參數】

參數描述	說明
vg_handle	輸入 VG 的 handle 控制碼
debug_info	輸入 VgDebug 結構體指標

【返回值】

參數描述	說明
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INVALID_HANDLE	VG handle 錯誤
MI_VG_RET_INVALID_DEBUG_POINTER	Debug 指標錯誤
MI_VG_RET_OPERATE_ERROR	VG 執行錯誤

【需求】

●標頭檔：mi_vg.h

●庫文件：libVG_LINUX.a 、libVG_LINUX.so

【注意】

debug_info 指標要先分配記憶體。

【舉例】

無。

3 資料類型

MI_VG_RET	回傳資訊結構體
MI_VG_HANDLE	VG handle
MI_VG_Point_t	VG 線段資訊結構體
MI_VG_Line_t	VG line 屬性參數結構體
MI_VgSet_t	VG 參數結構體
MI_VgResult_t	VG 運行結果返回結構體
MI_VgDebug_t	VG debug 參數結構體

MI_VG_RET

【說明】

回傳資訊結構體

【定義】

```
typedef enum _MI_RET_E
{
    MI_VG_RET_SUCCESS                       = 0x00000000,
    MI_VG_RET_INIT_ERROR                    = 0x10000301,
    MI_VG_RET_IC_CHECK_ERROR                = 0x10000302,
    MI_VG_RET_INVALID_HANDLE                = 0x10000303,
    MI_VG_RET_INVALID_USER_INFO_POINTER     = 0x10000304,
    MI_VG_RET_INVALID_ENVIRONMENT_STATE     = 0x10000305,
    MI_VG_RET_INVALID_ENVIRONMENT_POINTER   = 0x10000306,
    MI_VG_RET_INVALID_LINE_NUMBER           = 0x10000307,
    MI_VG_RET_INVALID_LINE_POINTER          = 0x10000308,
    MI_VG_RET_INVALID_COORDINATE_POINTER    = 0x10000309,
    MI_VG_RET_INVALID_FIRST_LINE_INFO       = 0x1000030A,
    MI_VG_RET_INVALID_SECOND_LINE_INFO      = 0x1000030B,
```

```
    MI_VG_RET_INVALID_THRESHOLD                 = 0x1000030C,
    MI_VG_RET_INVALID_THRESHOLD_POINTER         = 0x1000030D,
    MI_VG_RET_INVALID_INPUT_POINTER             = 0x1000030E,
    MI_VG_RET_OPERATE_ERROR                     = 0x1000030F,
    MI_VG_RET_INVALID_ALARM_POINTER             = 0x10000310,
    MI_VG_RET_INVALID_DEBUG_POINTER             = 0x10000311,
} MI_RET;
```

【參數】

Member	Description
MI_VG_RET_SUCCESS	成功
MI_VG_RET_INIT_ERROR	VG 初始化錯誤
MI_VG_RET_IC_CHECK_ERROR	VG platform 檢測錯誤
MI_VG_RET_INVALID_HANDLE	VG handle 錯誤
MI_VG_RET_INVALID_USER_INFO_POINTER	使用者資訊的指標錯誤
MI_VG_RET_INVALID_ENVIRONMENT_STATE	參數”環境狀態”錯誤
MI_VG_RET_INVALID_ENVIRONMENT_POINTER	環境指標錯誤
MI_VG_RET_INVALID_LINE_NUMBER	參數”線段數目”錯誤
MI_VG_RET_INVALID_LINE_POINTER	線段指標錯誤
MI_VG_RET_INVALID_COORDINATE_POINTER	線段座標的指標錯誤
MI_VG_RET_INVALID_FIRST_LINE_INFO	第一條虛擬線段資訊錯誤
MI_VG_RET_INVALID_SECOND_LINE_INFO	第二條虛擬線段資訊錯誤
MI_VG_RET_INVALID_THRESHOLD	閥值參數錯誤
MI_VG_RET_INVALID_THRESHOLD_POINTER	閥值指標錯誤
MI_VG_RET_INVALID_INPUT_POINTER	輸入影像的指標錯誤
MI_VG_RET_OPERATE_ERROR	VG 執行錯誤
MI_VG_RET_INVALID_ALARM_POINTER	警報指標錯誤
MI_VG_RET_INVALID_DEBUG_POINTER	Debug 指標錯誤

