**Модуль анализа видеоинформации**

**для железной дороги.**

**Назначение модуля.**

Выполняет слежение за участком железной дороги, обнаруживает наличие поезда, находит номер на железнодорожном вагоне, считывает номер, выполняет подсчет количества вагонов, скорость и направление их движения.

**Версия:**

1.0.0 (beta)

**Экспортируемые функции.**

trainsInit

TRAINSHANDLE trainsInit(TVARect& zone, double sens, int scale)

**Назначение:**

Создает экземпляр модуля поиска и считывания номера вагона в соответствии с заданными параметрами наблюдения.

**Параметры:**

**[in]**

TVARect& zone – область наблюдения для считывания номера. Область наблюдения должна быть больше размеров номера не менее чем в 2 раза.

double sens – в текущей версии не используется.

int scale - в текущей версии не используется.

**Возвращаемое значение:**

Дескриптор модуля, в случае успеха. NULL в противном случае.

**Примечание:**

Модуль инициализируется прямоугольником, в котором будет выполняться поиск, сопровождение, считывание номера.

trainClose

int trainsClose(TRAINSHANDLE trains)

**Назначение:**

Освобождает память, распределенную функцией trainsInit и уничтожает модуль, созданный этой функцией.

Параметры:

**[in]**

TRAINSHANDLE trains – дескриптор модуля, который получен с использованием функции trainsInit.

**Возвращаемое значение:**

В случае успеха возвращает VA\_OK в противном случае VA\_ERROR

**Примечание:**

-

trainsProcess

int trainsProcess(TRAINSHANDLE trains, unsigned char\* data, int width, int height, int bpp, char\* result, TVARect& rect)

**Назначение:**

Выполняет анализ изображения.

Параметры:

**[in]**

TRAINSHANDLE trains – дескриптор модуля, который получен с использованием функции trainsInit.

int width - ширина входного изображения.

int height – высота входного изображения.

int bpp - число БАЙТ на один пиксель изображения.

unsigned char\* data – указатель на массив пикселей изображения.

**[out]**

char\* result – номер вагона

TVARect& rect – область найденного номера

**Возвращаемое значение:**

В случае успеха возвращает VA\_OK в противном случае VA\_ERROR

**Примечание:**

* Если номер не найден, то переменная rect не изменяется. Переменная result по умолчанию заполняется символами # и в тех местах, где символ распознан # заменяется на значение цифры. Количество символов в переменной result 8, 9й символ устанавливается в значение \0.
* Эта функция должна определять факт наличия поезда и направление его движения. Для этого предполагается изменить формат вызова trainsInit и trainsProcess

trainsTargetInit

TARGETHANDLE trainsTargetInit(TVAInitParams& params);

**Назначение:**

Создает экземпляр модуля подсчета числа вагонов, в соответствии с заданными параметрами.

**Параметры:**

**[in]**

TVAInitParams\* params – параметры наблюдения.

|  |  |
| --- | --- |
| Параметр | Значение |
| TVAInitParams.Camera | Игнорируется |
| TVAInitParams.SaveLog | Игнорируется |
| TVAInitParams.Path | Используется для указания пути записи образа мишени в виде математической модели. |
| TVAInitParams.EventSens | Задает чувствительность детектора мишени. Должен принимать значения в диапазоне от 0 до 1 |
| TVAInitParams.EventTimeSens | Игнорируется |
| TVAInitParams.minSquare | Игнорируется |
| TVAInitParams.maxSquare | Игнорируется |
| TVAInitParams.NumZones | Задает число зон. ДОЛЖЕН БЫТЬ РАВЕН 1. |
| TVAInitParams.Zones | Указатель на массив зон наблюдения. Должен содержать одну зону, которая описывает мишень, установленную за железнодорожным полотном. См. примечание. |
| TVAInitParams.useTrack | Игнорируется |

**Возвращаемое значение:**

Дескриптор модуля, в случае успеха. NULL в противном случае.

