

VideoKifu v1.2.0

for Windows[®] operating systems

© 2012–2018 Andrea Carta & Mario Corsolini

URL: www.oipaz.net/VideoKifu.html

eMail: andrea.cartamclink.it

mario@corsolini.net

Contents

	Page
Front matter	1
Contents	2
1. Introduction	3
2. Licence	3
3. System requirements	4
4. How to install	4
5. How to use	5
5.1 Video camera setup	5
5.2 Use of VideoKifu	7
5.3 Errors during the analysis	8
5.3.1 Errors in the grid position	8
5.3.2 Wrong moves	8
5.4 Resuming of a previous analysis	9
5.5 Games broadcast	10
6. How to uninstall	11
7. Feedback, suggestions, bug-reports	11
8. Release history	13
9. Acknowledgements	14

1. Introduction

VideoKifu reconstructs the whole move sequence of a Go game from a possibly unattended video feed, either live or deferred. It produces an SGF file and a game record (the so called kifu); the move sequence may be published in real-time on the Internet, via FTP connection.

Demo version has restrictions both in maximum analysis time and in maximum number of moves, as shown in the following table:

goban size	max time	max moves
9×9	unlimited	999
13×13	5 minutes	42
19×19	10 minutes	90

Table 1: limitations of the demo version.

2. Licence

VideoKifu is licenced under the Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International Licence (CC BY-NC-ND 4.0). To view a copy of the licence, visit:

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

VideoKifu is **donationware**: if you like it and you want to contribute to its development, you may send a **donation** to the authors using the PayPal account VideoKifu@gmail.com.

3. System requirements

Minimum and recommended requirements to use VideoKifu efficiently are listed in table 2 at page 4.

	Minimum	Recommended
Operating system	Windows (version 7 or later)	
CPU	multi-core 32-bit	multi-core 64-bit
Memory	2 Gabyte RAM	8 Gabyte RAM
Display resolution	1024 × 768 pixels	1280 × 800 pixels
Hard drive	100 Miabyte free space (except videos)	
Peripherals	webcam 480p	smartphone or tablet 720p
Other requirements	tripod	tripod, dedicated Wi-Fi

Table 2: system requirements.

4. How to install

VideoKifu does not need any installation: just unzip and run it!

Should VideoKifu be used on a K or KN Windows edition, it could warn that the file MFPlat.dll is missing; if such were the case you may need to retrieve and install the “Media Feature Pack” for your specific Windows version from the Microsoft official website.

Furthermore, on old Windows versions you may need to retrieve and install the [Microsoft® .NET Framework 4](https://www.microsoft.com/en-US/download/details.aspx?id=24872) from the Microsoft official website: <https://www.microsoft.com/en-US/download/details.aspx?id=24872>

If VideoKifu is used under Windows 8.1 (or previous), with a high-definition display and custom DPI setting higher than 100%, the “XP style DPI scaling” tick box must be left in the default unchecked state.

5. How to use

5.1 Video camera setup

VideoKifu can analyse both pre-recorded or live videos, by means of a webcam, a smartphone or a tablet (or a video camera that Windows could identify as a webcam). Should a webcam be used, it is mandatory it should be able to record a video without excessive geometric distortions and keep in focus the whole goban, something that unfortunately not many webcams are capable of.¹ Alternatively any Android mobile device could be used either by means of a USB port (DroidCamX² needs to be installed and set up, both on mobile device and PC, according to the instructions available online) or by means of a Wi-Fi connection (IP Webcam³ needs to be installed, only on the mobile device). The latter is preferable when a fast and reliable Wi-Fi connection is available, as IP Webcam's free version is able to provide video at the optimal resolution of 1440×1080 pixels⁴ and also to fix the focus on the goban, even when players' hands are moving over it.

To use a smartphone or tablet connected to a Wi-Fi network as an "IP camera" you need to give VideoKifu its exact complete address. To obtain it open the IP Webcam app on your Android device (similar apps may be found for other platforms) and make the necessary adjustments for resolution, quality and focus. After that, scroll to the bottom and tap on "Start server". An IP address will appear in the bottom of the screen with a port number (something like `http://192.168.1.3:8080`). Copy the entire address into the web browser of your PC: it will open up the IP Webcam dashboard. Choose the option "Video renderer: Browser". You should see the video feed in the web browser: right-click on it and select "Copy image address". Paste the address into the box that appears when you select "IP camera" as video source

¹ It may be possible to correct the focus by means of the "Unsharp mask" filter in the "Options" tab. Click on the parameters to modify them.

