

Introduction

The automatic labeling of segments is based on computations using only link lengths (separations between markers). No other application specific information is required or used, resulting in an extremely general method. The entire process is incorporated into the **Identify** program which allows the user to define links and perform the labeling interactively.

Every effort has been made to make the process as simple and intuitive as possible but the underlying complexity of the problem imposes limits on what can be achieved without sacrificing generality. As a result the effective utilization of this capability will require some effort on the part of users to understand the principles involved, and require them to learn a number of commands.

The current implementation is still evolving because the techniques and procedures described in this documentation are novel and are based on limited experience with limited data sets. Users are encouraged to experiment with applications to their own data sets and share their experiences with the AMASS user community.

Applicability

The techniques described here are designed to facilitate the association of segments with marker labels and make it a less manually intensive process. Effectiveness will depend on the quality of the .seg data and a number of factors related in complex ways to the links that the user defines between markers.

The data base for the automatic labeling consists of a list of links (pairs of marker labels plus their separations) between markers whose separations are expected to be more-or-less constant throughout the .seg file, i.e. the process is most effective if the links are of constant or near constant length. A link may be defined between any pair of segments by simple mouse clicks on two points, but the effectiveness of a link in aiding segment identifications will depend on a number of factors. On the other hand the method is completely general and does not depend on any other input pertaining to the marker configurations.

Definitions

- ****Link**** A link is a defined connection between two markers. It consists of the labels of the two markers, a nominal separation of the markers in user distance units, the maximum and minimum separations of the two markers, and a three component RGB color value that is to be used to draw the link in the **view area**.
- ****User Identified segments (UIDs)**** During the identification process segments are associated with markers identified by their labels. Because computed identifications may be incorrect the software distinguishes between identifications performed by the user and those computed by the program. **UIDs** are segments that have been manually identified by the user, or have been confirmed by the user as being correct.

UIDs are always indicated in yellow in the **label**, **view**, and **continuity areas**, and in contrast to Computer **IDentified segments (CIDs)** the program will never automatically change their

identifications.

- ****Computer Identified segments (CIDs)**** Segment identified by the program will be called **CIDs** and they will always be indicated by green lines in the **continuity area**, green points in the **view area**, and green backgrounds in the **label area**, and they may be modified by the program. The user may confirm **CIDs** and convert them to **UIDs** by use of the **Confirm frame** or **Confirm all** (frames) buttons.
- ****Link systems**** It is useful to distinguish between several different kinds of link systems.

Body A body is defined as a collection of markers where every marker is connected by a link to at least one other marker. Computationally the program treats each body as a unit, i.e. it processes each body in turn. Bodies with many links and unidentified makers may take a long time to process.

Chain The simplest body is a chain of links where **n** markers are connected by **n-1** links to form a simple chain. We do not treat the case where **n = 2** because there is no way the data from a single link can be used to distinguish between two markers. Hence the simplest body consists of three markers connected by two links. Of course if these links are of the same length then we still cannot distinguish the two end markers. A chain provides the least data for carrying out the marker identifications. **Fully connected body** A fully connected body is a collection of markers where each marker is connected to every other marker in the body. In this case we have **n** markers connected by **n.(n-1)/2** links providing the maximum amount of data for the identification of the markers. Note that defining every link for such bodies with a larger number of markers (> 5) is somewhat redundant and may increase computation time without measurable benefit. **Composite body** A composite body is a most general type and contains a mixture of chains and fully connected bodies connected by links.

Identify buttons

This section briefly describes the buttons in the **command area** that are used to identify segments using link information.



- ****ID frame**** This button switches to a mode whereby each frame selected by the frame cursor is processed to compute segment identifications using the current link information. Each frame is treated independently of all other frames but all **UIDs** are preserved, i.e. the **CIDs** from any previous computation are first cleared. All resulting **CIDs** are shown in green.
- ****ID all**** The program attempts to identify the segments over all frames using the current link information, the **UIDs**, and new **CIDs**. In contrast to the **ID frame** mode **CIDs** are propagated to subsequent frames and an incorrect identification can cause errors in subsequent frames.
- ****Confirm frame**** This button will convert **CIDs** to **UIDs** (from green to yellow) but only for the current frame, i.e. only the green points shown in the **view area**, and under the red frame cursor in the **continuity area**. **Note:** As a convenience the **Enter** key also performs this function.
- ****Confirm all**** This button converts all (not just for the current frame) **CIDs** to **UIDs** (from green to yellow).
- ****Scale**** This spin box allows the user to temporarily scale all link lengths in the program memory, and is primarily useful for applying a link file to different sized subjects.