**Примечание:**

Модуль инициализируется прямоугольником, который описывает мишень, находящуюся за железнодорожным полотном. Мишень должна быть полностью видна в промежутке между любыми вагонами (цистернами, грузовыми, хопперами, пассажирскими, платформами и так далее) и любой из вагонов должен перекрывать ее. Размер мишени на изображении с HD видеокамеры не должен быть меньше 96 пикселей.

trainsTargetCreate

TRAINS\_API int trainsTargetCreate(TARGETHANDLE target, unsigned char\* data, int width, int height, int bpp);

**Назначение:**

Создает математическую модель мишени из входного изображения. Математическая модель мишени записывается в файл, путь к которому указан в структуре params, с которой был создан модуль.

**Параметры:**

**[in]**

TARGETHANDLE target – дескриптор модуля, который получен с использованием функции trainsTargetInit.

int width - ширина входного изображения.

int height – высота входного изображения.

int bpp - число БАЙТ на один пиксель изображения.

unsigned char\* data – указатель на массив пикселей изображения.

**Возвращаемое значение:**

В случае успеха возвращает VA\_OK в противном случае VA\_ERROR

**Примечание:**

Передаваемое в модуль изображение должно быть вырезано из исходного изображения с видеокамеры, в соответствии с зоной наблюдения.

trainsTargetProcess

TRAINS\_API int trainsTargetProcess(TARGETHANDLE target, unsigned char\* data, int width, int height, int bpp, double\* result, bool\* visible);

**Назначение:**

Выполняет поиск мишени на изображении и оценивает степень ее видимости.

**Параметры:**

**[in]**

TARGETHANDLE target – дескриптор модуля, который получен с использованием функции trainsTargetInit.

int width - ширина входного изображения.

int height – высота входного изображения.

int bpp - число БАЙТ на один пиксель изображения.

unsigned char\* data – указатель на массив пикселей изображения.

**[out]**

double\* result – оценка видимости мишени. Принимает значения от 0 до 1. 0 – мишень не видна, 1 – видна идеально, 0.5 может быть мишень, а может быть и нет.

bool\* visible – решение о том, что мишень видна или не видна.

**Возвращаемое значение:**

В случае успеха возвращает VA\_OK в противном случае VA\_ERROR

**Примечание:**

VA\_ERROR – возникает в следующих случаях:

- пустой модуль (NULL)

- модель не инициализирована. (Необходимо создать модель с использованием функции trainsTargetCreate)

- не установлена зона наблюдения в модуле.

- недопустимый прямоугольник в зоне наблюдения.

trainsTargetHasModel

TRAINS\_API int trainsTargetHasModel(TARGETHANDLE target, bool\* model);

**Назначение:**

Проверяет наличие математической модели в памяти. Модель инициализируется при создании модуля, в соответствии со структурой TVAInitParams

**Параметры:**

**[in]**

TARGETHANDLE target – дескриптор модуля, который получен с использованием функции trainsTargetInit.

**[out]**

bool\* model – наличие модели. True – модель загружена. False – модель не загружена.

**Возвращаемое значение:**

В случае успеха возвращает VA\_OK в противном случае VA\_ERROR

**Примечание:**

-

trainsTargetClearModel

TRAINS\_API int trainsTargetClearModel(TARGETHANDLE target);

**Назначение:**

Удаляет модель из памяти.

**Параметры:**

**[in]**

TARGETHANDLE target – дескриптор модуля, который получен с использованием функции trainsTargetInit.

**Возвращаемое значение:**

В случае успеха возвращает VA\_OK в противном случае VA\_ERROR

**Примечание:**

-

trainTargetClose

TRAINS\_API int trainsTargetClose(TARGETHANDLE target)

**Назначение:**

Освобождает память, распределенную функцией trainsTargetInit и уничтожает модуль, созданный этой функцией.

**Параметры:**

**[in]**

TARGETHANDLE target – дескриптор модуля, который получен с использованием функции trainsTargetInit.

**Возвращаемое значение:**

В случае успеха возвращает VA\_OK в противном случае VA\_ERROR

**Примечание:**

-