

СОДЕРЖАНИЕ

Введение

1. Виды лицензий

2. Инсталляция

3. Описание

a. Поддерживаемые маркеры и их кодирование

b. Функции библиотеки

c. Пример использования совместно с библиотекой OpenCV

4. Техническая информация

5. Кредиты

ВВЕДЕНИЕ

Библиотека OMarks предназначена для распознавания оптических маркеров. В версии 1.0 поддерживаются маркеры на основе баркодов Small Aztec. Существуют различные виды лицензий, как для бесплатного использования, использования для себя и распространения со своим программным продуктом. Функции библиотеки включают в себя: распознавание маркера на 8, 24, 32 битных изображениях; коррекция положения маркеров (для дополненной реальности) распространяемые как альфа версия. Вместе с библиотекой идет пример, который откомпилирован для версии OpenCV 2.4.3 (в разделе <http://intbusoft.com/omarks/> вы можете скачать необходимые OpenCV библиотеки для запуска). Основной язык работы с библиотекой C/C++.

1. ВИДЫ ЛИЦЕНЗИЙ

OMarks FREE. Данный вид лицензии предназначен для неограниченного бесплатного использования библиотеки (в том числе и в своих программных продуктах) с ограниченными возможностями распознавания: не более 3 маркеров на одном кадре, не более 10 символов закодировано в каждом маркере.

OMarks SM PRO LIMITED. Это платная лицензия, которая предоставляет все возможности по распознаванию маркеров Small Aztec, однако в этом случае библиотеку можно использовать только в своей компании и на своих компьютерах. Нельзя распространять **OMarks** совместно со своим программным и/или аппаратным обеспечением.

OMarks SM PRO FULL. Это платная лицензия, которая предоставляет все возможности по распознаванию маркеров Small Aztec, включая распространение совместно со своим программным и/или аппаратным обеспечением.

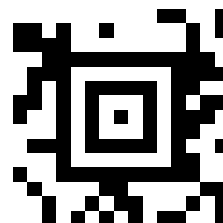
2. ИНСТАЛЛЯЦИЯ

Инсталляция не требуется. Необходимо прописать пути до Dll, lib и h файлов.

3. ОПИСАНИЕ

а. ПОДДЕРЖИВАЕМЫЕ МАРКЕРЫ И ИХ КОДИРОВАНИЕ

В версии 1.0. поддерживаются маркеры только типа Small Aztec. Данный вид бар-кодов является достаточно молодым (стандарт 2008 года), но одним из самых эффективных. Удобная центральная мишень позволяет детектировать код даже при соприкосновении с краями других графических образов. Пример Small Aztec кода:



В папке images\small aztec\ приведены 20 примеров маркеров на основе Small Aztec. Для кодирования других маркеров можно воспользоваться онлайн-сервисами или утилитой zint. Скачать утилиту можно по адресу: <http://sourceforge.net/projects/zint/>. Для кодирования в своей программе можно воспользоваться утилитой в командной строке:

```
zint -b 92 -o t.png -d "Sample Text"
```

б. ФУНКЦИИ БИБЛИОТЕКИ

Распознавание маркеров (OMark.h)

FindMarkers

Функция распознает оптические маркеры на входном изображении.

```
int FindMarkers(
```

```
char* Image_Data,  
int Image_Width,  
int Image_Height,  
int* All_Markers,  
char** Data_Result,  
int** Size_Data_Result,  
int** Points4_Mark,  
int Type_Marker,  
int Param  
);
```

Параметры:

Image_Data – входное изображение в виде массива графических данных обычного (не перевернутого, как в BMP) характера, может быть 8, 24, 32 битное, шаг строки считается как и в BMP – кратным 4 байтам;

Image_Width – ширина изображения;

Image_Height – высота изображения;

All_Markers – указатель на максимальное количество принимаемых на выходе маркеров, функция изменяет это значение на количество обнаруженных маркеров;

Data_Result – указатель на массив указателей символьного типа, в которые для каждого маркера будет возвращаться декодированный текст, указатели должны указывать на ранее выделенные области памяти;

Size_Data_Result – указатель на массив указателей на размеры буферов **Data_Result**, после выполнения функции эти значения изменяются на размер декодированного текста;

