MPII 3D Joint Location Ground Truth Annotation Tool Readme

Introduction

This tool implements the 3D joint annotation for 2D human pose datset MPII. Written in C++ & C#. Libraries used: OpenGL & OpenCV.

Main functionalities:

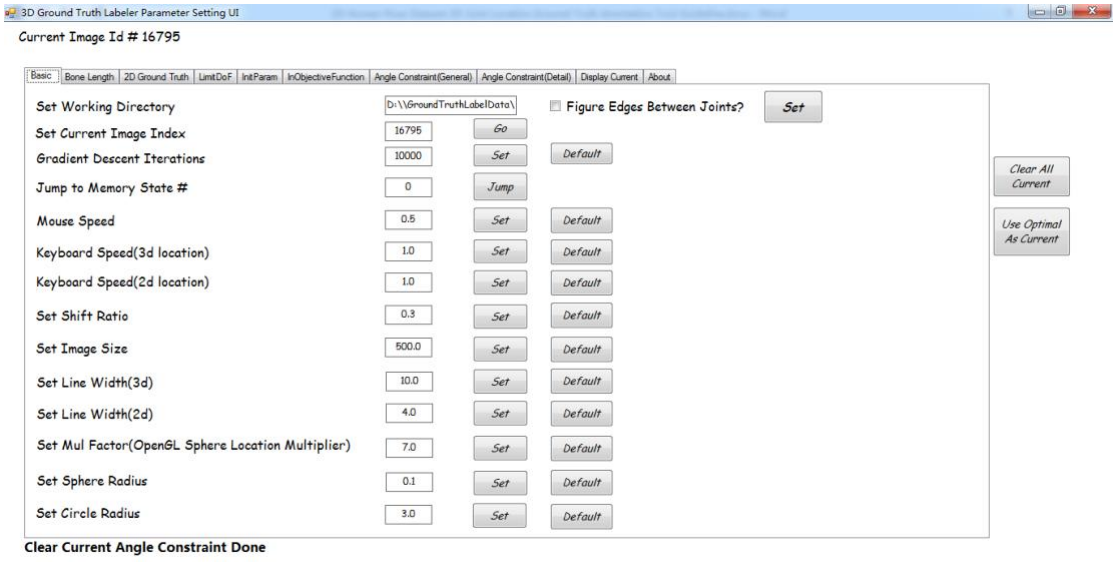
1） Freely translate, rotate and drag 3D coordinate via mouse and keyboard
2） Shift 3D coordinates of keypoints (left, right, up, down) using keyboard and mouse
3） Freely shift 2D keypoints annotation via keyboard and mouse
4） Given an option of (1) joint angle constraint (2) pose-conditioned joint angle limit (3) without any constraint. Optimize a 3D forward kinematics human model such that the 2D projection is minimum to the 2D annotation. The optimized 3D model is visualized by OpenGL. The corresponding 2D projection is shown in OpenCV windows. The visualization is real-time so that the user can observe the changes as the user shifts 2D and 3D keypoints
5） Change settings: bone length of all or current image; initial joint angle parameters (for FK model); constrained DoF (angle params); joint index in the objective function; objective function etc.
6） Auto save and permanently save result per image
7） Navigate to any image and load currently saved 2D and 3D annotations
8） Inspect current optimization result (joint angle parameters, 3D joint locations) in real
9） Set user-friendly preferences for OpenGL and OpenCV
10） Load settings for any backup. Tag supported
11） All saved results are text files with ".txt" suffix

Basically, the C++ OpenGL program provides a visualization interface for displaying 3D human model, where the user can interact with the system.

The C# interface sets certain things up, e.g. directories to save the optimized results etc

Main Frame

1)C#



The bottom status bar shows the current operation, like below

Set Current In Objective Function Id Done

Set Angle Constraint to Current Image Done

Finish Navigating to 5

Finish Displaying Current Optimal

Finish Replacing Current Optimal(result folder)

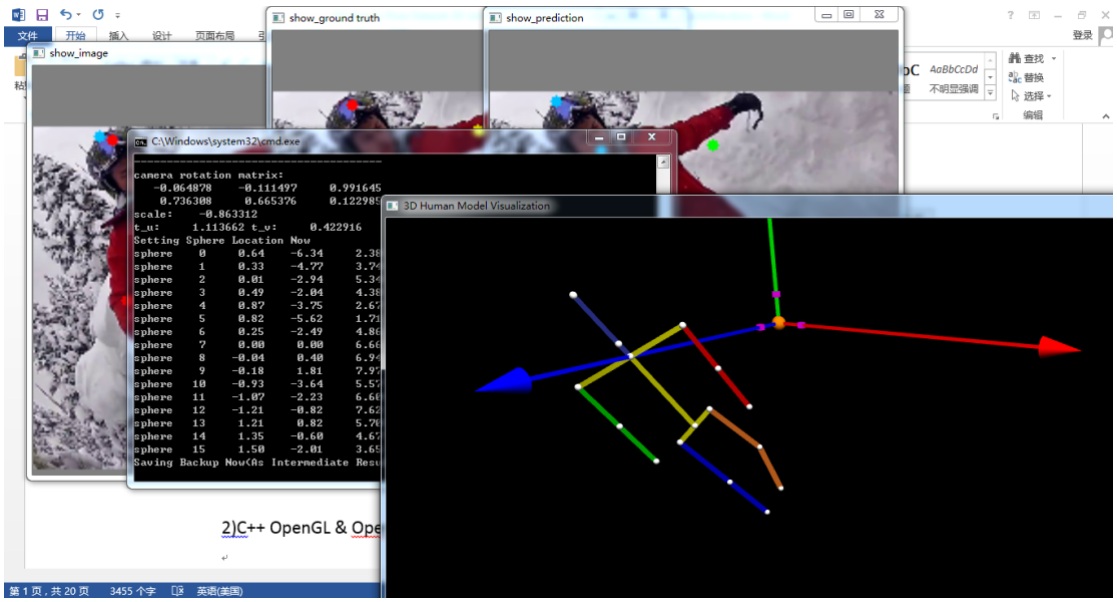
Finish Loading Angle Constraint File

Finish Loading Optimal As Base Parameter Settings

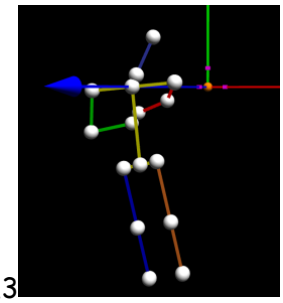
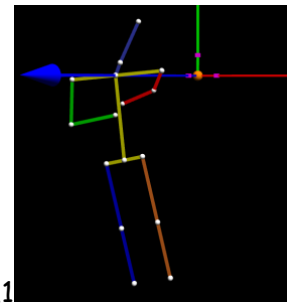
Clear Current Angle Constraint Done

Set This Bone Length to All Images

2)C++ OpenGL & OpenCV

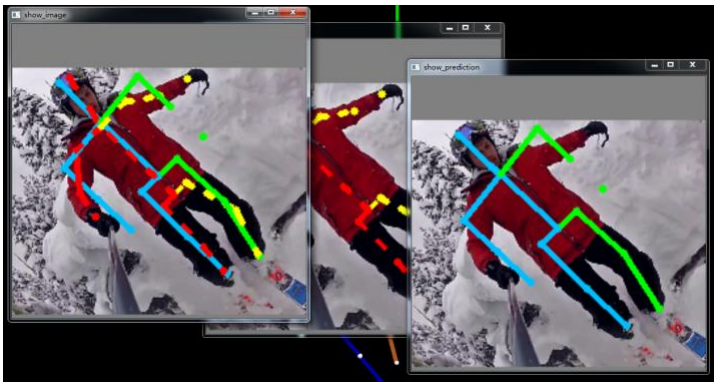


Basic settings

1. Set Working Directory
   Root dir
2. Set Current Image Index
   Click "Go" to reflect the changes in current_imageid.in.
3. Gradient Descent Iterations
   The default value is 10000
4. Jump to Memory State #
   With regard to each image, there are a couple of temporary states in the memory, to which the C++ program can switch. Details follow next.
5. Mouse Speed
   In the OpenGL interface, left click mouse+ drag = rotate view (rotate the coordinate system). Right click + drag = translate the coordinate. Select the keypoint to be changed and then right click + drag the mouse equals updating the depth "z" value of that keypoint. The speed of mouse is set here.
6. Keyboard Speed(3d location)
   Use 0-f (keypoint 0-15) to select the keypoint, ← ↑ → ↓ , PageUp, PageDown to modify the keypoint coordinates as well as shift the coordinate. Speed is set here.
7. Keyboard Speed(2d location)
   Shift speed when updating 2d annotations
8. Shift Ratio
   Press shift to make minor changes to the keypoints
9. Image Size
   Change the OpenCV window size
10. Set Mul Factor
    For visualization convenience, the optimized $\{x_i, y_i, z_i\}$ is multiplied by mul_factor.
11. Set Sphere Radius
    radius=0.3  radius=0.1 
12. Set Circle Radius

