



# VisualScores v1.0

**Minimalist video maker --  
simplify your music score video making process!**

## User Guide

Our previous program **SmartCropper** (URL: <https://github.com/Chen-and-Sim/SmartCropper>) has significantly simplified the process of making music score videos. However, in order to create the final video from the cropped images, you need to drag the images one by one to a video production software (like **Adobe Premiere**), adjust the time of each images precisely and then export the video. For long music scores, this process also takes huge amounts of time.

This video maker **VisualScores** is focused on creating music score videos. It has abandoned redundant features (like image filters) of prevalent video production softwares. What you only need to do is to press Enter to partition the audio file while it is playing. After that a simple music score video is created. Of course, besides music score videos, all videos which are composed of static images + music can be simply handled using **VisualScores**!

**VisualScores** is a green software, which requires no installation.

Just go to this URL: <https://github.com/Chen-and-Sim/VisualScores>, download, open **VisualScores.exe** and run it. Default interface is in English.

```
D:\VisualScores.exe
-x settings Show current settings.
-i <Path> [Pos] load <Path> [Pos]
Load an image to the image track.
-I <Path> [Pos] loadall <Path> [Pos]
Load all images in the folder located at <Path> to the image track.
-o <Path> [Begin] [End] loadother <Path> [Begin] [End]
Load an image to the background track or load an audio file to the
audio track.
-d <Tag> delete <Tag>
Delete the file tagged <Tag>.
-m <Tag> <...> modify <Tag> <...>
Modify the file tagged <Tag>.
-t <Tag> <Time> duration <Tag> <Time>
Set the duration of an image file not in the range of any audio file.
-p [Tag] partition [Tag]
Partition the audio file tagged [Tag] and determine the duration of
images in the range of the audio file.
-D <Tag> discard <Tag>
Discard the partition done to the audio file tagged <Tag>.
-e [Path] export [Path]
Export the video file to [Path].
For detailed descriptions please refer to the user manual.
VisualScores>
```

Type the command -1 (Enter) to switch to the Chinese interface.



**Copyright statement: This software is open-source under GNU GPLv3 license.**

## ◎ How to use

### ▲ Step 1: Load the files

**VisualScores** provides one image track, one audio track and one background image track. Let us see how to load files into these three tracks. Some parts of the program output will be explained later.

◆ **Load an image to the image track** Use the `load` command (short command: `-i`).

```
VisualScores> load pdf.png (or: -i pdf.png)
```

The program supports both relative and absolute path. We recommend placing the program in the same directory as the image, so that the previous path can be omitted.

◆ **Load multiple images to the image track at once** More often than not we would like to load all images in the same folder to the image track. Then we can use the `loadall` command (short command: `-I`).

```
VisualScores> loadall crops (or: -I crops)
```

(where `crops` is the name of the folder, and a relative/absolute path is available)

◆ **Load an audio file to the audio track**

Use the `loadother` command (short command: `-o`).

```
VisualScores> loadother music.mp3 (or: -o music.mp3)
```

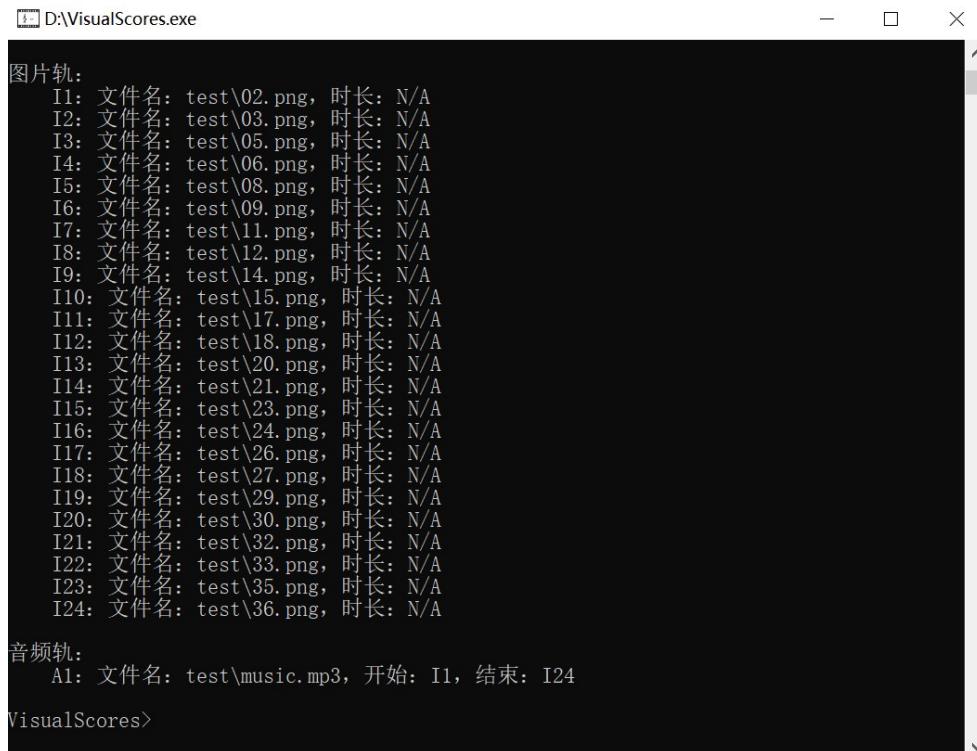
◆ **Load an image to the background image track** Sometimes we want to add a background image, or add some information of the work at a corner, then we can add images to the background image track. We still use the `loadother` command (short command: `-o`).

```
VisualScores> loadother bg.png (or: -o bg.png)
```