Link buttons

The **LINKS** mode button enables/disables alteration of the link data, and must be enabled if you wish to modify the link data that is currently resident in program memory.



- ****Clear**** Clicking this button deletes all link data currently in memory.
- ****Reload**** This button first deletes all links in memory and then reloads the links from the file whose name is stored in the current project file. To load links from a different link file use the menu **File/Load links...** command.
- ****Set**** This command re-computes the data for all links using the current UIDs. You must first delete or confirm any CIDs before using this command.
- ****Extend**** Use this button to extend the link length limits defined previously by incorporating lengths from UIDs in the current `.seg` file.
- ****Save**** This button becomes enabled once **Set** (links) has been performed. It saves the link data to the Project link file. To save to a different file use the menu **File/Save links as...** command.

Menu commands

The following commands pertaining to links are available from the main menu bar.

- ****File/Load links ...**** This item allows the user to load a link file other than the project link file specified in the Identify parameters tab. It will present an **Open file** dialog that allows the selection of the input file.
- ****File/Save links as ...**** If you wish to save the links to a file other than the project link file use this command. It will present a **Save file** dialog box.
- ****Edit/Links**** This command writes the current link data in memory to a temporary ASCII file and opens the file in a Notepad window to allow the user to view it or manually edit it. If the edited file is closed with a **Save** the link data in memory is replaced with the saved data. This facility should be used with caution as erroneous input will compromise the program.
- ****Edit/Link color**** The default color for drawing links is initially set to red but can be changed by selecting this item from the main tool bar. The selected color will then be applied to all the links that you create.
- ****View/Link vars.**** This command causes the links in the view area to be drawn in colors depending on how much each link varies in length (by percentage). It is useful for assessing link variations in a link file, and can be useful for finding an incorrectly defined link. Approximate color assignments are as follow:

Link varies by +/- 20% or more, Red +/- 15% Yellow +/- 10% Green +/- 5% Aqua +/- 0% Blue

Links

This section describes links in detail.

Defining links

Links may be created any time the **LINKS** mode is selected, and they may be defined in conjunction with segment identifications.

Use the **Left** mouse button to create a single link to a previously **Left** clicked point, and the **Right** mouse button to create multiple links with the previously **Right** clicked points.

To define a chain of links for points that have already been identified do the following:

- **Left click** on the first point.
- **Left click** on the next point, etc.
- To cancel the link creation, **Left click** for a second time (not a double-click) on the last body point.

To connect a set of points with all possible links and create a fully connected body:

- **Right click** on the first point.
- **Right click** on the next point, etc.
- To cancel the link creation, **Right click** for a second time (not a double-click) on the last body point.

For a rigid body of n points, a maximum of $n \cdot (n - 1) / 2$ links can be specified. Defining all links for bodied with a larger number of points (> 5) is somewhat redundant and may increase computation time without measurable benefit.

It is possible to use a combination of **Left** and **Right clicks** to create linkages for composite bodies in a single pass through the points.

Two bodies may be combined into a single body simply by linking one or more points in the first body with one or more points in the second body. Note however that it is better to use a larger number of smaller bodies. In some situations large bodies can make the computation times excessive.

In order to identify points in conjunction with link creation first select marker labels in the label area in the order in which they will be applied to unlabeled points, then proceed to create the links as described above. The selected marker labels will only be assigned when you click on unidentified points, and you may include already identified points freely during the sequence.

Deleting links

To delete a particular link it must be visible in the **view area** display. Move the mouse cursor so that it is near the link and then **Left double-click** while holding down the **Shift** key.

All links in memory may be deleted by use of the **Clear** (links) button.

You may also edit the links currently in memory by selecting **Edit/Links** from the main tool bar. This opens a **Notepad** window for editing and allows you to delete any line (which represents a link). When the **Notepad** window is closed with a **Save** the updated version is read into memory and applied by the program.