radius=10.0  radius=3.0 

13. Figure edges between joints



Basic operations

1. Make sure the file paths in FilePath.in are valid before running ThreeDGroundTruthLabeler.exe

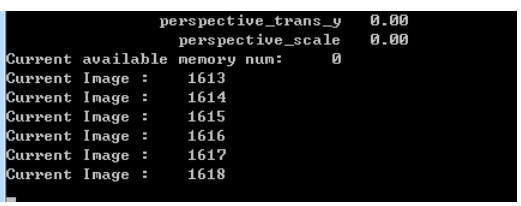
```
D:\\GroundTruthLabelData
F:\\msraintern\\humanpose\\dataset\\mpii\\wqf\\zxydata\\train\\img
F:\\msraintern\\humanpose\\poseprior\\prior\\anglespread.in
F:\\msraintern\\humanpose\\poseprior\\prior\\sepplane.in
F:\\msraintern\\humanpose\\poseprior\\prior\\secondbasis.in
F:\\msraintern\\humanpose\\poseprior\\prior\\bbx.in
F:\\msraintern\\humanpose\\poseprior\\prior\\boundries.in
```

1st line: working root directory

2nd line: image directory (uniform crop size 224x224)

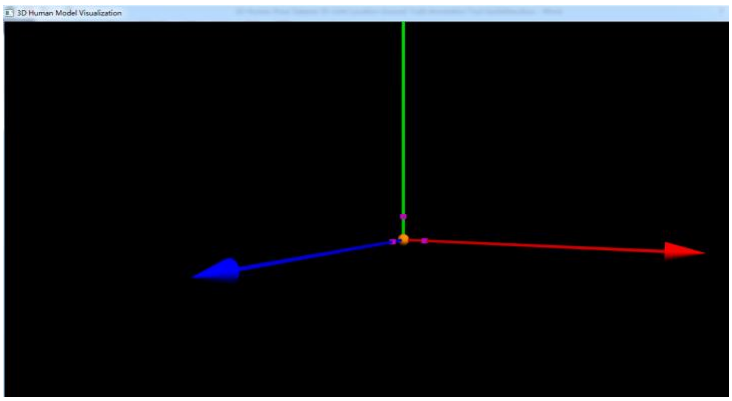
3rd to 7th: matlab file variables from the MatLab code of "pose-conditioned joint angle limits for 3d human pose reconstruction"

2. When the program starts, it will load images from "configuration\current_imageid.in". If the image is already optimized, the previously saved result from "
   result\param_xx.txt、pred_2d_xx.txt、pred_3d_xx.txt" will be loaded and visualized in both 2D and 3D.

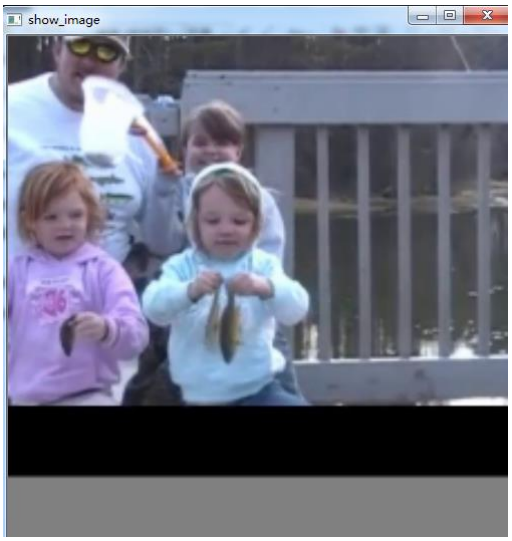
   Terminal will show Image Index 

   ```
   perspective_trans_y   0.00
   perspective_scale   0.00
   Current available memory num:     0
   Current Image :    1613
   Current Image :    1614
   Current Image :    1615
   Current Image :    1616
   Current Image :    1617
   Current Image :    1618
   ```

   Otherwise a blank OpenGL window pops up.



"show_image": the original image containing a human centered



3.

1) Left click and drag the mouse (keep the mouse pressed) to rotate the coordinate

2) Right click and drag the mouse to translate the coordinate. Move cursor to the right in order to let the coordinate shift left.

3) ←: the coordinate moves left, likewise for → ↑ ↓ PageUp: zoom out; PageDown: zoom in（press Shift to make minor changes）

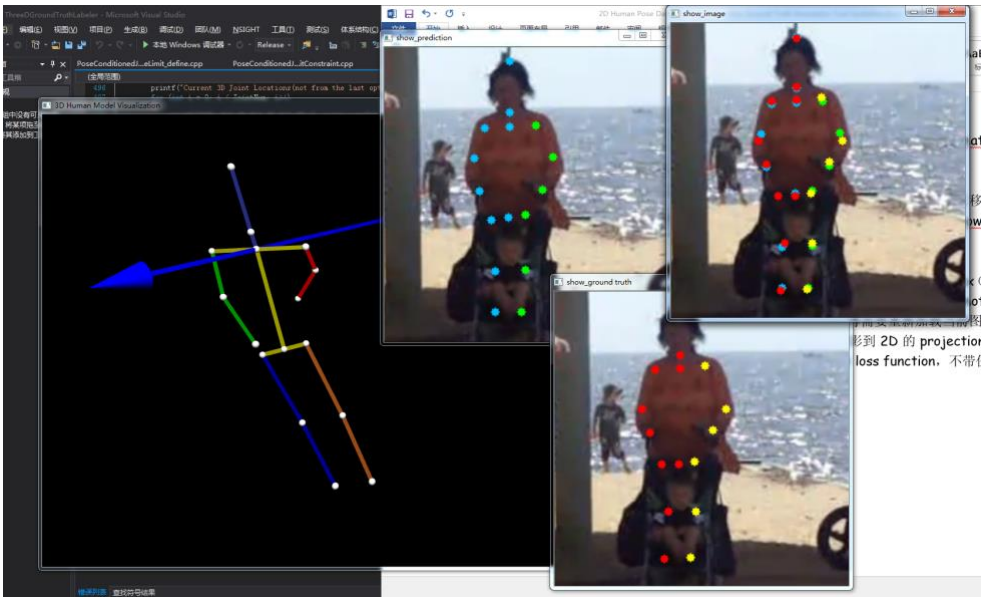
4)Ctrl+PageUp: navigate to the last image Ctrl+PageDown: navigate to the next image

Upon loading a new image

5)l(short for "load"): navigate to the index in "configuration\current_imageid.in" to reflect changes made in the C# interface. E.g. current image index or any other parameters are modified (initial motion parameter, keypoints considered for the objective function, the set of degree of freedom to be constrained etc.)

6)p(short for "projection"): Press "p" first to get an initial Optimize using only reprojection error: the Euclidean distance between projection of optimized 3D (under the FK model optimization) and ground truth 2D. Note

no constraint is enforced at all, let alone say "joint angle limit constraint"



Windows shown: (1) 3D model("3D Human Model Visualization") (2) prediction("show_prediction"、ground truth("show_ground truth") (3) overlay prediction on ground truth ("show_image")

7)P(capital p) optimization with joint angle limit constraint (for each joint angle, there is a lower bound and a upper bound)

8)o(short for "optimize") fit the human forward kinematics model to the current 3D annotation

