.

# CVBASE 06 Dataset: A Dataset for Development and Testing of Computer Vision Based Methods in Sport Environments

User manual and short reference

Janez Perš, janez.pers@fe.uni-lj.si

Ljubljana, October 26, 2005.

# Contents

# Abstract

*This document briefly describes the CVBASE 2006 dataset, which contains videos of sport activity and related data. The purpose of the dataset is to provide a sport environment testbed for various methods from the fields of computer vision and artificial intelligence. Originally, the dataset has been prepared in time for the Workshop on Computer Vision Based Analysis in Sport Environments (CVBASE 2006), organized in conjunction with ECCV 2006 in Graz, Austria. The dataset provides three types of data: videos, player trajectories, and expert annotations. Included are recordings from three different kinds of sport activity: handball, squash and basketball. While handball and squash data include video, trajectories and annotations, the basketball data is provided as video only. Overall, dataset includes approximately 1 hour of video.*

# License

Basically, we do not want unauthorized modifications of the dataset and we do not want dataset to be used in scientific fields for which it was not intended.

Please read the file `license.txt` which accompanies the dataset for the details. We tried to phrase the licence in the way that would give the researchers maximum freedom of work and publication and encourage wide use of the dataset.

# 1 Data sources

This chapter briefly summarizes the sources of video data, the process of encoding/transcoding the videos and the sources of annotations.

## 1.1 Handball subset

Handball video data was obtained by recording a "model" handball match (not regular league or tournament match). Players did not leave the court during the match and did not change their roles during the play. Videos, annotations and trajectories are from the first 10 minutes of the match.

### 1.1.1 Trajectories (player positions)

Trajectory data was obtained using automatic computer vision based tracking under field expert supervision (where possible). If automatic tracking did not perform well, manual tracking was employed by field expert. Positions were entered with variable frequency (1-12 frames interval, decision by expert) with linear interpolation on the frames between. The trajectories (from both sources) were smoothed by Gaussian kernel ($-3\sigma$ .. $+3\sigma = 25$ samples)

Trajectory data is provided for use in higher-level processing to provide a shortcut for the researchers who are not able to perform tracking by themselves.

Trajectory data is supplied both in court coordinates and in the coordinate system of cameras A and B.

RMS error in player position is approx. 0.3 meters directly beneath the camera and approx. 0.6 meters at the edge (or the middle) of the court.

*Do not treat trajectory data as gold standard – it is NOT provided for that purpose and is not sufficiently accurate.*

### 1.1.2 Annotations

Annotations were supplied by field expert (a coach) from observations of the video `HandballC.avi`. They are comprised of two separated sets, the first one describes individual player activity annotations (passes, shots, etc.) and the second one the team activity (offense, defense, etc.).

Annotations and trajectories are supplied just for one of the teams. Players of this team were wearing dresses of different colors with the purpose of easy identification. Videos A, B and C are provided synchronized.

### 1.1.3 `HandballA.avi` and `Handball B.avi`

- Duration: 10 minutes each
- Camera: 1/2" PAL CCD camera, fixed, bird's eye view
- Camera position: birds-eye view, stationary, wide angle lens
- Recorder: S-VHS videorecorder
- Digitizer: Pinaclesys DC30+ video capture card, HW MJPEG
- Initial format: 25 fps, M-JPEG, 384x576, 2 fields - interlaced
- Initial bitrate: 1559 KBit/s
- Odd fields discarded, transcoded to DivX 5.2.1 @ 1580 KBit/s

### 1.1.4 `HandballC.avi`

- Duration: 10 minutes
- Camera: Handheld camcorder
- Camera position: handheld, no limitations on motion, zoom, etc.
- Recorder: Handheld camcorder
- Digitizer: Pinaclesys DC30+ video capture card, HW MJPEG
- Initial format: 25 fps, M-JPEG, 768x576, 2 fields - interlaced
- Initial bitrate: 824 KBit/s
- Odd fields discarded, 1:2 reduction in horizontal direction.
- Transcoded to DivX 5.2.1 @ 1580 KBit/s

## 1.2 Squash subset

Source of the two squash videos are two different tournaments of different quality levels. `squash1.avi` was recorded on a tournament of recreative players, and `squash2.avi` was recorded at the Slovenian squash championship. Both videos were

Each video contains one whole set of the squash match. In both cases the recorded set was the first set of the match.

### 1.2.1 Trajectories (player positions)

Trajectories, provided as part of the squash subset were obtained using automatic computer vision based tracking under field expert supervision. No manual tracking was involved.

The trajectories were smoothed by Gaussian kernel (-$3\sigma$ .. +$3\sigma$ = 12 samples).

Trajectory data is provided for use in higher-level processing to provide a shortcut for the researchers who are not able to perform tracking by themselves. RMS error in player position is approx. 0.3 meters.