Changing link color

The user can select the color with which an individual link is drawn. To change the color of a visible link move the cursor to the vicinity of the link and **Left single-click** while holding down the **Shift** key. This will cause the color picker window to appear and allow you to choose either a standard color or a custom color for that link. The default color for drawing links is red, but can be changed by selecting **Edit/Link color** from the main tool bar. From then on the selected color will be assigned to all links that you create.

Graphing link lengths

You may use the capability described in **Graphing link lengths** to graph the length of any defined link.

Setting links

When a link is first created only its length at the current frame is known, and its minimum and maximum lengths are set at +/- 10% of its current length. These data are then used to calculate other identifications, but typically the initial estimated values will not be optimal. At any time the **Set** (links) button may be used to update the link data using all available UIDs to improve the accuracy and efficiency of the identification process. As a minimum you should use the **Set** (links) button to finalize the link data for a trial that you have identified and wish to save for use on additional .seg files.

Extending links

This command updates the current link length limits held in memory by incorporating the link lengths available from the current data. It can only decrease the minimum lengths and increase the maximum permitted lengths of links, and as such will mostly have an effect on the **ID all** function.

Saving links (Save)

Once **Set** (links) has been activated the **Save** (links) button becomes enabled allowing the link data to be saved to the project link file and used for identifying similar .seg files.

The link data may also be written to a different file by selecting **File/Save links as...** in the main menu bar.

If you use the **Save** (links) button and have not specified a project link file you will be presented with

the **Save links as...** option, and the opportunity to designate it as the project link file.

Editing links

The **Edit/Links** selection from the main menu bar writes the current link data to a temporary file and opens the file in a **Notepad** window to allow the user to manually edit it. If the edited file is closed with a **Save** the link data in memory is replaced with the saved data. This facility should be used with caution as erroneous input will compromise the program.

Link file format

The link file is an ASCII file that contains a comment line as the first line, followed by a line of data for each defined link. A line describing a link consists of:

- The label for the first marker
- A hyphen
- The label for the second marker
- A minimum distance between the two markers
- A mean distance between the markers
- A maximum distance between the two markers
- An RGB color value in the format (red, green, blue) where the color components are in the range 0 - 255

The numerical items must be separated by spaces. An example of a link file is:



Note that the minimum and maximum distances used are not true values but may be modified by the program to remove outliers caused by measurement noise.

Identify tools

The techniques for using links to aid in identification are still evolving, but described here are some methods that seem to work quite well, at least for the data sets on which testing has been carried out. The computation to identify segments based solely on distances is very complex and the outcome depends on many factors, some of which are:

- The total number of markers
- The types of bodies
- The number of links we are trying to fit to the markers
- The number of bodies we are trying to identify
- The body linkage configurations
- The continuities of the marker segments
- The variations in link lengths over all frames
- How similar are the link lengths to each other

Here are some general guidelines for creating useful link systems.

- A body must contain at least 3 markers
- Define bodies with smaller numbers of markers to avoid excessive computation times
- Avoid creating links whose lengths vary a lot
- Bodies with multiply connected markers are better than simple chained bodies because there is more data for the identifications
- Avoid creating symmetric objects, although a single manual marker identification will usually resolve a symmetry issue
- Situations with many links of approximately the same length may require excessive computation times
- The program is capable of distinguishing very subtle differences in link lengths provided they remain constant.

The techniques incorporated in the **Identify** program work very reliably for configurations where the link lengths are kept constant by attaching the markers to rigid objects. They become progressively less effective for links whose lengths vary substantially throughout the measurement. Because of the computational complexity it is difficult to predict how well the program will work in any given application.

The link methods implemented in **Identify** are very useful for identifying segments in multiple trials of the same or similar actions where a common link file can be applied. The techniques are also useful for single “one-of” trials, especially those containing many segments caused by obscuration of markers during the data capture. In fact in some situations it may be useful to turn off the segment joining capability in the **Track** program because the easy identification of many more segments is possible and may provide cleaner results.

The following subsections describe in detail the main tools and methods available for identifications based on link length information.

The ID frame mode

After initially defining or loading the links you will typically start the identification process by clicking on the **ID frame** button to put the program into the **ID frame** mode. This button may be toggled between on and off.