MI_VG_HANDLE

【說明】

VG 控制碼結構體。

【定義】

```
typedef void* MI_VG_HANDLE;
```

【參數】

無

MI_VG_Point_t

【說明】

VG 線段資訊結構體。

【定義】

```
typedef struct _MI_VG_Point_t
{
    int32_t x;
    int32_t y;
} MI_VG_Point_t;
```

【參數】

參數	說明
x	x 標點
y	y 標點

MI_VG_Line_t

【說明】

VG line 屬性參數結構體。

【定義】

```
typedef struct _MI_VG_Line_t
{
    VG_Point_t  px;
    VG_Point_t  py;
    VG_Point_t  pdx;
    VG_Point_t  pdy;
} MI_VG_Line_t;
```

【參數】

參數	說明
px	第一座標點
py	第二座標點
pdx	第一方向座標點
pdy	第二方向座標點

MI_VgSet_t

【說明】

VG 參數結構體。

【定義】

```
typedef struct _MI_VgSet_t
{
    uint16_t    object_size_thd;
    uint16_t    line_number;
    uint8_t     indoor;
    uint16_t    fx;
    uint16_t    fy;
    uint16_t    sx;
    uint16_t    sy;
    uint16_t    fdx;
    uint16_t    fdy;
    uint16_t    sdx;
    uint16_t    sdy;
    uint16_t    sfx;
    uint16_t    sfy;
    uint16_t    ssx;
    uint16_t    ssy;
    uint16_t    sfdx;
    uint16_t    sfdy;
    uint16_t    ssdx;
    uint16_t    ssdy;
} MI_VgSet_t;
```

【參數】

參數	說明
object_size_thd	檢測物體大小的閥值(影像大小的百分比).
Line_number	虛擬線段數目
indoor	環境參數 (0= 戶外 ;1= 室內).
fx	第一條虛擬線段中第一個線段點的 x 座標
fy	第一條虛擬線段中第一個線段點的 y 座標
sx	第一條虛擬線段中第二個線段點的 x 座標
sy	第一條虛擬線段中第二個線段點的 y 座標
fdx	第一條虛擬線段中第一個方向點的 x 座標
fdy	第一條虛擬線段中第一個方向點的 y 座標
sdx	第一條虛擬線段中第二個方向點的 x 座標
sdy	第一條虛擬線段中第二個方向點的 y 座標

sfx	第二條虛擬線段中第一個線段點的 x 座標
sfy	第二條虛擬線段中第一個線段點的 y 座標
ssx	第二條虛擬線段中第二個線段點的 x 座標
ssy	第二條虛擬線段中第二個線段點的 y 座標
sfdx	第二條虛擬線段中第一個方向點的 x 座標
sfdy	第二條虛擬線段中第一個方向點的 y 座標
ssdx	第二條虛擬線段中第二個方向點的 x 座標
ssdy	第二條虛擬線段中第二個方向點的 y 座標

【注意】

1， 警報方向是由方向點決定的，當物件由第一方向點的那一側移至第二方向點的那一側，就會觸發警報。

2， 如果第一和第二方向點的座標完全相同，表示沒有方向，只要物體穿越警報線則觸發警報。

3， 如果第一和第二方向點位於警報線的同一側，表示沒有方向，只要物體穿越警報線則觸發警報。

4， 如果第一和第二方向點皆位於警報線上，表示沒有方向，只要物體穿越警報線則觸發警。

5， 警報線段長度建議小於輸入影像寬的一半。

MI_VgResult_t

【說明】

VG 結果返回結構體。

【定義】

```
typedef struct _MI_VgResult_t
{
    uint16_t    alarm1;
    uint16_t    alarm2;
} MI_VgResult_t;
```

