Общие требования ко всем приложениям

1. Записано видео об использовании приложения (с двух ракурсов или совмещенное). Видео должно быть загружено на https://www.youtube.com/. Продолжительность >= 2 минут.   Примеры:

a.   <https://www.youtube.com/watch?v=DYzOSCX6gp0>

b.   <https://www.youtube.com/watch?v=r42z259-HHE>

2. Пользовательская документация доспуна в pdf или html формате и включает в себя:

a.   Как получить и установить приложение

 b.  Как сконфигурировать и запустить приложение

c.   Описание функциональности приложения

d.   Примеры использования

e.   Известные проблемы

3. Подготовлена презентация (ppt) о разработанном демо. Презентация должна содержать следующие части:   
a.    О проекте (описане, функционал и т.д.)

b.           О разработке (ссылки на проектные ресурсы, используемые инструменты, трюки и ловушки, и т.д.).

4. Все материалы проекта (комментарии в коде, документация, демо и пр.) на Английском языке.

5.            Приложение не должно содержать критических ошибок.

**Emotion tracker**

The main aim of this project is to develop a library and demo applications to track and write users emotions (sadness, smile, laugh, etc.) during watching movie. Test recognition of emotions with different conditions. Implement logic to “merge” files with emotions and provide average rating. Implement logic to gather eyes track and to build heat map for user’s attention. All of that should be done by using Intel RSSDK tools.

**Project tree**

**Project tree**

.

├── [**emotracker**](https://bitbucket.org/valber/emotracker/src/308cf96a5df68f8a46173bd04c1dfc35d6baf8b6/emotracker/?at=master) # emotions writer prototype (used only for research │ # purposes)

├── [**library**](https://bitbucket.org/valber/emotracker/src/308cf96a5df68f8a46173bd04c1dfc35d6baf8b6/library/?at=master) # EmoTracker libraries

│ ├── [**CSharpLibrary**](https://bitbucket.org/valber/emotracker/src/308cf96a5df68f8a46173bd04c1dfc35d6baf8b6/library/CSharpLibrary/?at=master) # C# wrapper for native library

│ │ ├──[Build.docx](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/library/CSharpLibrary/Build.docx?at=master) # Build instructions

│ │ └──[CSharpLibrary.sln](https://bitbucket.org/valber/emotracker/src/308cf96a5df68f8a46173bd04c1dfc35d6baf8b6/library/CSharpLibrary/CSharpLibrary.sln?at=master) # VS 2015 solution

│ ├── [**emotracker**](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/library/emotracker/?at=master) # Native C++ Windows library

│ │ ├──[**docs**](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/library/emotracker/docs/?at=master) # Doxigen documentation

│ │ ├──[Build.docx](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/library/emotracker/Build.docx?at=master) # Build instructions

│ │ ├──[emotracker.sln](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/library/emotracker/emotracker.sln?at=master) # VS 2015 solution

│ │ └── ... # etc.

│ └── ... # etc.

├── [**rssdk2video**](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/rssdk2video/?at=master) # Utility for converting rssdk format to video formats

│ │ # (it should be in utils dir)

│ ├── [Build.docx](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/rssdk2video/Build.docx?at=master) # Miscellaneous information

│ ├── [rssdk2video.sln](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/rssdk2video/rssdk2video.sln?at=master) # VS 2015 solution

│ └── ... # etc.

├── [**samples**](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/samples/?at=master) # Examples for using

│ ├── [**EmoMerge**](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/samples/EmoMerge/?at=master) # C# application to merge emotions TTML files

│ │ └──[EmoMerge.sln](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/samples/EmoMerge/EmoMerge.sln?at=master) # VS 2015 solution

│ ├── [**EmoTracker**](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/samples/EmoTracker/?at=master) # C# application used EmoTracker library for emotions

│ │ │ # recording

│ │ ├──[Build.docx](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/samples/EmoTracker/Build.docx?at=master) # Build instructions

│ │ ├──[EmoTracker.sln](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/samples/EmoTracker/EmoTracker.sln?at=master) # VS 2015 solution

│ │ └── ... # etc.

│ └── ... # etc.