In the **ID frame** mode the program attempts to calculate the segment identifications for the current frame indicated by the frame cursor. If you move the frame slider the identification computation starts only when you release the mouse button, allowing you to first position it at any desired frame.

The calculation uses the point coordinates and existing UIDs present in that frame, but does not use any CIDs that may have been computed previously, i.e. each frame is treated independently and the identifications are performed without regard to CIDs found in any other frame. On the other hand UIDs (in yellow) may only be modified by the user and are never altered without user direction. The resulting segment identifications are presented as CIDs and are drawn in green in the **view**, **continuity**, and **label** areas.

Depending on the data, quite frequently the computed identifications may be incorrect, especially if there are a number of links of similar lengths, the link lengths have large variations throughout the trial, or markers are missing from the view.

The strategy should be to search for frame positions where all or most CIDs are calculated correctly. If all CIDs in a frame are correct you can confirm them and convert them to UIDs (by use of the **Confirm frame** button or **Enter** key), or if some are incorrect you can first individually correct them.

In some situations the computation for a frame can take a long time. This is more likely in cases where a body has a large number of markers, a number of links are of similar length, or the link lengths vary greatly. A simple solution is to manually identify a point or two in the body. This will also work if there is symmetry that was not resolved correctly by the program.

The **ID frame** mode can be cancelled by clicking again on the **ID frame** button, clicking on the **ID all** button, or clicking on the **Set** (links) button.

ID all

The **ID all** button causes the program to try and identify all segments over all frames using the currently available link data and UIDs. The button only becomes enabled when the **ID frame** mode is on, and when clicked it

- Turns off the ID frame mode
- Confirms all CIDs in the current frame
- Cancels all other CIDs
- Attempts to compute identifications over all frames

You should first ensure that all CIDs for the current frame are correct since the computation uses the current frame as a starting point.

This computation may or may not produce useable results depending on the quality of the link length data and the marker data. It builds upon CIDs which may have errors that propagate throughout the file.

The program attempts to identify segments starting at the current frame and proceeds, in five frame increments, to the last frame or the last frame selected for output. In contrast to the **ID frame** mode CIDs are retained and used to help the identifications in subsequent frames. Erroneous CIDs are propagated to succeeding frames unless a link length exceeds its permitted limits at which point the identifications for the segments forming the link are cancelled. Once the end frame has been reached the program returns to the initial frame, and in five frame increments, proceeds backwards to the first frame in the file or the first frame selected for output.

The results may range from perfect to very poor depending on a number of complex interacting factors. In any case the action can be easily undone by clicking once on the **Clear IDs** button.

Caution: if no CIDs were found the Clear IDs button will cancel all segment identifications (that is UIDs).

If there are just a few incorrect CIDs you can cancel them or correct them before doing the confirmation, otherwise it is usually best to go back to the **ID frame** mode and either start **ID all** from a different frame, or identify more segments using the **ID frame** mode. Note that clicking on the **ID frame** button automatically cancels all CIDs.

Confirm frame (Enter key)

Any time all CIDs are correct in the currently displayed frame you can click on the **Confirm frame** button (or press **Enter**) to convert the CIDs to UIDs which will then add to the pool of data used to make additional identifications. Only the CIDs under the frame cursor and those shown in green in the label and view areas will be affected.

Confirm all

This command will typically be used after an **ID all** command and results in the conversion of all CIDs (drawn in green in the continuity area) to UIDs. Before using this command you should scan the file over all frames and make corrections as necessary to ensure that all CIDs are correct. Incorrect UIDs will typically compromise all successive identification computations.

Scale

The Scale value can only be modified when the program is in the **ID frame** mode and the **LINKS** mode is switched off, and it is most easily changed by placing the mouse cursor over the box and rotating the mouse wheel. The indicated scaling is temporarily applied to the link lengths when doing the identification computations, and is designed to enable the links defined for one application to be applied to another application where the subject (or object) may have a different size compared to the original. Note that the scaling does not alter the defined link length data. You can only modify the link length data by creating UIDs and then using the **Set** or **Extend** commands.