² <https://www.dev47apps.com/>

³ <https://play.google.com/store/apps/details?id=com.pas.webcam>

⁴ In order to improve the speed, audio could be disabled and video quality could be slightly decreased from the default value (100) in the "Video Settings" menu.

in VideoKifu (it may be something like `http://192.168.1.3:8080/video`). Be careful to confirm the address (with <Enter> key) only when you are sure the camera is connected and working, otherwise the program could hang or crash. After the end of the analysis close IP Webcam selecting “Stop” from the “Actions...” menu.

The video camera must be placed on the side of the goban (whose sides will be almost parallel to the frame's), as high as possible. The grid of the goban must be thoroughly in focus, entirely visible (including the edges) and it must take up most of the frame, or else the program could fail to locate it.

It's important for the focus not to change during the game; before the video starts don't forget to check, by means of the video camera's software, the autofocus to be inactive.


Video camera's resolution should be high, possibly 720p or 1080p, and never less than 480p. Bear in mind that with higher resolutions too much time could be spent on the analysis, whereas lower resolutions could prevent the stones to be detected. If the lines on the goban are very thick a low resolution could be necessary.

Try not to move the goban nor the camera during the game: should that happen the program is able to correct the shifts at once, but only if they are small (approximately the radius of a stone, at most) between two consecutive frames. Larger shifts may also be automatically corrected (if at least half the grid stays on sight), but a delay of some seconds is to be expected. If the automatic correction does not work it will be necessary to manually correct the location of the grid, as explained later.



The goban must be evenly illuminated; take care not to change too much the lighting during the game, especially trying not to have areas of the goban in the shade or, vice versa, areas of light so intense to appear almost white because of the reflections of light sources. If possible, check white balance of your video camera and colour accuracy before shooting the video.

Finally, if your camera has automatic rotation, disable it. A video shot in portrait orientation must be analysed in portrait orientation, a video shot in landscape orientation (which is usually better) must be analysed in landscape orientation.

5.2 Use of VideoKifu

When VideoKifu analyses a game — either live or deferred (if such were the case a video file should be chosen by means of the specific button on the main window) — at first it tries to locate the goban's grid; when found, the grid will be plotted on the screen, above the real one, in red; then, should not change its position, its colour will vary from red to green. Should the position not change for long enough the grid's colour will remain light green, and the button  will be enabled: pushing it will let the analysis start. VideoKifu will wait for the first stone to be played, and every time one will be put on the goban the move will be detected and the list on the right will be updated, as well as the position on the centre.

The analysis should be started a short time before the beginning of the game (after nigiri, if it is performed). VideoKifu is able to analyse already started games, yet with two restrictions: handicap stones (if there are any) might be entered manually and there should not be more than nine white stone and nine black stones on the goban (anyway, the recognised move sequence will be almost random). If there are more stones the program might not be able to initiate the analysis, certainly not in a short space of time.


Should not occur any problem, all the stones will be detected and, once the video ended (with the button  if live, otherwise at the end of the file), the user, after having possibly amended the moves' list, will be able to create the SGF file and the kifu (if the specific options has been selected) by means of the button  (also available during the game, pausing the analysis). The program will also try to compute the score, given the game has not been resigned (likely the program will acknowledge the resignation, too).

Other options and information about the game (such as players' names, komi and so on) are not mandatory and may be input, as every other, even after the start of the game. Only the rules of the game are mandatory (by default Japanese rules are suggested).


During the analysis it is possible, by means of the <PrtSc> key (print screen), to save the current frame in PNG format, as shown on the left.

That file, as well as any other file generated by the program, is saved in the “Destination directory” chosen by the user.

5.3 Errors during the analysis

During the analysis errors should not occur. Yet the goban could be moved, intentionally or not, completely shifting the grid position; stones could also be erroneously detected, although such an event should be extremely uncommon. The user may correct these errors during the analysis, and it’s advisable to do so as quickly as possible in order to avoid more errors further on. After pushing the button  the errors can be corrected these ways:

5.3.1 Errors in the grid position





In order to correct the grid position the user will push the button : this way the program will try and recompute its position, and the grid will be plotted over the goban; if the grid would not look correct it will be possible to straighten it either clicking on the goban’s corners (just moving the mouse on the corners will zoom them) or by means of the keyboard’s arrow keys that will shift it accordingly. Once straightened, the grid should be validated by pushing the big button on the centre of the frame.

Some gobans have not evenly spaced grid lines; in such cases it is possible that internal intersections are inaccurate even though the borders of the grid are correctly recognised. You may try and avoid the problem by changing the method of evaluation of the internal intersections,⁵ pressing the <Esc> key during the above-mentioned manual calibration and confirming the option that gives the best results.