When you use the `loadother` command, the program will automatically determine which track the file should be loaded into judging from the extension of the file. Besides, we recommend you to let the aspect ratio of the background image be consistent with those in the image track.

## ▲ Step 2: Partition the audio file

After loading all files, we may see the program output as follows:



We can discover that each file is associated with a **tag**, which is I1 ~ I24 and A1 in the example. Moreover, the audio file **tagged** A1 shows its **range**: I1 ~ I24. This means that while the video is playing the audio file A1, images I1 ~ I24 are displayed at the same time. Besides, the display duration of images I1 ~ I24 equals to the duration of the audio file A1.

◆ **Partition the audio file** Use the `partition` command (short command: -p).

`VisualScores> partition A1 (or: -p A1)`

After giving this command, the program will pop up a window to display the images in the **range** of the audio file A1. After 3 seconds, the program will play the audio file A1 (without popping up new windows), and display the first image in the **range** of A1 in the pop-up window.

⚠ **Note 1:** The time playing the audio file and the time displaying the first image file is not exactly the same, with the former one late for 0.5 to 1 second (sometimes longer). **You should pay full attention and press Enter when the first note is played so as to start timing.** (You do not need to worry about the error brought by reaction time. We have set the begin time 0.2 seconds earlier than actual time.)

When you press **Enter** for the first time the program will inform you that you have begun partition. After that, you should press **Enter** whenever you need to switch to the next image. Then the program will record the time and display the next image in the pop-up window. When you have switched from the last but one to the last image, the program will tell you that you have succeeded. Eventually the program will take the recorded time as cut-off points to partition the audio file.

In the process of partitioning, you can press **Esc** at any time to stop the process.

## ▲ Step 3: Export the video file

When all image files have their **duration** set (i.e. there is no **N/A** in the output message), we can export the video file.

◆ **Export the video file** Use the `export` command (short command: `-e`).

**VisualScores>** `export` (or: `-e`)

The video will be exported to `_video.mp4` at the same directory.

## ◎ Advanced usages

Below we will surround arguments with `<>` to represent necessary arguments and with `[ ]` to represent arbitrary arguments.

## ▲ Load the files

By adding some arguments at the end of a command we have previously mentioned, we can specify the **position** and **range** of the files.

⚠ **Note 2:** The maximum number of files in each track is 300.

## ◆ Load an image to the image track

`VisualScores> load <Path> [Pos] (or: -i <Path> [Pos])`

Argument `<Path>` is the relative or absolute path of the file, including extension.

Argument `[Pos]` is the position of the file. If it is not given, then the file will be loaded to the end of the image track; if it is `0`, then the image will be loaded to the front of the image track; if it is a number `n`, then the image will be loaded behind the `n`-th image (i.e. the image **tagged In**) in the image track.

**⚠ Note 3:** Supported image formats: `bmp`, `jpeg`, `png`, `tiff`, `webp`, `ico`.

## ◆ Load all images in the same folder to the image track

`VisualScores> loadall <Path> [Pos] (or: -I <Path> [Pos])`

Arguments `<Path>` and `[Pos]` are as above.

**⚠ Note 4:** If a file is loaded into the **range** of an audio file or a background image file, then the **range** is changed, and the partition done to the audio file will be discarded.

## ◆ Load an audio file to the audio track or load an image file to the background image track

`VisualScores> loadother <Path> [Begin] [End]  
(or: -o <Path> [Begin] [End])`

Argument `<Path>` is as above.

Arguments `[Begin]` and `[End]` determine the **range** of the file loaded. For example, if they are `2` and `4`, then the **range** of the file is `I2 ~ I4`.

If arguments `[Begin]` and `[End]` are not specified, then the **range** is the whole image track.