Clear IDs

The action of this button has been modified to provide two levels of functionality. If any CIDs are present (indicated by green in the **continuity area** and green points in the **view area**) then clicking on this button will delete all of the CIDs and leave UIDs unchanged. A second click of the button will then delete all UIDs (yellow). **Caution:** If there are no CIDs, then the first click of the button will delete all UIDs.

Canceling assignments

Both CIDs and UIDs for a point may be cancelled by **double Left-clicking** on the point in the **view area**. Additionally they can be cancelled by placing the mouse cursor near the point so that the marker label shows and then pressing the **Delete** key.

For CIDs another method may be used. Selecting a CIDs label in the label area cancels the identification and also selects the label for use in the next identification to be applied. To cancel the identifications of a number of CIDs it is possible to select their labels in the **label area** and then cancel them all with a **double Left-click** within the **label area**, or by changing the displayed frame with the frame slider or mouse wheel.

Transferring computed assignments

You can select any number of labels belonging to CIDs in the **label area** (ones with a green background) just as you can for unassigned labels, and then apply them to unlabeled points in the normal manner. As you select each label the identification of the segment having that label will be cancelled.

The first label you select will have its background changed to red, indicating that the label will be applied to the next selected unidentified point in the **view area**. In the **view area** the program will draw a red circle around the points whose segments do not overlap segments already given that label and can accept that label without conflict.

The assignment can actually be made to any other unidentified point, however you will be presented with a message that there is a conflict in the assignment, and asked to resolve it.



In this case you can cancel the assignment (**Cancel**), apply the assignment and cancel the identification of the overlapped segment (**Reassign**), or apply the assignment and retain the overlapped assignment (**Assign also**). The last option should only be used in situations where the frame overlap is caused by an error in joining of segments during the tracking, and involves the overlap of at most a few frames.

Set (links)

The ability of the program to identify marker segments is highly dependent on accurate link length data which is usually not available when the link is first defined. When a link is first created the program records its length at the current frame and estimates its minimum and maximum lengths which are then used to identify other segments. At any time we may use the Set (links) command to have the program use all existing UIDs to improve the accuracy of the length data for all links defined so far.

Typically **Set** (links) should be used to better define the link length data after segments have been identified at several locations throughout the file (to better define the link length variations), and before a call to **ID all**. It should also be used after all markers have been identified and before saving the link data to a link file.

Note that before using the **Set** or **Extend** command you must first confirm or cancel the identifications of any CIDs.

Set vs. Extend

The **Set** command measures the current UIDs to define the link lengths and limits. The **Extend** command measures the same UIDs but modifies the currently defined lengths to include the new data. If you create new links in a file then **Extend** has the same function as **Set**. The **Extend** function will “extend” the link data created with another .seg file to include the lengths present in the current .seg file. Note that increases in link length limits may make the identification capability less effective.

Identifying without a link file

The suggested procedure for creating a link file or identifying a one-of trial is essentially the same, and is described in this section. It is difficult to prescribe a specific sequence of actions to be used to identify any one .seg file, hence we describe what actions are available, provide some general guidelines in the this section, and leave it to users to experiment and refine their own techniques.

Select frames

You may first wish to set the frame limits to avoid computations in part of the file where the data is poor. To set the first relevant frame move the frame slider to the desired position and press the **Home** key. To set the last frame use the **End** key after moving to the last desired frame.

Identify markers

The next step is to move the frame slider to a location in the file where most markers are visible and can be readily identified. Identify the markers in your first body by selecting the appropriate labels in the **label area** and then clicking on the points in the **view area**.

Create the links

Next click the **LINKS** mode button to allow the creation of links, and then create the links for the body by Left and/or Right clicking on the identified segments as described in [Defining links](#) section. Note that the identification and link creation actions may be combined. At this point you may proceed to define links for other bodies or process the just created body.

Use the ID frame mode

Once you have created the desired links click on the **ID frame** button and move the frame slider to a new position (you can also use the mouse wheel to do this). Once the mouse button is released the program will attempt to identify segments in the current frame with the marker labels that form the body. The program will display computer identified segments (CIDs) in green.

The strategy should be to move the frame slider to positions where CIDs are calculated correctly, and then confirm the assignments by use of the **Confirm frame** button (also the **Enter** key) to change the green CIDs to yellow UIDs.