9)O(capital o) 3D optimization using joint angle limit

10)h(short for "help")

```
Current 3D Joint Locations(not from the last optimization):
                    right_ankle     0.459280    -0.882813     1.090532
                    right_knee      0.233324    -0.692988     1.033843
                     right_hip     -0.026158    -0.461664     0.993121
                      left_hip      0.132869    -0.409353     0.883696
                     left_knee      0.370185    -0.661759     0.933416
                    left_ankle      0.581968    -0.861426     1.006087
                         pelvis     0.053355    -0.435508     0.938408
                         thorax     0.000000     0.000000     0.971357
                     upper_neck    -0.000832     0.068879     0.980171
                       head_top    -0.041588     0.315510     1.004668
                    right_wrist     0.134920    -0.324119     1.134303
                    right_elbow    -0.064913    -0.175910     1.109793
                 right_shoulder    -0.244632    -0.032967     1.010959
                  left_shoulder     0.244632     0.032967     0.931755
                      left_elbow    0.114967    -0.112889     0.775509
                      left_wrist    -0.062904    -0.283932     0.735431
Angles:
                    global_trans_x   0.00
                    global_trans_y   0.00
                    global_trans_z  -0.03
                      global_rot_x   4.29
                      global_rot_y  -9.14
                      global_rot_z   6.98
                  right_knee_rot_x  -4.57
                   right_hip_rot_x -27.73
                   right_hip_rot_y   5.04
                   right_hip_rot_z  23.50
                    left_hip_rot_x -26.61
                    left_hip_rot_y   0.05
                    left_hip_rot_z  17.77
                   left_knee_rot_x  -6.76
                  upper_neck_rot_x   2.85
                  upper_neck_rot_y   0.04
                  upper_neck_rot_z   0.78
                    head_top_rot_x  -1.63
                 right_elbow_rot_x  17.76
              right_shoulder_rot_x -32.68
              right_shoulder_rot_y  10.51
              right_shoulder_rot_z  42.20
               left_shoulder_rot_x  42.08
               left_shoulder_rot_y  21.44
               left_shoulder_rot_z -40.85
                left_elbow_rot_x  -29.73
                      hip_rot_y  -24.85
                      hip_rot_z   7.69
                perspective_rot_x   7.64
                perspective_rot_y  -2.13
                perspective_rot_z -110.51
             perspective_trans_x   1.06
             perspective_trans_y   0.34
               perspective_scale  -0.70
Current available memory num:     0
```

11)m(short for "memory") save the temporary state of the current best optimized model to the memory in C++ for easy jumping.

12)j(short for "jump") jump to the state in configuration\memory_id.in

13)s(short for "save") save the best optimization for the current frame (the final state). The states saved (to files) are as follows
:
result\param_xx.txt (optimized joint angle)
result\pred_2d_xx.txt (2D projection of the optimized model)
result\pred_3d_xx.txt (3D Joint Location of the optimized model)

14) S(capital s) just as backup (not the final result)

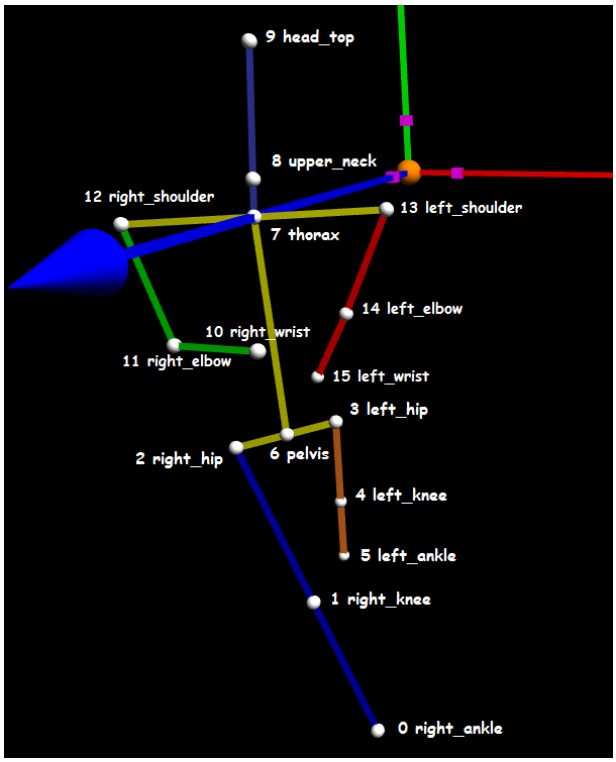
We can have multiple copies for a single frame as backup states. Say the index for the backup of the current frame is ID, saved in "configuration\saveindex.in". Upon releasing the "S" key, files are saved. The files included are:

intermediate\angleconstraint_xx_ID.in --- joint angle limit constraint
intermediate\bonelen_xx_ID.in --- skeletal bone length
intermediate\initparam_xx_ID.in --- initial motion parameter
intermediate\inobj_xx_ID.in --- joint index in energy function
intermediate\islimited_xx_ID.in --- DoF(s) to be constrained
intermediate\interpretation_xx_ID.in --- tags for the backup ID
intermediate\param_xx_ID.txt --- the optimized joint angle
intermediate\pred_2d_xx_ID.txt --- 2D projection of 3D optimization
intermediate\pred_3d_xx_ID.txt --- 3D keypoint locations

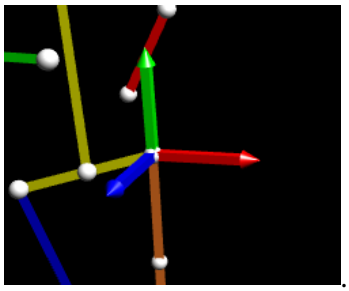
Modified annotations are saved to
intermediate\mod_xx_ID.txt
where xx is the currentimage id

15) C Apply joint angle limit comnstraint from the paper "Pose-conditioned Joint Angle Limits for 3D Human Pose Reconstruction". Output invalid bones and find the closest 3D point clouds to the current 3D keypoints while satisfying the constraint. The "o" event is triggered automatically to fit a 3D model satisfying the forward kinematics constraint.

16) The diagram below shows the index for each joint



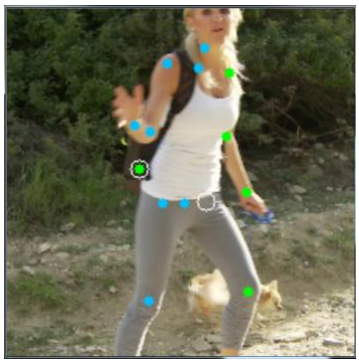
Press the key 0-9、a-f(10-15) to select one specific keypoint and show the local coordinate of that keypoint (red, green, blue)



Use the key ←→↑↓、PageUp、PageDown to adjust x, y, z values of the keypoint（PageU: keypoint is moving towards the user; PageDown: keypoint is moving away from the user）, press and hold the right click key -> drag to the left == keypoint moving towards the user, press and hold the right click -> drag to the right == keypoint moving far away from the user

After manual adjusting, "o" is pressed to fit a FK model to the manual adjustment.

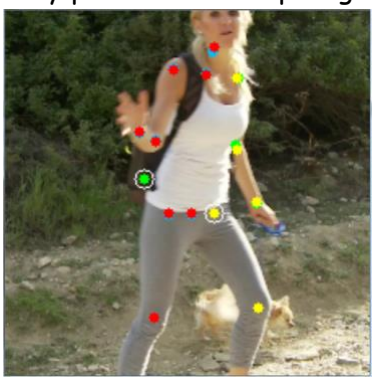
Always press "Esc" to return to the global view (lose focus), and deselect one specific keypoint. At which point the local coordinate system will miraculously disappear.

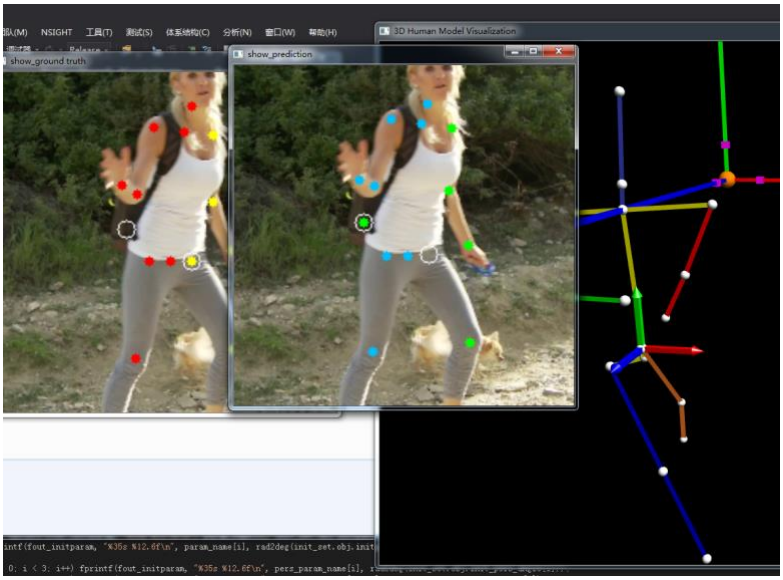
 （The one besieged by a white circle is the joint currently being moved. The green solid fill shows the predicted projection (of the optimization) while the hollow white circle is the ground truth）

This is the window for "show_ground truth":

 （Different from above, the hollow white circle corresponds to the prediction, while the solid white fill is the ground truth）

Overlay prediction on top of ground truth, we get the window "show_image":





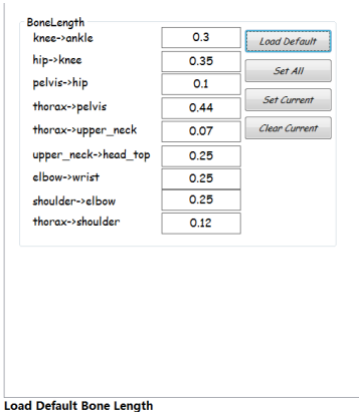
To adjust ground truth

Alt+←/→/↑/↓ (minor changes: Shift+Alt+←→↑↓)