├── [**utils**](https://bitbucket.org/valber/emotracker/src/84e060bcad8f5a7200ab09518421b7f61bb4c614/utils/?at=master) # Utilities to use emotions tracks

│ ├── [**GazePainter**](https://bitbucket.org/valber/emotracker/src/b99019e0cb5b052de0c93cee63d8bfcb2136b020/utils/GazePainter/?at=master) # Map recorded gaze onto the video

│ └── ... # etc.

├── [Doxyfile](https://bitbucket.org/valber/emotracker/src/b99019e0cb5b052de0c93cee63d8bfcb2136b020/Doxyfile?at=master) # Doxygen configuration file to build documentation

├── [survey.doc](https://bitbucket.org/valber/emotracker/src/b99019e0cb5b052de0c93cee63d8bfcb2136b020/survey.doc?at=master) # About this project

├── [survey.pdf](https://bitbucket.org/valber/emotracker/src/b99019e0cb5b052de0c93cee63d8bfcb2136b020/survey.pdf?at=master) # About this project

├── [test.ttml](https://bitbucket.org/valber/emotracker/src/b99019e0cb5b052de0c93cee63d8bfcb2136b020/test.ttml?at=master) # TTML emotions record sample

└── ...

**Installation**

**Prerequisites**

emotracker:

To build it you need:

**Microsoft Visual Studio 2015**

**Microsoft Visual C++ 2015**

**Intel® RealSense™ SDK 2016 К2**

library/emotracker:

To build it you need:

**Microsoft Visual Studio 2015**

**Microsoft Visual C++ 2015**

**Intel® RealSense™ SDK 2016 К2**

library/CSharpLibrary:

To build it you need:

**Microsoft Visual Studio 2015**

**Microsoft Visual C# 2015**

**Intel® RealSense™ SDK 2016 К2**

**emotracker library**

rssdk2video:

To build it you need:

**Microsoft Visual Studio 2015**

**Microsoft Visual C++ 2015**

**Intel® RealSense™ SDK 2016 К2**

**OpenCV2**

samples/EmoMerge:

To build it you need:

**Microsoft Visual Studio 2015**

**Microsoft Visual C# 2015**

samples/EmoTracker:

To build it you need:

**Microsoft Visual Studio 2015**

**Microsoft Visual C# 2015**

**Intel® RealSense™ SDK 2016 К2**

**emotracker library**

**CSharpLibrary library**

utils/GazePainter:

To build it you need:

**Microsoft Visual Studio 2015**

**Microsoft Visual C+ 2015**

utils/GazeHeatMap:

To build it you need:

**Microsoft Visual Studio 2015**

**Microsoft Visual C# 2015**

**Installing**

library/emotracker:

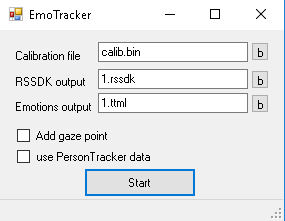
1. Open emotracker.sln with Microsoft Visual Studio 2015
2. Check path to RSSDK include directory
   1. Project -> Properties
   2. C/C++ -> General -> Additional include path it should contain RSSDK include path: $(RSSDK\_DIR)/include
3. Check path to RSSDK libraries directory
   1. Project -> Properties
   2. Linker -> General -> Additional library path it should contain RSSDK library path $(RSSDK\_DIR)/lib/$(PlatformName)
4. Build and save the library emotracker.dll, under the local bin directory

library/CSharpLibrary:

1. Open CSharpLibrary.sln with Microsoft Visual Studio 2015
2. Build and save the library CSharpLibrary.dll, under the local bin directory

**Usage**

**Case 1. Track and write users emotions during video playback**

1. Go to the projects bin directory
2. Run EmoTracker.exe and set calibration file (can be obtained by calibrate process used FF\_EyeTracking), output for stream from camera and emotions subtitles file