*Do not treat trajectory data as gold standard – it is NOT provided for that purpose and is not sufficiently accurate.*

### 1.2.2 Annotations

Annotations were provided by sport expert by observing each of the supplied videos. They are comprised of two separated sets, the first one describes individual player activity annotations (strokes, types of strokes) and the second one describes phase of the play (rally or passive phase).

Stroke annotations also contain position of the ball at the moment of the stroke (in camera coordinates). This position was entered manually using mouse. Due to high speed shutter setting of a camera, ball is visible in most of the frames.

### 1.2.3 `Squash1.avi`

- Duration: 9 minutes, 11 seconds
- Camera: PAL CCD camera (JBL), high speed shutter
- Camera position: birds-eye view, stationary, wide angle lens
- Recorder: S-VHS videorecorder

- Digitizer: Pinaclesys DC30+ video capture card, HW MJPEG
- Initial format: 25 fps, M-JPEG, 384x576, 2 fields - interlaced
- Initial bitrate: 1504 KBit/s
- Odd fields discarded
- Transcoded to DivX 5.2.1 @ 1580 KBit/s (multipass, 3 passes)

#### 1.2.4 `Squash2.avi`

- Duration: 10 minutes, 24 seconds
- Camera: PAL CCD camera (JBL), high speed shutter
- Camera position: birds-eye view, stationary, wide angle lens
- Recorder: S-VHS videorecorder
- Digitizer: Pinaclesys DC30+ video capture card, HW MJPEG
- Initial format: 25 fps, M-JPEG, 384x576, 2 fields - interlaced
- Initial bitrate: 1504 KBit/s
- Odd fields discarded
- Transcoded to DivX 5.2.1 @ 1580 KBit/s (multipass, 3 passes)

### 1.3 Basketball subset

Basketball subset contains videos in two different quality modes. Videos depict first few minutes of the play from the slovenian basketball league.

Higher quality videos (shorter) overlap with the first partt of the lower quality videos. No annotations or trajectory data are provided.

#### 1.3.1 `basketballA.avi` and `basketballB.avi`

- Duration: 5 minutes each
- Camera: 1/2" PAL CCD camera, fixed, bird's eye view
- Camera position: birds-eye view, stationary, wide angle lens
- Recorder: S-VHS videorecorder

- Digitizer: Philips DVDR-75 DVD recorder
- Initial format: 25 fps, MPEG 2 (DVD), 720x576, 2 fields - interlaced
- Initial bitrate: 5.07 MBit/s
- Resized (bilinear) to 360x288, zero padded to 368x288.
- Transcoded to DivX 5.2.1 @ 1580 KBit/s.

### 1.3.2 `HiResBasketballA.avi` and `HiResBasketballB.avi`

- Duration: 2 minutes each
- Camera: 1/2" PAL CCD camera, fixed, bird's eye view
- Camera position: birds-eye view, stationary, wide angle lens
- Recorder: S-VHS videorecorder
- Digitizer: Philips DVDR-75 DVD recorder
- Initial format: 25 fps, MPEG 2 (DVD), 720x576, 2 fields - interlaced
- Initial bitrate: 5.07 MBit/s
- Transcoded to DivX 5.2.1 @ 4000 KBit/s, interlaced.

# 2 Reading and interpreting the data

This chapter describes the organization of the dataset, including the purpose of the individual files and the ways to load and interpret data.

Instructions often refer to Matlab functions that are provided for easy access to the data. However, dataset can be used without Matlab as well, if you write your own interface to the provided files. All Matlab functions that are used for data access for this dataset start with the prefix `CVBASE06`, followed by the name of the sport to prevent function name clashes. If you use Matlab, it may be beneficial to add directory with the dataset related funtions to your Matlab path.

All videos with annotations, positions and annotation dictionaries are in the tab separated format. First row (or first and second row in some cases) contain names of the columns.

Dataset contains videos from three types of sports. They are only briefly described. Users, unfamiliar with the rules of the particular sport and the activities that teams and players perform are advised to follow the web links to pages with more detailed explanations.

## 2.1 Handball

Handball is a well known indoor sport in Europe, and is sometimes referred to as "European Handball" or "Team Handball". It should not be confused with the different sport of the same name, played mainly in USA.

Handball is a team sport. Each team has 6 players and a goalie (a goalkeeper). It is played with a ball that is smaller than the ball used in basketball. The court is 40 meters long and 20 meters wide. The objective of the team is to score points by throwing a ball into the opponents goal. Ball is passed between players by throwing (by hand). There is a 6 meter zone around the goal that players are not allowed to enter with a ball (in other words, they have to throw before stepping into the area). After obtaining a ball (either by opponent's goal or mistake) the team goes into offensive, running across the court. After losing a ball, team quickly returns to defend its own goal. The match lasts one hour (two halves, 30 minutes each).