5.3.2 Wrong moves

In order to correct any error in the moves the user will find the most suitable position on the moves’ list on the right and will do the necessary adjustments


⁵ Those methods are: computed spacing (based on the initial recognition of the grid) and projective spacing (computed by geometrical means only).

by looking at the frame on the left and its graphical depiction on the centre (synchronized between them and with the moves' list). It's possible to delete a move by means of the button  (the move to be deleted will be the selected one), add another one by means of the button  (the new move will be added after the selected one), and also invert the moves following the selected one by means of the button  (because if a move is not detected the following ones will be detected as couples, in wrong order). Once the corrections ended the analysis will resume by means of the button .

Let's keep in mind that, if the analysis is live, moves played during a pause might not be detected afterwards (or could be, but in wrong order): that's likely if such moves are more than two, and almost sure if they are more than four. That's why it's strongly advisable to make any correction as quickly as possible. In order to obtain this, some corrections (move, swap or remove stones) may be applied by directly right-clicking in either picture of the goban (real or calculated one), with no need of previously pausing the analysis. Of course if the analysis is deferred instead, it will resume from the same point in the video where it had been stopped.

Any "Pass" played before the end of the game will have to be recorded manually, as at present the program does not manage them.

5.4 Resuming of a previous analysis

It is possible to resume a previous analysis, already come to an end, if the user wishes to verify and correct any possible errors. For this purpose a file, with extension VK, is saved at end of each analysis: the program can later open this file (using the filter "VideoKifu analyses" in the file browser window), which contains all the information needed to resume the analysis starting from any move, that the user should click on the list on the right. The frame shown on the left, as well as the position on the centre will be synchronized accordingly. The analysis will resume from the selected move, after pushing the button .

In the case the user wishes to resume a live analysis, a video file, in MP4 format, will also be available in the "Destination directory": this file is

automatically saved by the program, and contains all the frames VideoKifu analysed during the game. The most powerful the PC used for the analysis, the most frames will be saved in the video (that should be played at 6 FPS, according to the program's parameters, although the actual speed could be different: in any case, the value is changeable in the command line by means of the parameter `/FPS:n`, `n` being between 1 and 240).

5.5 Games broadcast

The moves detected during a game may be broadcast by means of a form that can be open by pushing either the relative button in the "Options" tab or the small "broadcast" icon in the main window's lower right corner.

In order to activate the broadcast you should specify the folder where the templates are located; templates include stones' pictures and HTML files (main index template and single games template). Each one of those files is included in the folder "vktv" of VideoKifu's package. Also the complete absolute address of a pre-existing folder on some FTP server should be specified, along with username and password needed to access it, as the games will be published there. After that it will become possible to send the requested files by means of the specific button and, should everything go smoothly, the button that will activate the broadcast will be enabled. Should there be any problems they will be exposed in the warnings' area.

The broadcast can be started or stopped at any time. At the end of the game the files created by VideoKifu could be edited (and resent to the server) manually.

The aforementioned models are HTML files, that may contain some meta-commands managed by VideoKifu. The meta-commands' full list is shown in table 3 at page 12. To understand how they work it is suitable to analyse the content of the files "vktv\vktv_index.html" and "vktv\vktv_game.html" included in VideoKifu's package.

We'd have liked for VideoKifu to broadcast analysed games on some of the known Go servers, such as KGS, IGS or OGS. Yet connection to KGS not only requires Java's installation, but also use of a proprietary interface, something

we should not impose on our users; IGS never answered our requests, so we could not even understand how to connect to their servers; OGS is still in development and supplies neither documentation nor the support required for someone who would like to connect and broadcast a game. If sometime in the future any of those problems will be solved, we'll add the feature to VideoKifu right away!

6. How to uninstall

Close VideoKifu (if needed), delete its folder and any file created during its use. That will completely uninstall VideoKifu. :-)

If you had to install [.NET Framework 4](#) or the “Media Feature Pack” or a client for a smartphone or tablet, you may remove them through the usual “Add/Remove” applet of the Control Panel (be careful: they may be shared with other applications).

7. Feedback, suggestions, bug-reports

Any comments, suggestions and (most of all) bug-reports are welcome. Please use the eMail address andrea.carta@mclink.it or mario@corsolini.net

It is advisable to specify “VideoKifu” in the subject field and to report the version of both the program and the operating system. While submitting an error it is also advisable to attach the last VK file (if available) saved by VideoKifu in the chosen “destination directory”.