【參數】

參數	說明
alarm1	第一條虛擬線段的警報狀態,，0 = 正常， 1 = 警報
alarm2	第二條虛擬線段的警報狀態,，0 = 正常， 1 = 警報

MI_VgDebug_t

【說明】

VG 調試參數結構體。

【定義】

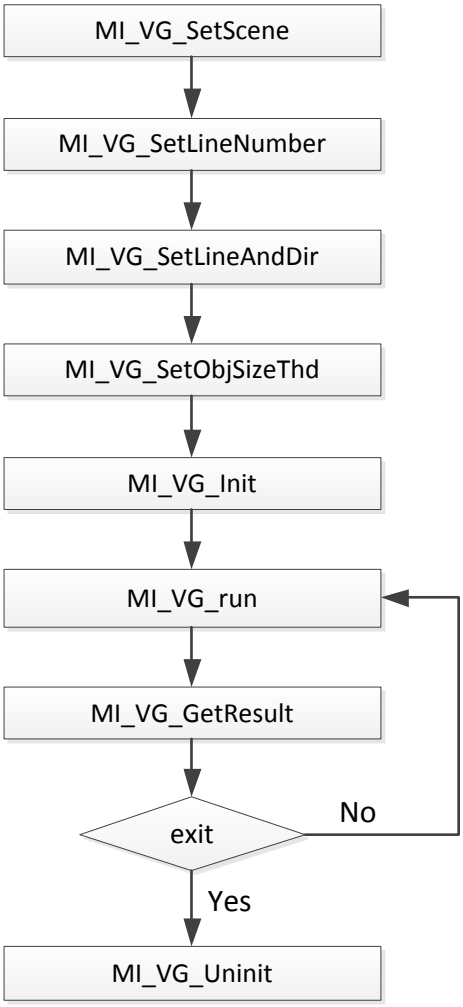
```
typedef struct _MI_VgDebug_t
{
    uint16_t    background_state;
    uint32_t    version;
    uint32_t    debug_object_size;
    uint32_t    debug_state;
    uint16_t    debug_fsp_x;
    uint16_t    debug_fsp_y;
    uint16_t    debug_fep_x;
    uint16_t    debug_fep_y;
    uint16_t    debug_fx;
    uint16_t    debug_fy;
    uint16_t    debug_sx;
    uint16_t    debug_sy;
    uint16_t    debug_fdx;
    uint16_t    debug_fdy;
    uint16_t    debug_sdx;
    uint16_t    debug_sdy;
    uint16_t    debug_ssp_x;
    uint16_t    debug_ssp_y;
    uint16_t    debug_sep_x;
    uint16_t    debug_sep_y;
    uint16_t    debug_sfx;
    uint16_t    debug_sfy;
    uint16_t    debug_ssx;
    uint16_t    debug_ssy;
    uint16_t    debug_sfdx;
    uint16_t    debug_fdy;
    uint16_t    debug_ssdx;
    uint16_t    debug_ssdy;
} MI_VgDebug_t;
```

【參數】

參數	說明
background_state	背景訓練結果 (0：成功 ；1：失敗)
version	Library 版本
debug_object_size	檢測物體大小閥值
debug_state	Debug 模式狀態. (0：關閉 ；1：打開)
debug_fsp_x	第一塊感興趣區域之起點 x 座標
debug_fsp_y	第一塊感興趣區域之起點 y 座標
debug_fep_x	第一塊感興趣區域之終點 x 座標
debug_fep_y	第一塊感興趣區域之終點 y 座標

debug_fx	第一條虛擬線段中第一個線段點的 x 座標
debug_fy	第一條虛擬線段中第一個線段點的 y 座標
debug_sx	第一條虛擬線段中第二個線段點的 x 座標
debug_sy	第一條虛擬線段中第二個線段點的 y 座標
debug_fdx	第一條虛擬線段中第一個方向點的 x 座標
debug_fdy	第一條虛擬線段中第一個方向點的 y 座標
debug_sdx	第一條虛擬線段中第二個方向點的 x 座標
debug_sdy	第一條虛擬線段中第二個方向點的 y 座標
debug_ssp_x	第二塊感興趣區域之起點 x 座標
debug_ssp_y	第二塊感興趣區域之起點 y 座標
debug_sep_x	第二塊感興趣區域之終點 x 座標
debug_sep_y	第二塊感興趣區域之終點 y 座標
debug_sfx	第二條虛擬線段中第一個線段點的 x 座標
debug_sfy	第二條虛擬線段中第一個線段點的 y 座標
debug_ssx	第二條虛擬線段中第二個線段點的 x 座標
debug_ssy	第二條虛擬線段中第二個線段點的 y 座標
debug_sfdx	第二條虛擬線段中第一個方向點的 x 座標
debug_sfdy	第二條虛擬線段中第一個方向點的 y 座標
debug_ssdx	第二條虛擬線段中第二個方向點的 x 座標
debug_ssdy	第二條虛擬線段中第二個方向點的 y 座標

4 流程



5 舉例

Sample code 請參考 VG_Sample_Code.c