More about handball:

`http://en.wikipedia.org/wiki/Handball`

`http://www.usateamhandball.org/usath/sport_basics.htm`

### 2.1.1 Initializing handball subset

When using Matlab interface to the dataset, handball subset must be first initialized. In initialization phase, all supplied data except the videos is read into two Matlab variables. Subsequent access to trajectories, annotations and annotation dictionaries therefore does not require disk access and is quite fast.

**Matlab function:**

`CVBASE06HandballInit(PATH)`

String argument `PATH` should be the path to the root directory of the unpacked dataset. Please use the trailing slash (or backslash).

It will load the contents of the following files:

- `handball/PlayerList.txt` - List of the players and their numbers, tab separated.
- `handball/positions.txt` - Positions of the players for each of the frames from the video sequences. Players are denoted by their "names", which correspond to the colors of their dresses for easy visual identification. For each player, positions are provided in court coordinates and the coordinates of both cameras.
- `handball/TeamActivity.txt` - Team activity annotations. The format of annotations is as follows: first column contains the starting time of team activity (in seconds), and the second column contains the ending time of activity. Whole duration of the video should be covered by team annotations (e.g. when one activity ends, the next starts). The third column contains annotation mark, which can be decoded using the supplied annotation dictionary.
- `handball/TeamActivityDictionary.txt` - Team activity dictionary. It should be used to decode annotation marks for team activities. Matlab functions perform the translation into numerical codes automatically when the annotation files are loaded.
- `handball/PlayerActivity.txt` - Player activity annotations. The format of annotations is as follows: first column contains the time when activity was

observed (in seconds). Second column contains number of the player, which performed the activity. Player activites do not have any duration (e.g. they are observed at certain time instant). The third column contains annotation mark, which can be decoded using the supplied annotation dictionary.

- `handball/PlayerActivityDictionary.txt` - Player activity dictionary. It should be used to decode annotation marks for team activities. Matlab functions perform the translation into numerical codes automatically when the annotation files are loaded.

### 2.1.2 Accessing handball videos

Videos are stored in the `handball/` directory, and are in AVI format, compressed with DivX 5 codec (see previous chapter for details). They can be decompressed using appropriate codecs, which are videly available on Internet. See the dataset download page `http://vision.fe.uni-lj.si/cvbase06/downloads.html` for links to programs that are able to convert videos to other formats.

**Matlab function:**

```
[A,B,C,T] = CVBASE06HandballGetFrames (FRAMENUM)
```

It will load single frame from each of the handball video streams. 3-D arrays `A`, `B`, and `C` contain truecolor RGB data on return, and `T` contains time (seconds) at which frames were captured (from the frame rate and `FRAMENUM`.)

Videos A and B were recorded from the birds-eye perspective, with static camera. Video C was recorded from the side, and the cameraman was following the action.

**Matlab function:**

```
CVBASE06HandballShowFrames (FRAMENUM)
```

It will load single frame from each of the handball video streams at the frame position `FRAMENUM`, and show all three frames in one figure. Player positions will be marked by circles with player names attached to them. In many frames, players are clearly visible on videos A and B simultaneously. Therefore, blue circles denote that the other camera has been used for tracking of a particular player, and red circle denotes that the current camera has been used.

### 2.1.3 Accessing handball player positions

Player positions are stored in the tab separated file `handball/positions.txt`. First row contains the names of the players. Second row contains the descriptions of individual columns. Player positions for each player are given as pairs of coordinates in the coordinate system of the court (columns named `X` and `Y`), in the coordinate system of camera A (`XA` and `YA`) and in the coordinate system of camera B (`XB` and `YB`). Column, named `Camera` gives the information about the camera used for tracking a particular player on each frame. Value of 0 denotes camera A, value of 1 denotes camera B.

**Matlab function:**

```
[X,Y,XA,YA,XB,YB,VALIDCAM] = CVBASE06HandballGetPos (PLAYERNUM)
```

will provide position data (the trajectories) for the player `PLAYERNUM`. `X, Y, XA, YA, XB` and `YB` are vectors, one element for each frame of video. `X` and `Y` are the coordinates in court coordinate system. `XA, YA` are the coordinates in the image coordinate system of camera A, and `XB, YB` are the coordinates in the image coordinate system of camera B. `VALIDCAM` contains information whether camera A or B has been used to track a player at particular frame. `VALIDCAM` value of 0 denotes camera A and 1 denotes camera B.

**Matlab function:**

```
CVBASE06HandballShowTrajectories (PLAYERNUM)
```

will show court-coordinate system trajectories over a image of handball court. Without argument, it will show trajectories for all of the players in different colors.

### 2.1.4 Accessing handball annotations

Handball dataset provides two types of annotations. The first set of annotations describes team activity (a group activity, e.g. certain type of defense in which whole team participates). These activities span across certain interval of time. This set of annotations is provided in the tab separated file `handball/TeamActivity.txt`. The dictionary, which can be used to decode the meaning of annotations and to enumerate them is supplied in the file `handball/TeamActivityDictionary.txt`. Decoding of annotations is automatic when the provided Matlab functions are used to access the data.