**⚠ Note 5:** Supported audio formats: `wav`, `mp3`, `flac`, `aac`.

**⚠ Note 6:** You can not overlap audio files and you can overlap background image files.

## ▲ Modify the files

Here the word ‘modify’ does not mean modifying the file itself, but modifying the parameters like **position**, **range** and **duration**.

### ◆ Delete a file

`VisualScores> delete <Tag>` (or: `-d <Tag>`)

Argument `<Tag>` is the **tag** of a file.

**⚠ Note 7:** If the file deleted is in the **range** of an audio file or a background image file, then the **range** is changed, and the partition done to the audio file will be discarded.

### ◆ Change the position of an image file in the image track

`VisualScores> modify <Tag> <Pos>` (or: `-m <Tag> <Pos>`)

Argument `<Tag>` is the **tag** of a file.

Argument `<Pos>` is the position the file is moving to. If it is `0`, then the image will be moved to the front of the image track; if it is a number `n`, then the image will be moved to behind the `n`-th image (i.e. the image **tagged In**) in the original image track.

### ◆ Change the range of an audio file in the audio track or an image file in the background image track

`VisualScores> modify <Tag> <Begin> <End>` (or: `-m <Tag> <Begin> <End>`)

Argument `<Tag>` is the **tag** of a file.

Arguments `<Begin>` and `<End>` determine the new **range** of the file.

When you use `modify` command, the program will automatically determine which specific command you would like to give judging from the **tag**.

**⚠ Note 8:** If you modify the range of an audio file which has been partitioned, then the partition will be discarded.

## ◆ Set repetition

**VisualScores>** repeat <Begin> <End> <Times> (or: -r <Begin> <End> <Times>)

Arguments <Begin> and <End> determine the range of repetition.

Argument <Times> is the number of times of repetition. **Times** = 1 means repeating one time, i.e. the image(s) will show 2 times in the video.

**⚠ Note 9:** The maximum number of repetition times is 50. The number of images in the image track should not exceed file limit 300 after counting repeated images multiple times.

## ◆ Change the duration of an image file in the image track

**VisualScores>** duration <Tag> <Time> (or: -t <Tag> <Time>)

Argument <Tag> is the **tag** of a file.

Argument <Time> is the **duration** of a file in seconds.

**⚠ Note 10:** Any file in the **range** of an audio file can not change its **duration** via this command.

**⚠ Note 11:** The range of **duration** is 0.1 ~ 10000 seconds.

# ▲ Partition the audio file

## ◆ Partition the audio file

**VisualScores>** partition [Tag] (or: -p [Tag])

Argument [Tag] is the **tag** of a file. If it is not given, then the first audio file that is not partitioned is choosed.

## ◆ Discard the partition done to an audio file

**VisualScores>** discard <Tag> (or: -D <Tag>)

Argument <Tag> is the **tag** of a file.

## ▲ Export the video file

### ◆ Export the video file

`VisualScores> export [Path] (or: -e [Path])`

The argument `[Path]` is the full filename of the video file, including a (relative or absolute) path. If `[Path]` is not given, then the video file will be exported to `_video.mp4` at the same directory. You can use the string ‘`desktop`’ in either case to refer to your desktop directory.

⚠ **Note 12:** Supported video format: mp4.

⚠ **Note 13:** The total length of the video can not exceed 10000 seconds.

## ▲ Basic commands

### ◆ Show the information of the program

`VisualScores> about (or: -a)`

### ◆ Show help

`VisualScores> help (or: -h)`

### ◆ Switch the language of the program

`VisualScores> language (or: -l)`

### ◆ Quit the program

`VisualScores> quit (or: -q)`

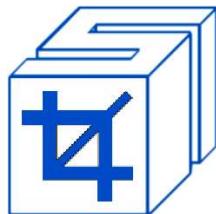
### ◆ Show current settings

`VisualScores> settings (or: -x)`

## ◎ Remarks and acknowledgements

- ◆ **VisualScores** is suitable beyond music scores! All videos which are composed of static images + music, even animated images can be simply handled using **VisualScores**!
- ◆ **VisualScores** may provide some simple effects and add **mov**, **wmv** and **avi** video formats in the future.
- ◆ **VisualScores** is written by Ji-woon Sim.  
NetEase Music: <https://music.163.com/#/user/home?id=503598921>
- ◆ **VisualScores** can not be created without the support of WenGe CHEN (Zhihu: @Raymond)!

We also recommend using our free and open source --



**SmartCropper** 智剪

*Crop sheet music images fast!*

乐谱裁剪神器

<https://github.com/Chen-and-Sim/SmartCropper>



**ChordNova** 智弦

*Beyond boundaries!*

和声生成神器

<https://github.com/Chen-and-Sim/ChordNova>



Copyright statement: This software is open-source under GNU GPLv3 license.