Points4_Mark – указатель на массив указателей на массивы типа int из 8 элементов, которые после выполнения функции будут содержать координаты маркера на изображении в порядке: X0, Y0, X1, Y1, X2, Y2, X3, Y3;

Type_Marker – тип маркера, который требуется найти на изображении, пока поддерживается только Small Aztec, поэтому обязательно должно быть – OMARKS_TYPE_SMALL_AZTEC;

Param – флаги:

OMARKS_IMAGE8 – 8 битное изображение (Серое);

OMARKS_IMAGE24 – 24 битное изображение (BGR);

OMARKS_IMAGE32 – 32 битное изображение (BGR);

OMARKS_QUALITY_1 – качество распознавание 1;

OMARKS_QUALITY_2 – качество распознавание 2;

OMARKS_QUALITY_3 – качество распознавание 3;

От 1 до 3 увеличивается качество, но снижается производительность.

Функция возвращает 0, если маркеров не обнаружено. 1 – если обнаружен один или более маркеров, 2 – неизвестный формат изображения, 3 – неизвестный типа маркера, 4 – неизвестная ошибка, 5 – ошибка распознавания.

Динамические свойства маркеров (OMark.h)

Данные функции пока не полностью протестированы и предназначены для дополненной реальности, в частности сглаживание дрожания маркера и детектирования его движения.

DMarksInit

Функция инициализирует динамический объект.

```
void* DMarksInit(  
);
```

Функция возвращает указатель на инициализированный участок памяти, для работы с динамическими характеристиками.

DMarksRelease

Функция освобождает использованную память, выделенную ранее функциями DMarksInit и DMarksProcessed.

```
void DMarksRelease(  
    void** Marks  
);
```

Параметры:

Marks – указатель на инициализированный ранее функцией DMarksInit указатель памяти.

DMarksProcessed

Функция, вызываемая после вызова FindMarkers для того, чтобы сравнить с предыдущими характеристиками маркеров и исключить дрожание и пропадание маркеров.

```
void DMarksProcessed(  

```

```
char* Image_Data,  
int Image_Width,  
int Image_Height,  
void** Marks,  
int* All_Markers,  
char** Data_Result,  
int** Size_Data_Result,  
int** Points4_Mark,  
int flags  
);
```

Параметры:

Image_Data – указатель на изображение, такое как в FindMarkers;

Image_Width – ширина изображения;

Image_Height – высота изображения;

Marks – указатель на инициализированный функцией DMarksInit участок памяти;

All_Markers – указатель на количество найденных FindMarkers маркеров;

Data_Result – возвращаемый FindMarkers массив указателей;

Size_Data_Result – возвращаемые FindMarkers размеры декодированных маркеров;

Points4_Mark – возвращаемые FindMarkers точки маркеров;

flags – флаги:

OMARKS_IMAGE8 – 8 битное изображение (Серое);

OMARKS_IMAGE24 – 24 битное изображение (BGR);

OMARKS_IMAGE32 – 32 битное изображение (BGR);

DMARK_TREMBLE – убрать дрожание маркера между кадрами;

DMARK_MOTION – восстанавливать потерянные маркеры при отсутствии движения.

Функция изменяет динамический объект **Marks** и корректирует характеристики маркеров, возвращенные функцией FindMarkers.

с. ПРИМЕР ИСПОЛЬЗОВАНИЯ СОВМЕСТНО С БИБЛИОТЕКОЙ OPENCV

В каталоге `samples\CameraOpenCV\` приведен пример взаимодействия OpenCV с библиотекой OMarks. Интерфейсом между библиотекой OMarks и примером является файл OMarksOpenCV.h. В данных файлах показано как работать с библиотекой OMarks.

4. ТЕХНИЧЕСКАЯ ИНФОРМАЦИЯ

Библиотека реализована под платформу Windows и была протестирована на Windows 7 64 бит и Windows XP. Обязательным требованием к работе программы является установка Microsoft Visual C++ 2010 SP1 Redistributable Package (x86):

<http://www.microsoft.com/en-us/download/details.aspx?id=8328>

5. КРЕДИТЫ

OMarks – version 1.0

2013

IntBuSoft Ltd

<http://intbusoft.com>