**Matlab function:**

`[TIMESTART, TIMEEND, FRAMESTART, FRAMEEND, ACT] = CVBASE06HandballGetTeamActivity`

will return group (team) activity annotations in vector `ACT`, and the corresponding times (seconds) and frames when the activity was started and ended are returned in first four vectors. Activities are represented by integer numbers, which can be translated into descriptions using the following function:

**Matlab function:**

`[SHORTDESC, LONGDESC] = CVBASE06HandballTeamDict (ACT)`

will return short and long description of annotated team activity, supplied as (scalar!) `ACT`. Short description is the annotation which was coined by sport experts to use in annotation files. Long description is more informative.

**Matlab function:**

`[TIME, FRAME, ACT] = CVBASE06HandballGetTeamActivityEx`

will return group (team) activity annotations in expanded form. Annotations are returned as vector `ACT`, the corresponding times (in seconds) for each element of `ACT` is returned in vector `TIME`, and the corresponding frames in vector `FRAME`. NOTE: The resulting vectors are of same size than the player position vectors, and the number of elements corresponds to the number of video frames in video streams. This is essentially the same information than provided by `CVBASE06HandballGetTeamActivity`, but differently organized.

The second set of annotations describes the activities of individual players during the match (e.g. passing a ball, throwing at a goal, etc.). These activities are observed in certain instant of time and do not span across interval of time. This set of annotations is provided in the tab separated file `handball/PlayerActivity.txt`. The dictionary, which can be used to decode the meaning of annotations and to enumerate them is supplied in the file `handball/PlayerActivityDictionary.txt`. Decoding of annotations is automatic when the provided Matlab functions are used to access the data.

**Matlab function:**

`[TIME, FRAME, ACT] = CVBASE06HandballGetPlayerActivity (PLAYERNUM)`

will return individual player activity annotations in vector `ACT`, and the corresponding time (seconds) and frame when the activity was annotated in vectors

`TIME` and `FRAME`. Activities are represented by integer numbers, which can be translated into descriptions using the following function:

**Matlab function:**

```
[SHORTDESC, LONGDESC] = CVBASE06HandballPlayerDict (ACT)
```

will return short and long description of annotated player activity, supplied as (scalar!) `ACT`. Short description is the annotation which was coined by sport experts to use in annotation files. Long description is more informative.

## 2.2 Squash

Squash is a racquet sport, played indoors in closed court (walls are integral part of the court, since the ball bounces off the walls). Two players are present in a court during the match. After the serve, the opponent must run to the ball and strike it before bounces off the floor for the second time. The ball has to bounce off the front wall inside certain area after the stroke.

The match is comprised of sets (similar as in tennis), and the player who wins more sets is declared a winner of the match. Each set is comprised of rallies, which represent "active phase" of the play. Rally starts with a serve. Rally ends when player fails to strike the ball in time, and the point (or/and the serve) is awarded to the opponent. It can also end for other reasons ("let" is called, etc.) The intervals between the rallies are called passive phases - the players generally move slower and prepare for a next rally, which will start with a next serve.

More about squash:

`http://en.wikipedia.org/wiki/Squash_(sport)`

`http://www.worldsquash.co.uk/rules.html`

### 2.2.1 Initializing squash subset

When using Matlab interface to the dataset, squash subset must be first initialized. In initialization phase, all supplied data except the videos is read into two Matlab variables. Subsequent access to trajectories, annotations and annotation dictionaries therefore does not require disk access and is quite fast.

Differently from handball subset, the two squash videos and the corresponding annotations are independent of each other. They represent different sets in different matches on different tournaments. Matlab interface functions reflect that - the number of the requested set (1 or 2) has to be passed to most of the functions.

**Matlab function:**

`CVBASE06SquashInit(PATH)`

String argument `PATH` should be the path to the root directory of the unpacked dataset. Please use the trailing slash (or backslash).

It will load the contents of the following files:

- `squash/squash1positions.txt` and `squash/squash2positions.txt` - Positions for both of the players for each of the frames from the corresponding video sequence. Columns `X1` and `Y1` contain the court coordinates of the player 1, and columns `X1Cam` and `Y1Cam` contain the positions in the camera coordinates. `X2, Y2, X2Cam` and `Y2Cam` contain same information for the second player. Additionally, this file contains the information about the phase of the play for the each frame in the column `Phase`. Value of 1 denotes rally (active phase) and 2 denotes passive phase. To decode phase annotations, the following file can be used:

- `squash/PhasesDictionary.txt` - Play phases dictionary. It could be used to phase numbers back into descriptions.