Color semantics:

red right part of ground truth annotation
yellow left part of ground truth annotation
blue right part of prediction
green left part of prediction

Set skeletal bone length



Load Default Bone Length

Click "load default" to load the default bone length

Click "set all" to apply bone length in the text box to all images, then save to the working directiory\BoneLength.in

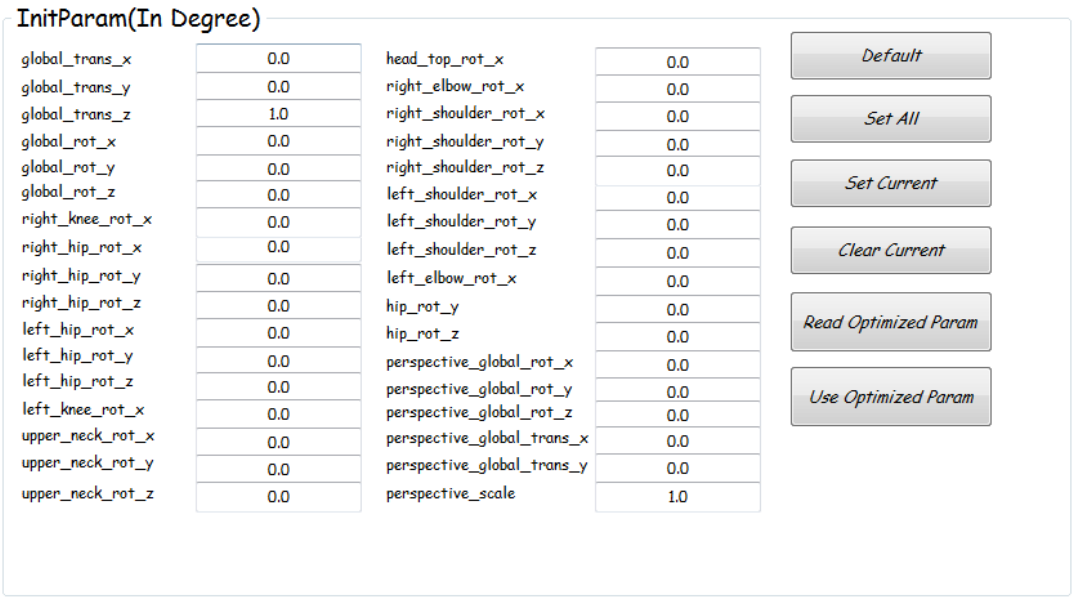
Click "set current" to only apply the length to the current image. Save path will be configuration\bonelen_xx.in (xx is the current image id)

Click "clear current" to delete bonelen_xx.in, and apply universal bone length

2D Ground Truth

To reset the initial joint annotation of the current image before optimization, simply click "Clear Modified Ground Truth". All changes reflected in (gt_2d_modified\mod_xx.txt) will be deleted, whereupon pressing "l" again will load the original ground truth rather than the modified one (the loading priority is set that gt_2d_modified > origin_data)

InitParam (initial motion parameter; joint angle parameters; DoFs)



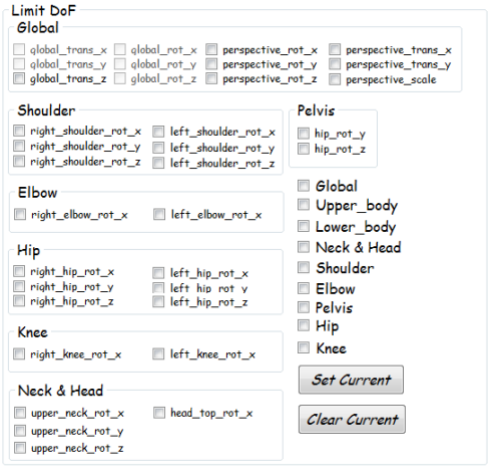
Initial DoFs are:
global_trans_x，global_trans_y，global_trans_z，perspective_global_trans_x，perspective_global_trans_y. Angle degree measure is adopted. Press Default to use default value（globa_trans_z=1.0, perspective_scale=1.0, 0.0 elsewhere）

"Set Current" saves to configuration\initparam_xx.in
"Clear Current" will remove the init param for the current image: configuration\initparam_xx.in
Read Optimized Param load best angle result for the current image from "result\param_xx.in" to the text box.

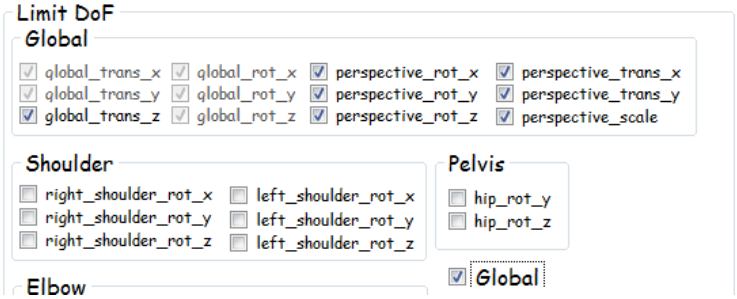
Use Optimized Param will trigger "Read Optimized Param" and save the initial parameter for the current image to configuraiton\initparam_xx.in

DoFs that are constrained

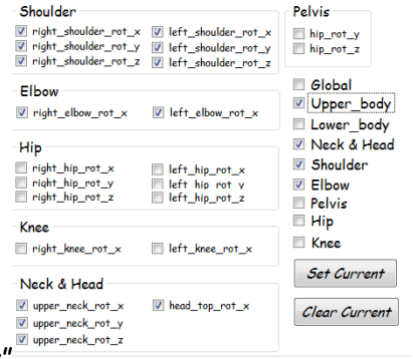
Some DoFs are not learnable (tunable). Instead, initial params will be loaded.



Click "Global"

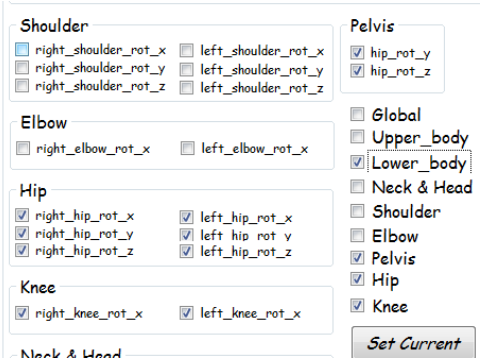


Click "Upper_body"

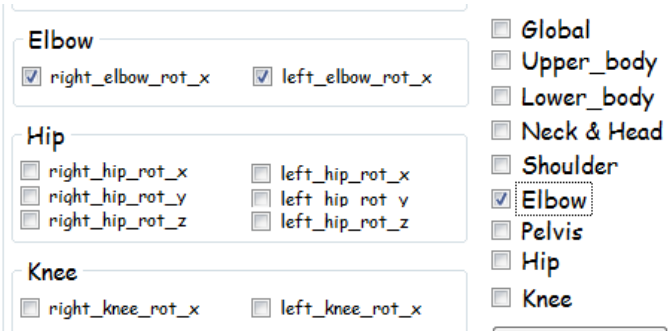


Click "Neck & Head Shoulder Elbow", checkboxs of related DoFs will be checked

Click "Lower_body"



Click "Elbow"



……

Click "Set Current" to reflect constraints to the current image, and save to "configuration\initparam_xx.in"

Click "Clear Current" to clear constraints of the current image. The default loading is working_directory\NoLimitDoF.in(no DoF constraints)

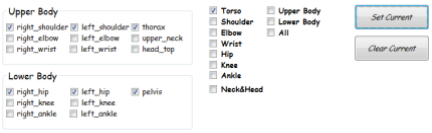
Keypoints in the objective function

To conveniently perform a hierarchical optimization, it is encouraged to select"t orso"

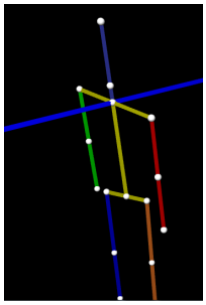
to ease the burden of optimization and first optimize the body. "Set Current" will save current setting to configuration\inobj_xx.in. "Clear Current" will restore the default setting from objective_directory\AllInObj.in(All keypoints considered)

An example of hierarchical optimization

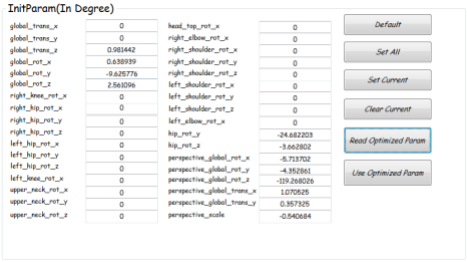
Hierarchical:



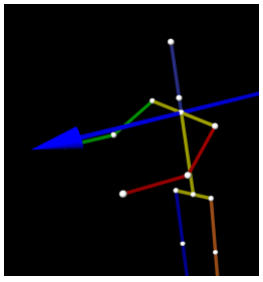
Set “torso” only in InObjectiveFunction.