Correct the identifications

However even the best frames may have CIDs that are incorrect, especially if there are a number of links of similar lengths, the link lengths have large variations throughout the trial, or markers are

missing from the view. In such situations you should manually cancel and/or modify assignments incorrect assignments, or just move the frame cursor to a different section of the file.

The initial goal should be to have UIDs for all markers you are labeling at several locations over the frame range that is of interest so as to incorporate the full length range of each link. Actually, if the markers are firmly attached to a rigid body then any frame supplies this information except for measurement noise.

Once you have good identifications at several locations in the file, you can proceed with the **Set (links)** command.

Set links

Use the **Set (links)** button to have the program update the link length data using all available UIDs.

Use ID all

Once you have refined the link length data you can use the **ID all** command to have the program try to identify segments over all frames. After the calculation scan all frames by moving the frame cursor and check the correctness of the CIDs. If all CIDs are correct you can click the **Confirm all** button and then direct your attention to any segments that are still not identified. If only a few CIDs are in error you can correct them before doing the confirmation. If there are many errors click on **Clear IDs** to cancel all CIDs and then go back to the **ID frame** mode where you can try starting **ID all** from a different frame, or manually identify segments at additional frame locations.

Set links again

Once you have good identifications for all frames of interest you should once again **Set (links)** if you wish to save a link file to use with other **.seg** files.

Save links

Use the **Save (links)** button to write the link data to the project link file. You may also use the **File/Save links as...** menu item to save the data to a different file.

Identifying using a link file

This section describes some techniques for using an appropriate link file to identify the segments in a **.seg** file. When a **.seg** file is opened in **Identify** the project link file specified in the Identify parameters is automatically loaded by the program.

Select frames

If desired first select the frame range over which you are interested in getting *.c3d* output by moving the frame cursor to the beginning frame and pressing the **Home** key, and then moving it to the end frame and pressing the **End** key.

Use the ID frame mode

Click on the **ID frame** button to enter the individual frame mode.

First you may wish to adjust the [link scaling](#) if there is a uniform size difference in the separation of markers in the *.seg* file used to define the links and the current *.seg* file. The easiest way to change the scaling is to place the mouse cursor over the **Scale** box and use the mouse wheel to modify the % value. Note that the **Scale** box is only enabled when the **LINKS** mode is turned off.

The scaling parameter temporarily modifies the link length data and is only used for point identifications. If you turn on the **LINKS** mode the scaling is reset back to its original value (100%), and the **Scale** box becomes disabled. The scale value has no effect on the link data saved to the link file, or the link data when the **Set** or **Expand** commands are used because link data are always derived just from the UIDs.

Move the frame cursor to various locations in the file and find a location where all or most of the CIDs are correct. If a frame shows all correct CIDs press the **Enter** key (or click on the **Confirm frame** button) to convert the CIDs (green) to UIDs (yellow). If one or more CIDs are incorrect you need to correct them before confirming the frame.

Once you have a frame with good UIDs you may move to another frame location and repeat the process, or try the **ID all** function.

Correct the identifications

Even the best frames may have CIDs that are incorrect, especially if there are a number of links of similar lengths, the link lengths have large variations throughout the trial, or markers are missing from the view. In such situations you can manually cancel and/or modify assignments before doing the confirmation.

Use ID all

At any time you can try using the **ID all** button. Its first action is to confirm all CIDs in the current frame (same as using the **Confirm frame** button or pressing **Enter**). Hence before using it first make sure that all CIDs in the current frame are correct. It also cancels all other remaining CIDs (those not present in the current view area) before doing the computation. Once the computation is finished you should scan through the file to check the correctness of the CIDs and make corrections and deletions as appropriate. At any point you may use the **Confirm frame** or **Confirm all** buttons to convert CIDs to UIDs.

If the **ID all** results are unsatisfactory, click on **ID frame** again to cancel all CIDs and recalculate the

IDs for the current frame.

From:
<https://has-motion.com/wiki/> - Wiki Documentation Site

Permanent link:
https://has-motion.com/wiki/doku.php?id=other:amass:documentation:identification_using_links&rev=1719432829

Last update: **2024/06/26 20:13**