- `squash/Squash1PlayerActivity.txt` and `squash/Squash2PlayerActivity.txt` - Player activity annotations. "Player activities" consists of ball strikes, including serves. Each row contains one strike, with the time and frame when the activity was observed. The column `StrokePlayer` contains the number of the player who performed the activity. Following information is also provided: type of the stroke (column `StrokeType`), outcome of the stroke (`StrokeOutcome`) and the information whether the stroke was forehand or backhand (`ForehandBackhand`) column. Stroke types are described in more detail at the end of this section. Outcome of the stroke is most often the continuation of the play, error of the player or point achieved.

- `squash/ShotTypeDictionary.txt` - Shot type dictionary. If you use Matlab function for dataset access, the annotations will be translated into numbers automatically.

- `squash/ShotOutcomeDictionary.txt` - Shot outcome dictionary. If you use Matlab function for dataset access, the annotations will be translated into numbers automatically.

- `squash/ForehandBackhandDictionary.txt` - Shot type (forehand or backhand) dictionary. If you use Matlab function for dataset access, the annotations will be translated into numbers automatically.

### 2.2.2 Accessing squash videos

Videos are stored in the `squash/` directory, and are in AVI format, compressed with DivX 5 codec (see previous chapter for details). They can be decompressed using appropriate codecs, which are videly available on Internet. See the dataset download page `http://vision.fe.uni-lj.si/cvbase06/downloads.html` for links to programs that are able to convert videos to other formats.

**Matlab function:**

```
[A,T] = CVBASE06SquashGetFrame (SETNUM, FRAMENUM)
```

will load single frame from the specified squash video s stream. Video stream number (set 1 or set 2) is specified in `SETNUM`. 3-D array `A` contains truecolor RGB data. `T` contains time (seconds) at which frame was captured (from framerate and `FRAMENUM`.

**Matlab function:**

```
CVBASE06SquashShowFrame (SETNUM, FRAMENUM)
```

It will load single frame `FRAMENUM` from the squash video specified by `SETNUM`. Player positions will be marked by circles of different colors. If the specified frame contains player activity annotation (stroke), the position of the ball will be marked with triangle, along with other annotation data.

Try the following example:

```
CVBASE06SquashShowFrame(1,63);
```

**Matlab function:**

```
CVBASE06SquashReplayActivity (SETNUM, PLAYERNUM, ACTIDX)
```

will sequentially load 11 frames from the dataset `SETNUM` at position defined by `PLAYERNUM` and `ACTIDX` and play them on sceen. Player positions are marked with circles, and position of the ball at the moment of annotation is marked by triangle, along with other annotation data. `ACTIDX` is index into the vector of activities (e.g. `ACTIDX` = 1 means first annotated activity for player `PLAYERNUM`, `ACTIDX` = 2 means second, etc.)

Try the following example:

```
CVBASE06SquashReplayActivity(1,1,1);
```

### 2.2.3 Accessing squash player positions

Player positions are stored in the tab separated files `squash/squash1positions.txt` and `squash/squash2positions.txt`. The files contain positions of both of the players, both in court and camera coordinate systems. Additionally, this file contains play phase annotations.

**Matlab function:**

```
[X,Y,XC,YC] = CVBASE06SquashGetPos (SETNUM, PLAYERNUM)
```

will extract positions (trajectories) for the player `PLAYERNUM` (possible values: 1 or 2) in the set `SETNUM` (possible values: 1 or 2). `X, Y, XC` and `YC` are vectors, their length is equal to the number of valid frames frames (1 element per frame). `X` and `Y` are the positions in the court coordinate system; `XC` and `YC` are the positions in the camera coordinate system. Please note that videos can extend several seconds past the provided player position data; those video frames should be discarded.

**Matlab function:**

```
CVBASE06SquashShowTrajecties (SETNUM, PLAYERNUM)
```

will show court-coordinate system trajectories over a image of squash court. Without the second argument, it will show trajectories for both players in different colors.

### 2.2.4 Accessing squash annotations

Squash dataset provides two types of annotations. The first set of annotations describes phase of the play (passive phase vs. rally). These activities span across certain interval of time. This set of annotations is provided as separate column in the tab separated files `squash/squash1positions.txt` and `squash/squash2positions.txt`.

Second type of annotations describes player activities and contains the moments when either player strikes a ball. These activities have no duration. They are provided in files `squash/Squash1PlayerActivity.txt` and `squash/Squash2PlayerActivity.txt`.

**Matlab function:**

```
ACT = CVBASE06SquashGetPhasesEx(SETNUM)
```

will return phase (rally or passive phase annotations) in vector `ACT` for the set `SETNUM`. NOTE: The elements of the resulting vector correspond to the video frames in video streams and to elements of player position vectors.

**Matlab function:**

```
[TIME, FRAME, STYPE, SOUTCOME, SFB, SX, SY] =
CVBASE06SquashGetPlayerActivity (SETNUM, PLAYERNUM)
```

will return individual player activity annotations (strokes). Vector `STYPE` will contain stroke type. Decode it using function `CVBASE06SquashShotTypesDict` or the corresponding dictionary `Squash/ShotTypeDictionary.txt`.