, press “s” to save this state。Load optimized param in InitParam to the base of the next optimization（pipeline: “Read Optimized Param”→”Use Optimized Param”→”Set Current”）. Next, check “Pelvis”,”Global” in LimitDoF→”Set Current”, check “Elbow”, “Wrist” in InObjectiveFunction→”Set Current”



Press “l” to reload the settings in OpenGL, then press “p” to optimize. At this point, all DoFs concerning pelvis and global are fixed, the only keypoints taken into accout are “elbow” and “wrist”



Angle Constraint

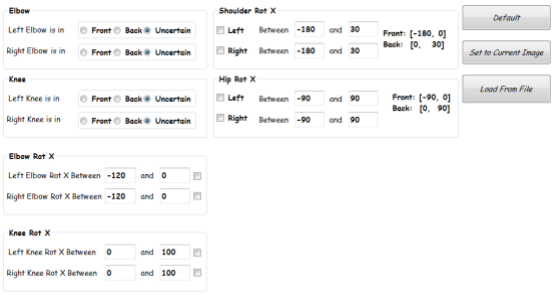
Set lower bound and upper bound for some angles.

	Lower_Bound	Upper_Bound
global_trans_x	-3600	3600
global_trans_y	-3600	3600
global_trans_z	-3600	3600
global_rot_x	-3600	3600
global_rot_y	-3600	3600
global_rot_z	-3600	3600
right_knee_rot_x	0.0	100
right_hip_rot_x	-90	90
right_hip_rot_y	-45	45
right_hip_rot_z	-90	0.0
left_hip_rot_x	-90	90
left_hip_rot_y	-45	45
left_hip_rot_z	0.0	90
left_knee_rot_x	0.0	100
upper_neck_rot_x	-60	90
upper_neck_rot_y	-30	30
upper_neck_rot_z	-30	30

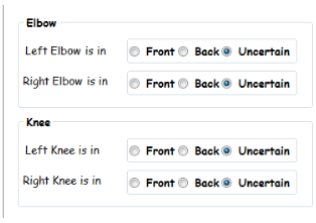
	Lower_Bound	Upper_Bound
head_top_rot_x	0.0	30
right_elbow_rot_x	-120	0.0
right_shoulder_rot_x	-180	30
right_shoulder_rot_y	-90	90
right_shoulder_rot_z	-180	0.0
left_shoulder_rot_x	-180	30
left_shoulder_rot_y	-90	90
left_shoulder_rot_z	0.0	180
left_elbow_rot_x	-120	0.0
hip_rot_y	-90	90
hip_rot_z	-30	30
perspective_global_rot_x	-90	90
perspective_global_rot_y	-90	90
perspective_global_rot_z	-90	90
perspective_global_trans_x	-10	10
perspective_global_trans_y	-10	10
perspective_scale	-100	100

Default
Set All
Set Current
Clear Current
Load From File

Click "Default" to load from working_directory\angle_constraint.in.
"Set All" saves to working_directory \angle_constraint.in
"Set Current" saves to configuration\angleconstraint_xx.in
"Clear Current" deletes configuration\angleconstraint_xx.in
"Load From File" imports angle constraint of the current image to the text box



It's easily observed from the image that Elbow, Knee are in the front/at the back of body. Ambiguity from projecting 3D to 2D arises, in large part from the nebulous front/back orientation of elbow and knee.

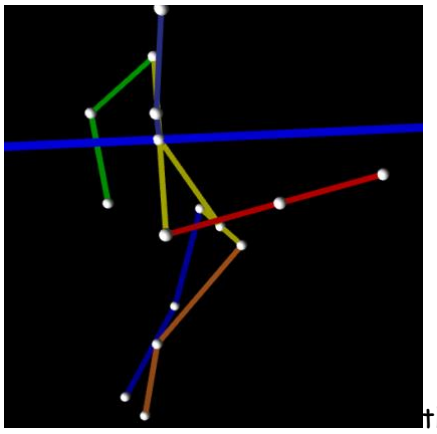
One way to get around this is by assigning front/back relationship between nearby keypoints 

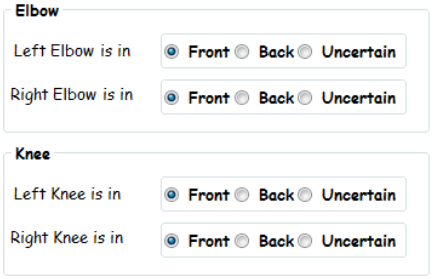
to restrain joint angles. Another complicated method is to concretely set the angles, e.g. the rotation angle of shoulder along axis X i.e. shoulder_rot_x, [-180,0]: elbow in front of the body. [0,30]: elbow at the back of the body. hip_rot_x[-90,0]: knee in the front, [0,90]: knee at the back

Other than that, set Elbow Rot X and Knee Rot X can effectively change the movable range of Wrist and Ankle.

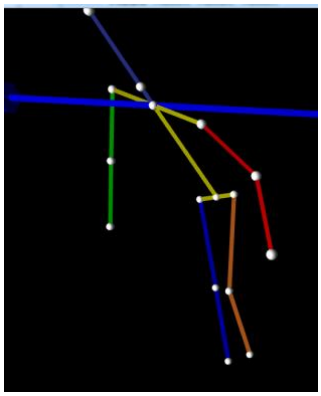
Angle Constraint 1(id: 1639)

 , press “p” (only 2d projection loss)

 the left arm is facing back, which is contradictory to the image evidence. Plus, the body is twisting dramatically.



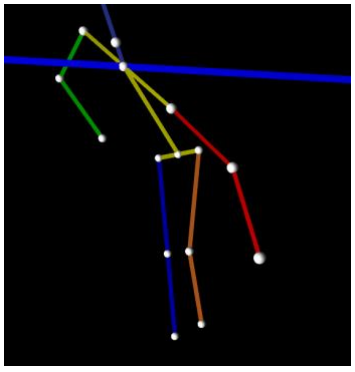
By setting elbow and knee forward, along with”P” : the optimization with joint angle limit constraint, the result follows:

 2D prediction  .

At this time we find a mismatch between pred and gt for the joint right_elbow and right_wrist. We see right_elbow is bending significantly. The following figure sets

the angle range to [-120, -70], which is more reasonable.

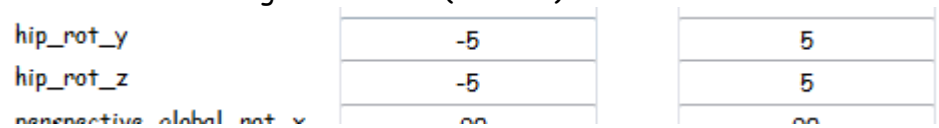
（Check means apply this constraint）



. However, the shoulder and hip still are not acceptable.

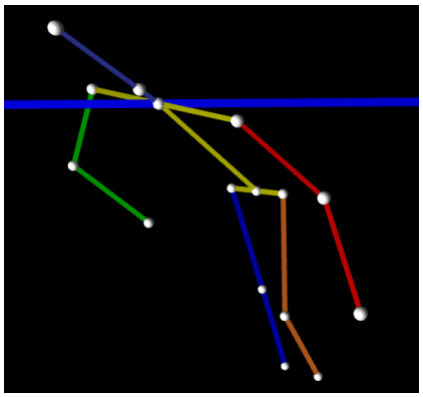
Then we set hip_rot_y, hip_rot_z to [-10,10].