VideoKifu is a multilingual application: contact us if you are willing to translate it into another language!

meta-command	index	game	alone
<code><!-- VideoKifuTV Games List [names] --></code>	✓		✓
<code><!-- VideoKifuTV Auto Refresh [s] --></code>		✓	✓
<code><!-- VideoKifuTV Game Details [size] --></code>		✓	✓
<code><!-- VideoKifuTV Game ID --></code>		✓	
<code><!-- VideoKifuTV Game Title --></code>		✓	
<code><!-- VideoKifuTV Goban --></code>		✓	✓
<code><!-- VideoKifuTV Moves --></code>		✓	✓
<code><!-- VideoKifuTV SGF --></code>		✓	✓
<code><!-- VideoKifuTV Stone Size --></code>		✓	
<code><!-- VideoKifuTV Video Source --></code>		✓	
<code><!-- VideoKifuTV Warnings --></code>		✓	✓
<code><!-- VideoKifuTV EN Only --></code>	✓	✓	
<code><!-- VideoKifuTV IT Only --></code>	✓	✓	
<code><!-- VideoKifuTV Version --></code>	✓	✓	

Table 3: HTML meta-commands managed by VideoKifu.

Commands checked in the “index” column may be used in the game index template file, while those checked in the “game” column may be used in single games template file; commands checked in the “alone” column must be used “alone”, without anything else on the same row (they will be replaced by suitable contents, keeping the indentation). Warnings’ background colours may be customised by setting the variables `-errors-background` and `-warnings-background`. These are the optional parameters’ possible settings:

`names` = Titles | Sources | Titles+Sources (default);

`s` = seconds before automatic page refreshing;

`size` = rules’ details’ dimension, expressed in any HTML measure unit (to omit rules’ details the field must be left empty, keeping square parenthesis).

8. Release history

- Version **1.2.0** — September 13th 2018
 - Added recognition of nigiri and (approximatively) of game's end.
 - Added various levels of enlargement of video stream's image.
 - Added analysis with all-manual grid location and tracking.
 - Bug fixed: sometimes manual calibration could freeze the program.

- Version **1.1.0** — August 8th 2018
 - Added analysis of already started games.
 - Added asynchronous tracking of grid position in the end game.
 - Added the button “Publish current position” to the “Manage broadcast” window.
 - Added an indicator for the number of processed frames per second.
 - Improved management of warnings, whose list is now included in the SGF file.
 - Bug fixed: saved videos could be broken or reset in case of interruption of the video feed.
 - Bug fixed: using IP Webcam the camera list was not correctly updated; furthermore, neither the VK file nor the address of the last used camera were saved.
 - Bug fixed: it was not possible to check broadcast settings' modifications on the fly.
 - Minor cosmetic changes.

- Version **1.0.0** — May 28th 2018
 - Added asynchronous detection of stones.
 - Added asynchronous tracking of grid position (not in end game).
 - Added management of IP cameras' video feeds.
 - Added some kind of moves' editing by clicking on goban pictures.
 - Added online broadcast of detected moves.
 - Added “Unsharp mask” filter.
 - Added image cropping for live video frames.
 - Added an icon to notify if the grid is too close to the frames' edges.
 - Added rules' editing even during the analysis.
 - Added warnings generated during the analysis to the SGF file.
 - Improved ko detection, with rules violations' notification.
 - Improved notifications of uncertain situations.
 - Modified <F2> key's function: it activates moves' list editing (to open the “About VideoKifu” window use <Ctrl>+<F1> instead).
 - Bug fixed: handicap stones were not counted correctly after a manual moves' editing.
 - Bug fixed: “Swap stones' colour...” button often wrongly set stone's colours.

- Version **0.9.9demo** — July 14th 2017
 - First public release (demo version).

9. Acknowledgements

Thanks to the authors of OpenCV⁶ and Emgu CV⁷ for providing their useful software.

Most of the icons used in VideoKifu belong to the collections: Must Have by VisualPharm,⁸ Oxygen by Oxygen Team,⁹ Sleek XP Basic by Hopstarter

⁶ <http://opencv.org/>

⁷ <http://www.emgu.com/>

⁸ <http://www.visualpharm.com/>

⁹ <http://www.oxygen-icons.org/>

(Jojo Mendoza),¹⁰ Network Set 1 by DevCom¹¹ and Farm-Fresh by FatCow Web Hosting.¹²

The authors want to thank all the people who helped them with ideas and valuable suggestions, particularly: Dani Ferrari for his advice on the stone recognition algorithm, Hyun-Soo Park¹³ (and coworkers) for his works on a static score-counting algorithm, as well as the beta testers!

And, obviously, thanks to Laura and EmmeTi!!

Well, that's all about it, happy playing!!!

¹⁰ <http://hopstarter.deviantart.com/>

¹¹ <http://www.devcom.com/>

¹² <http://www.fatcow.com/free-icons>

¹³ Department of Computer Information Technology, Kyungdong College of Techno-Information, 224-1, Buho, Hayang, Kyungpook, Korea.