Vector `SOUTCOME` will contain stroke outcome. Decode it using the function `CVBASE06SquashShotOutcomeDict` or the corresponding dictionary `Squash/ShotOutcomeDictionary.txt`.

Vector `SFB` will contain type of stroke (forehand or backhand). Decode it using function `CVBASE06SquashFBDict` or the corresponding dictionary `Squash/ForehandBackhandDictionary.txt`

Vectors SX and SY will contain coordinates of the ball in the camera coordinate system at the time the shot was observed. Do not mix this with the coordinates of the player at the same time!

Corresponding time (seconds) and frame when the activity was annotated is contained in the vectors `TIME` and `FRAME`.

NOTE: The resulting vectors of this function are far shorter than when retrieving player trajectories, since they contain only the times when certain activity was observed. One cannot retrieve player activity by addressing `STYPE(currentframe)`!

Corresponding dictionaries for all annotations are provided in separate files and can be acessed through the following functions:

**Matlab function:**

```
[SHORTDESC, LONGDESC] = CVBASE06SquashPhasesDict (PHASE)
```

**Matlab function:**

```
[SHORTDESC, LONGDESC] = CVBASE06SquashShotTypesDict (ACT)
```

**Matlab function:**

```
[SHORTDESC, LONGDESC] = CVBASE06SquashShotOutcomeDict (ACT)
```

**Matlab function:**

```
[SHORTDESC, LONGDESC] = CVBASE06SquashShotFBDict (ACT)
```

### 2.2.5 Types of strokes and their meaning

The squash subset annotations provide a detailed classification of the type of the stroke for each stroke by either of players.

Following figures illustrate the meaning of the annotations. The path of the ball before hittng the front wall is drawn in solid line. The path after hitting the front wall is drawn in dashed line. Most of the stroke types have their "volley" variant, which means that the ball did not touch the ground before the stroke.

## 2.3 Basketball

Basketball is a well known team sport. More about basketball can be found at:

`http://en.wikipedia.org/wiki/Basketball`

### 2.3.1 Initializing basketball subset

When using Matlab interface to the dataset, basketball subset must be first initialized. In initialization phase, path to the videos is stored into Matlab memory. Since there are no trajectories or annotations provided, nothing else is loaded.

**Matlab function:**

```
CVBASE06BasketballInit(PATH)
```

String argument `PATH` should be the path to the root directory of the unpacked dataset. Please use the trailing slash (or backslash).

### 2.3.2 Accessing basketball videos

Dataset provides basketball videos in two formats: in the "standard quality" format (e.g. similar to handball and squash videos) and in the "high quality" for-