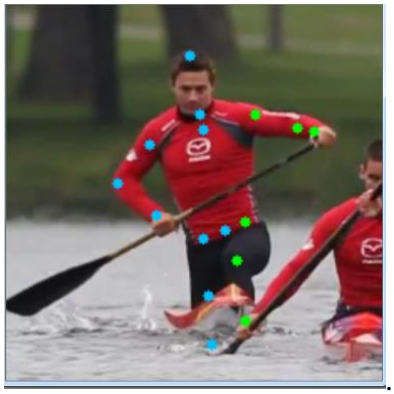
Load From File in Angle Constraint(General)



→ "Set Current"

Reload in C++ by "l" -> "P" to re-optimize



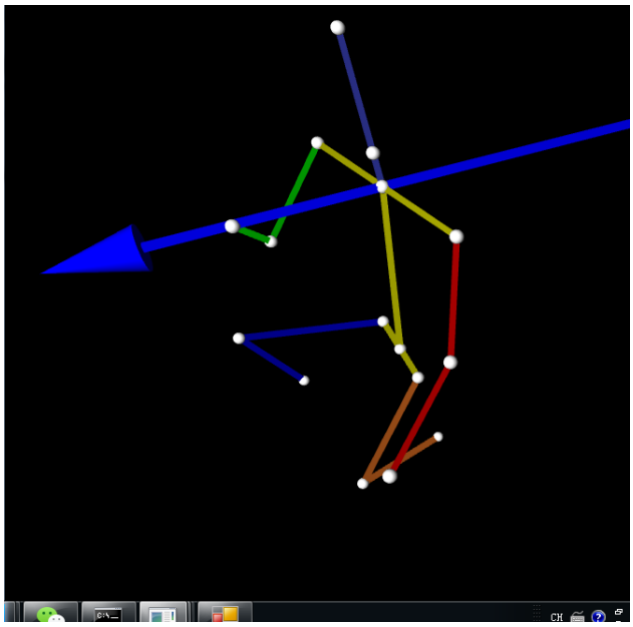


. Now we are done.

Angle Constraint 2（1670）

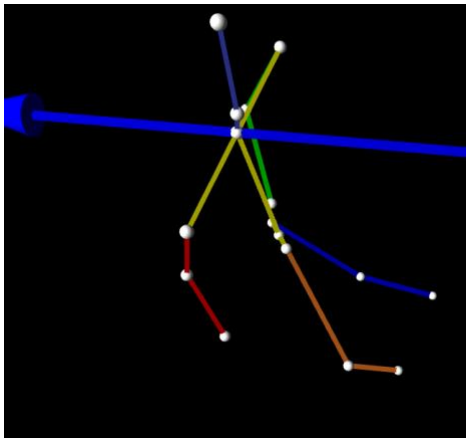
Set right hip rot x to be at least -60 degree

Set hip_rot_y in between [-5, 5] to prevent from over-bending

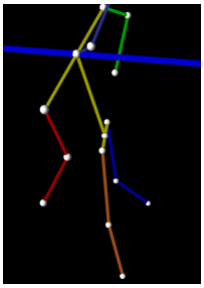


Angle Constraint 3(16712)

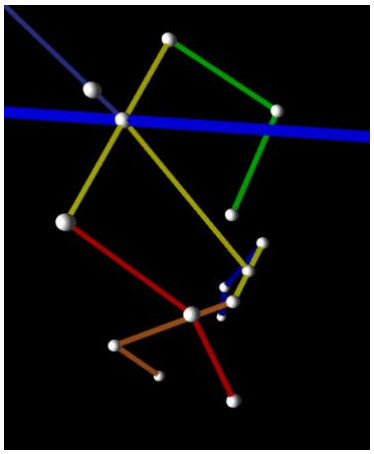
optimize w/o constraint



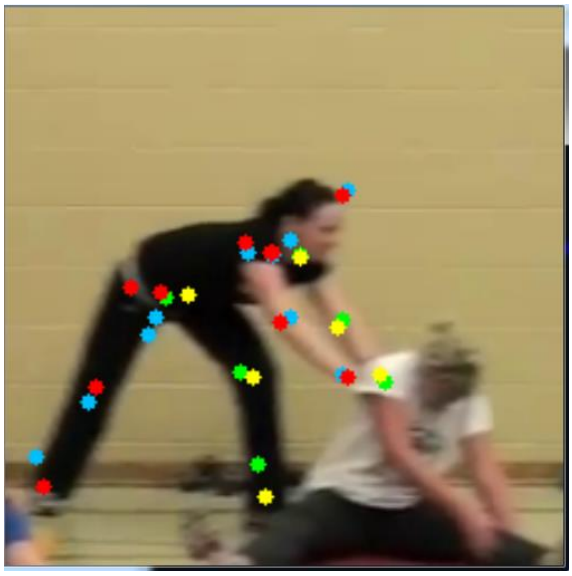
Optimize with constraint



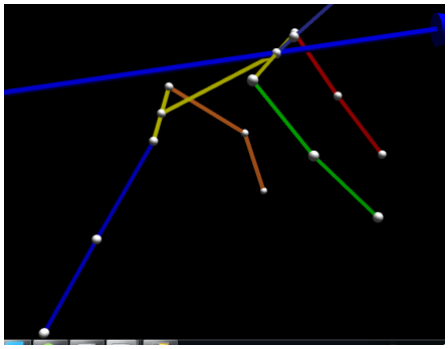
Result: The left leg is incorrect. We now constrain left hip rot xto fall in the range [-90,-20]

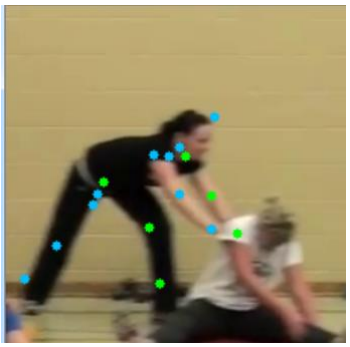


Angle Constraint 4(16784)

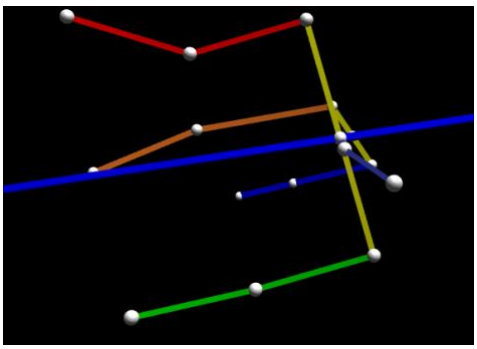


Enforce constraints



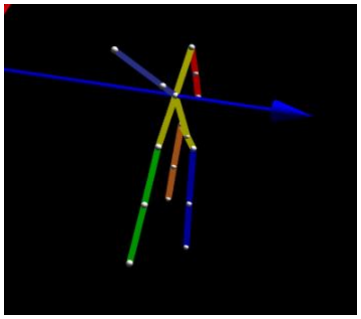


W/o constraints:

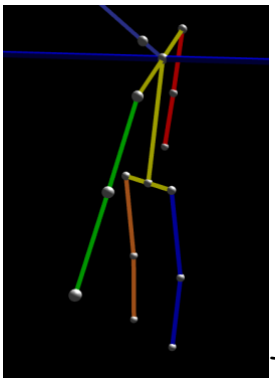


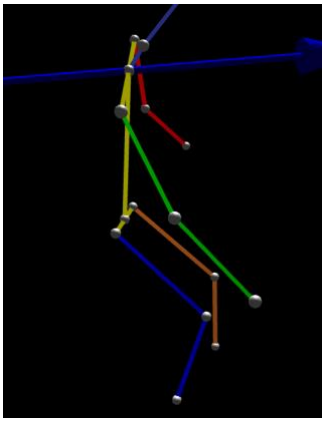
Angle Constraint 5(16791)

Naive Constraint:



Considering the bending of left knee and right knee is

considerably large, we set it to be rougly [30, 50]  Then we

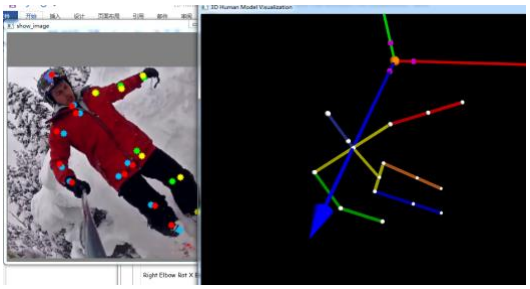
adjust hip_rot_y and hip_rot_z to be in [-10, 10] 

 The projection almost match.

Pose-Conditioned Joint Angle Limit Constraint

The original MatLab code is rewritten in C++.