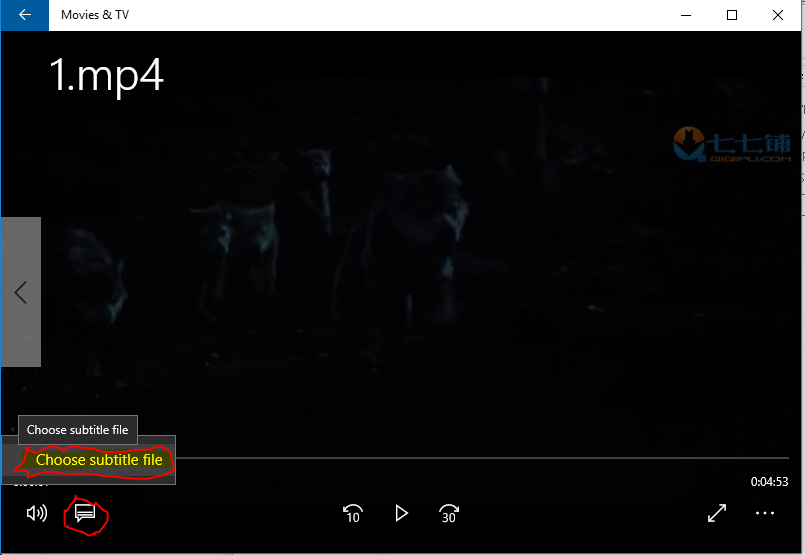
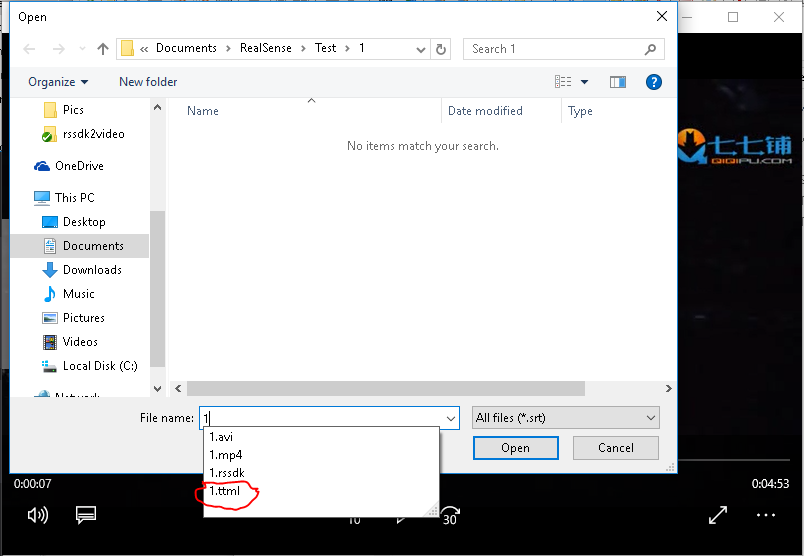
t.ex.: 1.rssdk and 1.ttml

1. Click Start button
2. To finish recording press Stop button
3. As a result, it is a two files 1.rssdk with camera output record and 1.ttml with emotions track and gaze directions

How to use results

* ttml файл содержит текст с разбивкой по времени (см.[https://www.w3.org/TR/**ttml**2/](https://www.w3.org/TR/ttml2/)) позволяющий, в том числе, быть представленным в виде субтитров к видеоконтенту, одним из поддерживающих такой формат плееров, является приложение Movies & TV

в Windows 10

1. Следует выбрать проигрываемый файл (например, видео файл, который был запущен во время снятия трека эмоций) для воспроизведения с помощью Movies & TV
2. Подключить проигрывание субтитров
3. При выборе файла, файлы с расширением ttml будут не видны, следует явно указать имя файла с субтитрами
4. можно, также, в качестве проигрываемого файла использовать запись снятую с камеры, для чего, в начале, выделить из записи в формате rssdk поток, в проигрываемом стандартными плеерами формате. Например, имея файл записи 1.rssdk и собранный проект rssdk2video, выделить видеопоток в 30 fps с разрешением 480x270, посредством запуска в командной строке:

**Case 2**

**Case 3**

**Contributing**

1. Fork it!
2. Create your feature branch: git checkout -b my-new-feature
3. Commit your changes: git commit -am 'Add some feature'
4. Push to the branch: git push origin my-new-feature
5. Submit a pull request :D

**History**

This is first release

**Credits**

Thanks to Intel for the funny days I have spent with its code.

**License**

I think it should be Copyleft.

But seems Intel want Apache License 2.0