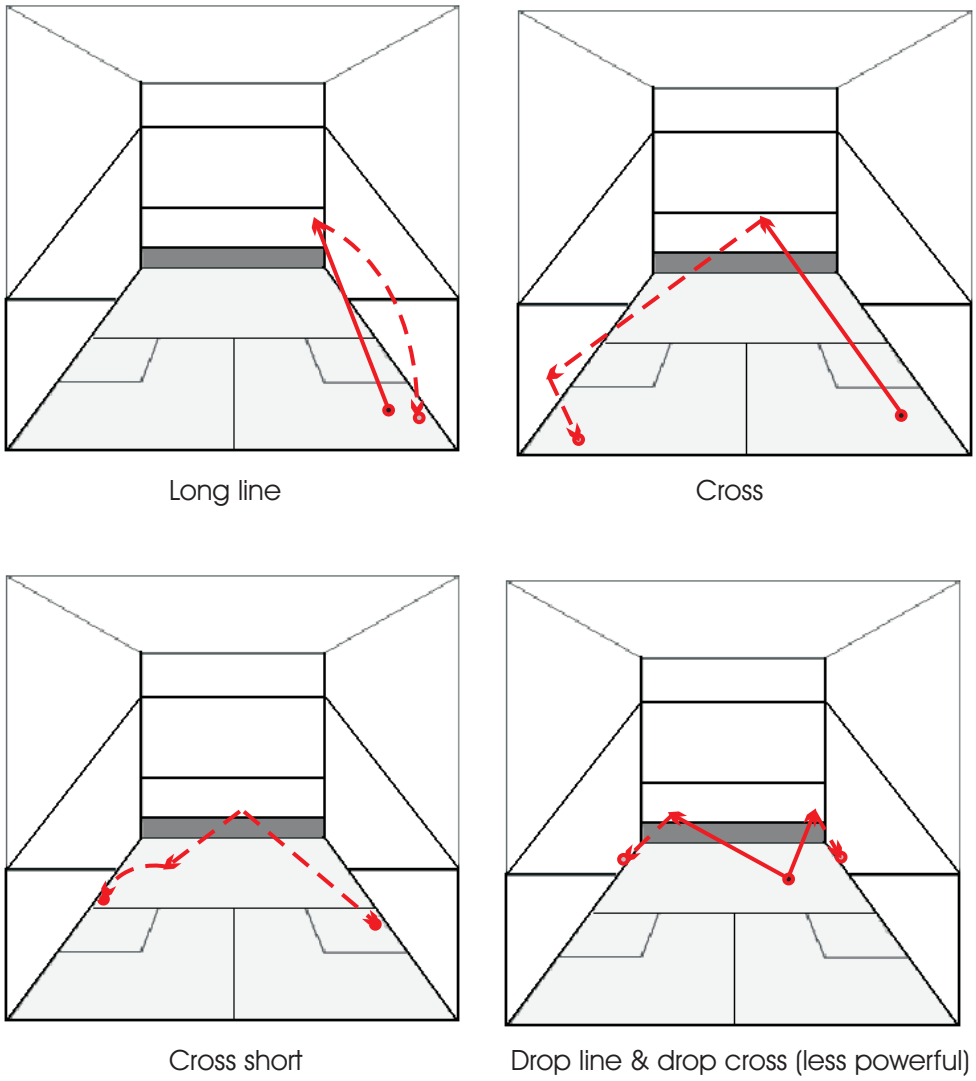


Figure 2.1: *Squash stroke types, 1/3*

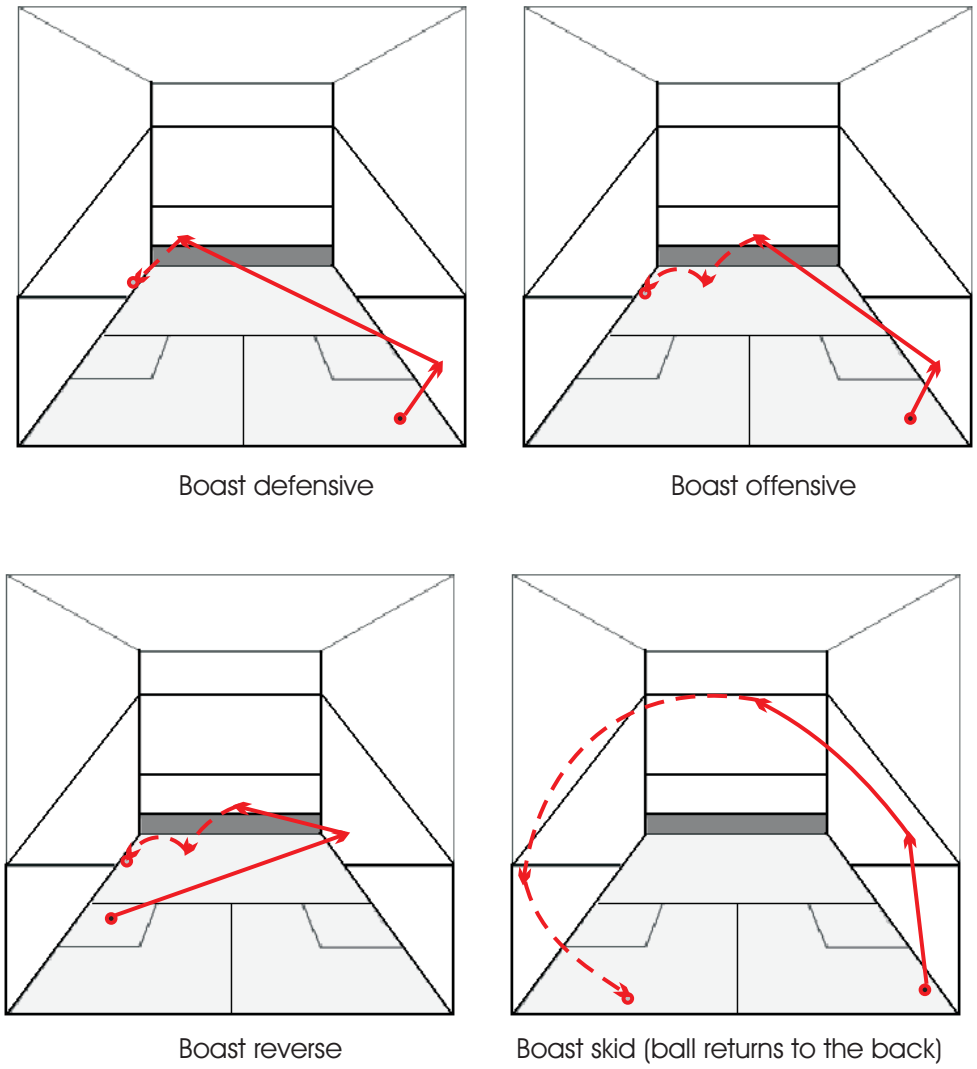


Figure 2.2: *Squash stroke types, 2/3*

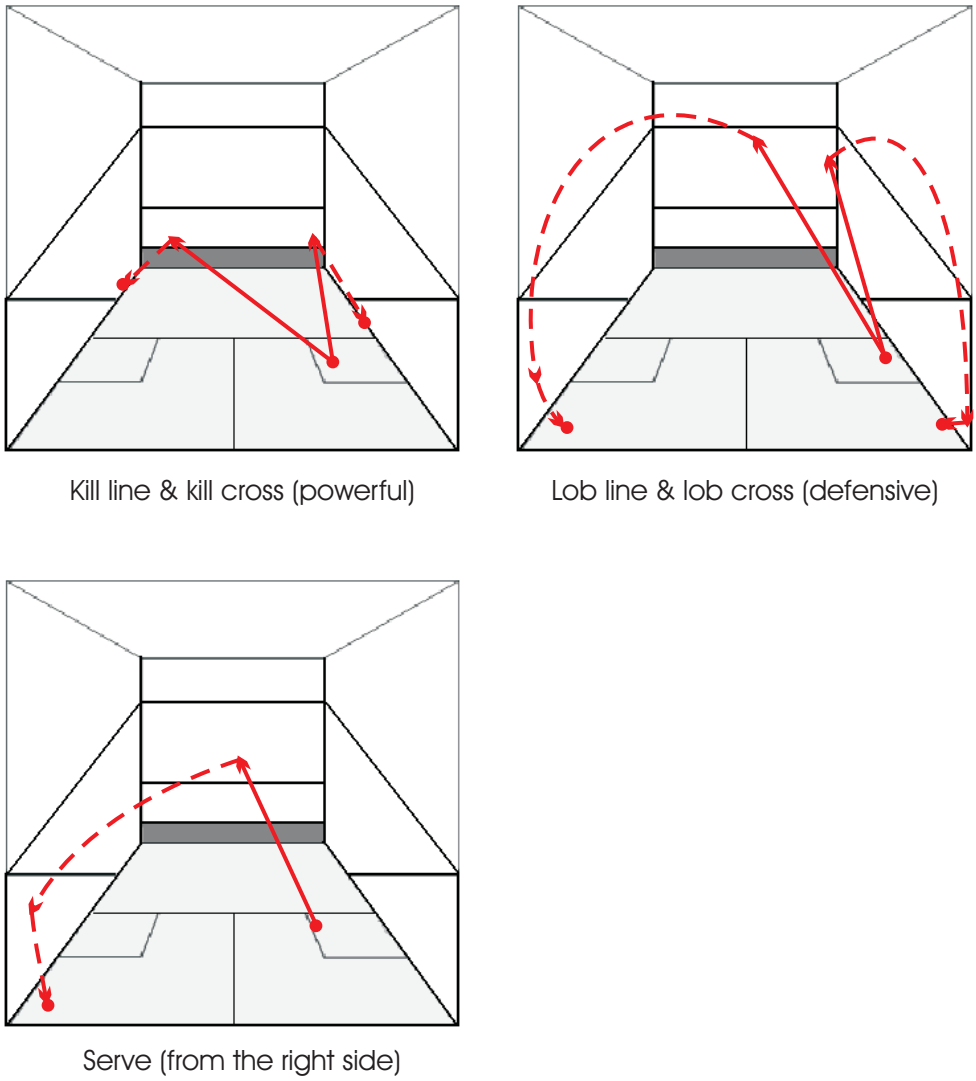


Figure 2.3: *Squash stroke types, 3/3*

mat, which provides double resolution in both directions and significantly higher bitrate.

High quality videos are shorter than the low quality counterparts, but they originated from the same high quality source. Therefore, hey entirely overlap with the (longer) lower quality videos.

Similar to handball, there are two birds-eye perspective videos (A and B), that, when combined, cover the whole court surface. Videos depict single match and are synchronized.

**Matlab function:**

`[A,T] = [A,B,T] = CVBASE06BasketballGetFrames (FRAMENUM)`

will load single frame from each of the basketball video streams from the standard quality basketball set. 3-D arrays `A` and `B` contain truecolor RGB data. `T` contains time (seconds) at which frames were captured (from framerate and `FRAMENUM`).

**Matlab function:**

`[A,B,T] = CVBASE06BasketballGetFramesHiRes (FRAMENUM)`

will load single frame from each of the basketball video streams from the high quality basketball set. 3-D arrays `A` and `B` contain truecolor RGB data. `T` contains time (seconds) at which frames were captured (from framerate and `FRAMENUM`).

For the difference between the quality settings for both sets of video streams see the first chapter of the manual.

**Matlab function:**

`[A,B,T] = CVBASE06BasketballShowFrames (FRAMENUM)`

will load single frame from the each of the standard quality basketball video streams and display them on screen.

# 3 Support

Authors cannot guarantee any kind of support for the provided dataset. However, if you think that you found an error in the dataset (either in the data or in the Matlab code) you are welcome to report it to `janez.pers@fe.uni-lj.si`.