When all bones are legal>

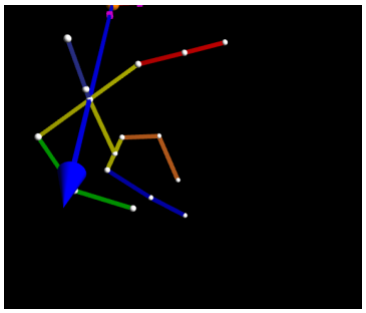


press "C",

The prompt says

```
All bones are valid!
```

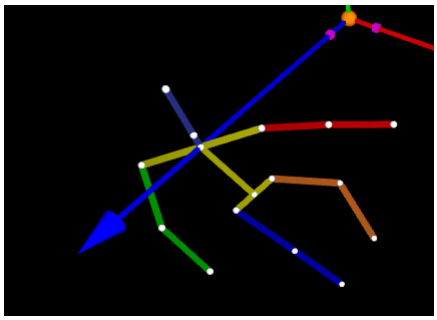
Otherwise for example manually adjust left_knee as follows



then press "C"

The program will replace the current joint with the closest joint satisfying the constraint in the paper "pose-conditioned joint angle limits for 3d human pose reconstruction". A prompt will say

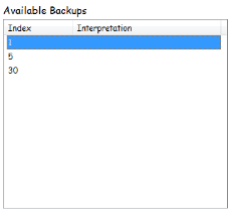
```
right_elbow       --->   right_wrist      is invalid!
left_knee         --->   left_ankle       is invalid!
Replacing it with the closest valid points under the pose-conditioned joint angl
e limit constraint now!Current Iteration       1 last error:   0.16741991 error
```



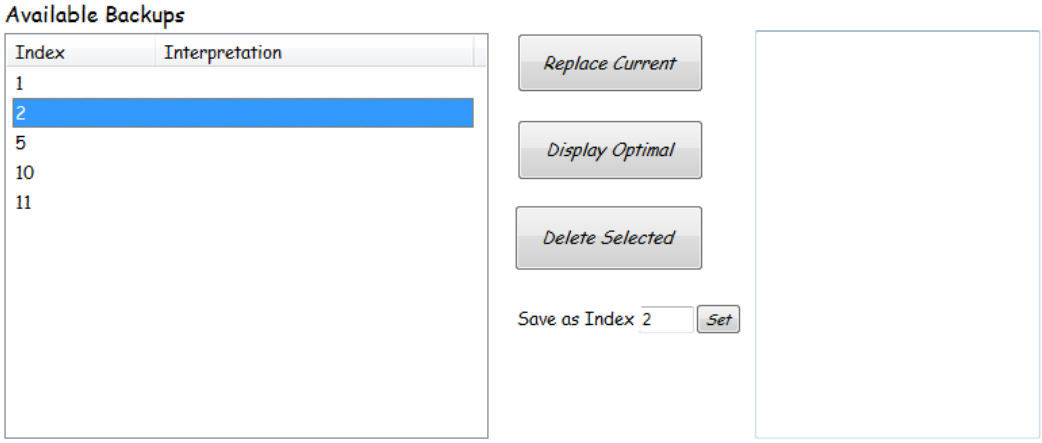
Display settings and backups for the current image

For each image, we can save multiple copies (by pressing "S") to represent results under different settings.

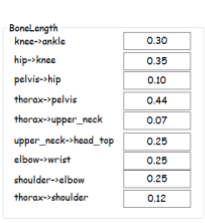
The available backup index is:

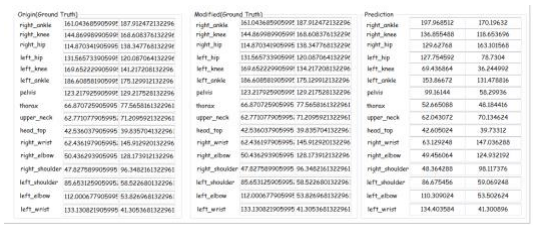


Click index in the listbox will navigate the that specific backup (checkpoint). And show the corresponding bone length etc.

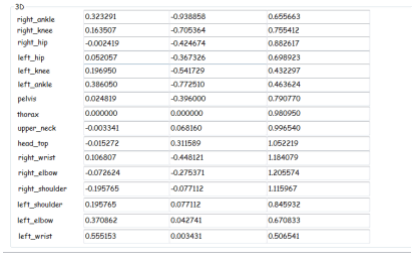


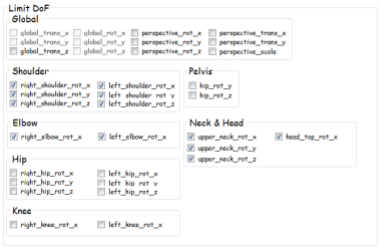
Tags (notes) will also be shown for easy future reference. Modifications in the textbox will sync to files and the listbox (column Interpretation) in real time.

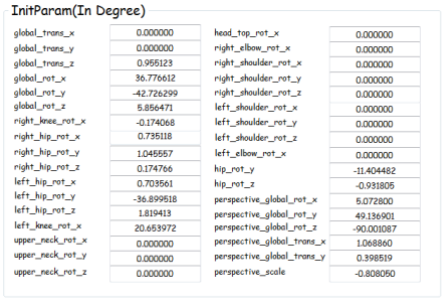


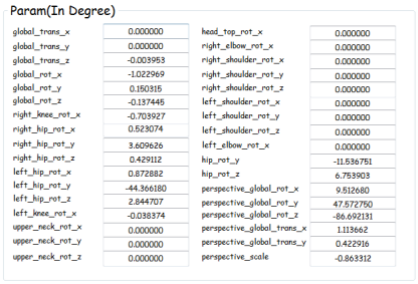


（coordinates within 224×224）







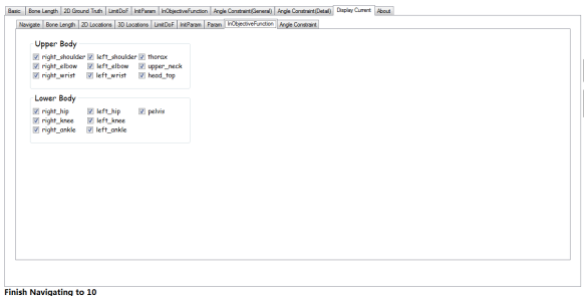


InObjectiveFunction

AngleConstraint

	Lower_Bound	Upper_Bound
global_trans_x	-3600.000000	3600.000000
global_trans_y	-3600.000000	3600.000000
global_trans_z	-3600.000000	3600.000000
global_rot_x	-3600.000000	3600.000000
global_rot_y	-3600.000000	3600.000000
global_rot_z	-3600.000000	3600.000000
right_knee_rot_x	0.000000	100.000000
right_hip_rot_x	-90.000000	0.000000
right_hip_rot_y	-45.000000	45.000000
right_hip_rot_z	-90.000000	0.000000
left_hip_rot_x	-90.000000	0.000000
left_hip_rot_y	-45.000000	45.000000
left_hip_rot_z	0.000000	90.000000
left_knee_rot_x	0.000000	100.000000
upper_neck_rot_x	-60.000000	90.000000
upper_neck_rot_y	-30.000000	30.000000
upper_neck_rot_z	-30.000000	30.000000

	Lower_Bound	Upper_Bound
head_top_rot_x	0.000000	30.000000
right_elbow_rot_x	-120.000000	0.000000
right_shoulder_rot_x	-180.000000	0.000000
right_shoulder_rot_y	-90.000000	90.000000
right_shoulder_rot_z	-180.000000	0.000000
left_shoulder_rot_x	-180.000000	0.000000
left_shoulder_rot_y	-90.000000	90.000000
left_shoulder_rot_z	0.000000	180.000000
left_elbow_rot_x	-120.000000	0.000000
hip_rot_y	-90.000000	90.000000
hip_rot_z	-30.000000	30.000000
perspective_global_rot_x	-90.000000	90.000000
perspective_global_rot_y	-90.000000	90.000000
perspective_global_rot_z	-90.000000	90.000000
perspective_global_trans_x	-10.000000	10.000000
perspective_global_trans_y	-10.000000	10.000000
perspective_scale	-100.000000	100.000000



Click "Delete Selected" to delete the current selected backup.
Click "Display Optimal" to read optimal param, 2D, 3D from working_directory\result\
Click "Replace Current" to save selected backup (state) under the "result\"

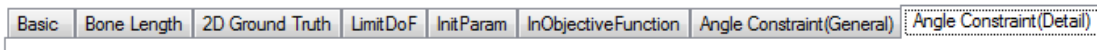
folder

Clear All Current

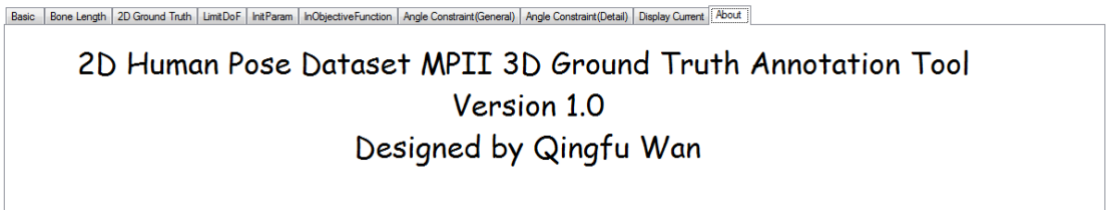
Restore to factory settings.

Use Optimal As Current

To use the settings (bone length; initparam) related to the optimal result ("result\") of the current image, click "Use Optimal As Current". The current settings will automatically be loaded to the textboxs under different tabs.



Version



Code structure

1. *C++*
   1) include

Name	Line of Code	Function
basic.h	110	Declarations of basic variables and functions
HumanModel_define.h	40	Declarations of the human model
HumanModel.h	80	Declarations of the forward kinematics layer(human model layer)
InitConfiguration.h	28	Declarations of the initial setting

		module(bone length, initparam etc.)
WeakPerspective.h	17	Declarations of the weak perspective projection layer
JointLocationLoss_2d.h	15	Declarations of the 2d joint location loss layer
JointLocationLoss_3d.h	15	Declarations of the 3d joint location loss layer
AngleConstraintLoss.h	17	Declarations of the angle constraint loss layer
ProjectionParamLoss.h	17	Declarations of the projection parameter loss layer
GradientDescent_2d.h	36	Declarations of the gradient descent module(2d joint location loss)
GradientDescent_3d.h	32	Declarations of the gradient descent module(3d joint location loss)
DisplayJointOnImage.h	120	Declarations of the module of displaying joint on image
ModelOptimization.h	20	Declarations of the human model optimization module(2d & 3d optimization)
PoseConditionedJointAngleLimit_define.h	126	Declarations of some arrays and functions used in the PCJAL paper
PoseConditionedJointAngleLimitConstraint.h	14	Declarations of the PCJAL constraint functions
global2local.h	4	Declarations of the

		conversion from global coordinate to local coordinate
local2global.h	4	Declarations of the conversion from local coordinate to global coordinate
All	680	

2) source

Name	Line of Code	Function
basic.cpp	311	Basic functions
HumanModel.cpp	204	The forward kinematics layer(human model layer)
InitConfiguration.cpp	71	The initial setting module(bone length, initparam etc.)
WeakPerspective.cpp	79	The weak perspective projection layer
JointLocationLoss_2d.cpp	38	The 2d joint location loss layer
JointLocationLoss_3d.cpp	38	The 3d joint location loss layer
AngleConstraintLoss.cpp	26	The angle constraint loss layer
ProjectionParamLoss.cpp	42	The projection parameter loss layer
GradientDescent_2d.cpp	201	The gradient descent module(2d joint location loss)
GradientDescent_3d.cpp	178	The gradient descent module(3d joint location loss)
DisplayJointOnImage.cpp	120	The module of displaying joint on image
ModelOptimization.h	101	The human model optimization

		module(2d & 3d optimization)
PoseConditionedJointAngleLimit_define.cpp	98	Some arrays and functions used in the PCJAL paper(e.g. get normal spherical to Cartesian)
PoseConditionedJointAngleLimitConstraint.cpp	337	The PCJAL constraint functions
global2local.cpp	44	The conversion from global coordinate to local coordinate
local2global.cpp	43	The conversion from local coordinate to global coordinate
ogl.cpp	981	OpenGL functions(keyboard & mouse functions & set sphere locations …)
ThreeDGroundTruthLabeler.cpp	732	Main Function
All	3644	

2. C#

Name	Line of Code	Function
frmmain.cs	2092	Main file

3. All

C++ 1)include 680
2)src 3644
C# 2092
Total 6416

Folder description

1. Working root directory
   folder configuration、gt_2d_modified、intermediate、origin_data、result
   file AllInObj.in inobjectivefunction
   BoneLength.in: default bone length

InitParam.in: initial parameter

NoLimitDoF.in: initial constrained DoF index

angle_constraint.in: default angle constraint

2. configuration

current_imageid.in

descent_iterations.in

memory_id.in the memory id to jump to

mouse_speed.in

keyboard_speed_3d.in the speed to adjust 3d using keyboard 一个数字，表示键盘调整关节 3d 坐标速度

keyboard_speed_2d.in the keyboard speed to adjust 2d

shift_ratio.in the ratio of shift (minor changes) to normal

tshow_image_size.in

line_width_3d.in line (edge) width connecting 3d spheres

line_width_2d.in line width connecting 2d keypoints

mul_factor.in

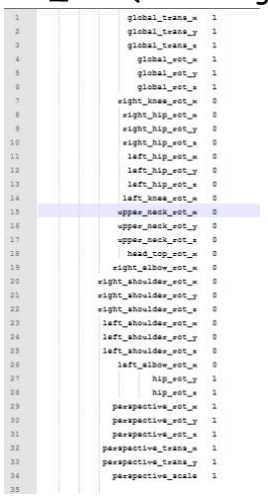
sphere_radius.in

circle_radius.in

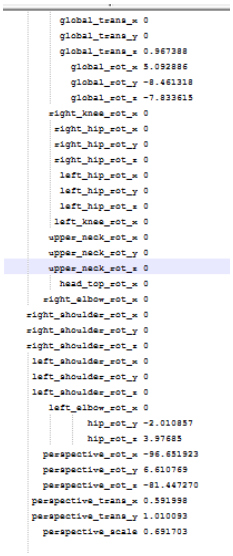
fig_edges.in whether to connect keypoints by edges (lines)

saveindex.in the index to the next memory state backup

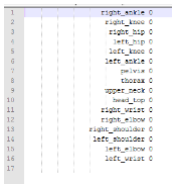
islimited_xx.in(xx is imageid) the Boolean values of each DoF (constrain or not)



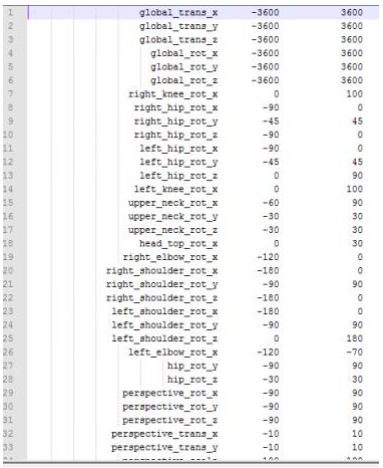
initparam_xx.in



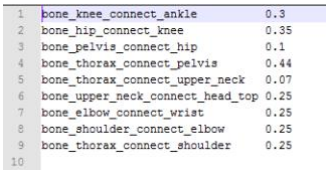
inobj_xx.in



angleconstraint_xx.in

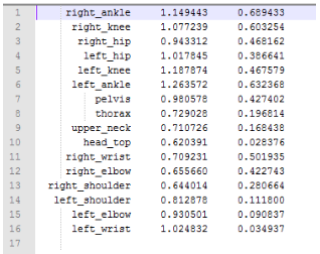


bonelen_xx.in



3. gt_2d_modified

mod_xx.txt the modified 2d ground truth annotation



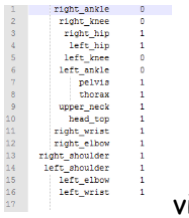
4. intermediate
   angleconstraint_xx_ID.in
   bonelen_xx_ID.in
   initparam_xx_ID.in
   inobj_xx_ID.in
   islimited_xx_ID.in
   interpretation_xx_ID.in notes on the backup ID
   param_xx_ID.txt the optimized joint angle 优化的角度
   pred_2d_xx_ID.txt the 2D projection of optimized 3D
   pred_3d_xx_ID.txt the optimized 3D
5. origin_data

```
    image_id   635.00
   person_id     0.00
   crop_size   813.24
      bbx_x1   604.38
      bbx_y1   156.38
      bbx_x2  1417.62
      bbx_y2   969.62
     head_x1   872.00
     head_y1   311.00
     head_x2   983.00
     head_y2   439.00
 image_width  1920.00
image_height  1080.00

```

config_xx.txt

```
      right_ankle     2.039879     1.283662
       right_knee     2.109782     1.042181
        right_hip     2.188157     0.828238
         left_hip     2.255941     0.824001
        left_knee     2.323725     1.052773
       left_ankle     2.399982     1.292135
            pelvis     2.222049     0.826119
            thorax     2.222049     0.586757
       upper_neck     2.212578     0.550833
         head_top     2.170091     0.389672
     right_wrist     2.243232     0.857893
     right_elbow     2.209340     0.726561
  right_shoulder     2.198748     0.588875
   left_shoulder     2.243232     0.582520
      left_elbow     2.175447     0.732916
      left_wrist     2.073771     0.834592

```

gt_2d_xx.txt



vis_xx.txt visibility of each keypoint

6. result
   param_xx.txt optimizesd joint angle
   pred_2d_xx.txt projection (2D) of optimized 3D coordinate
   pred_3d_xx.txt optimized 3D coordinates
7. images
   Human images of size 224x224, cropped from the original image. (Human-centered)