Progetto Struct - Omar Moh'd

Introduzione

Struct è una sezione della libreria Julia "LAR.jl". Quest'ultima viene utilizzata per eseguire calcoli geometrici su complessi cellulari espressi attraverso la Rappresentazione Algebrica Lineare(LAR).

L'obiettivo di Struct è quello di rappresentare oggetti complessi e descriverli nel loro rispettivo sistema di coordinate, il che risulta particolarmente comodo per specificarne i vertici.

Gli oggetti complessi sono rappresentati mediante modelli chiamati proprio "Struct", trattati e intesi, anche nel codice, come grafi orientati aciclici. Queste sono strutture gerarchiche formate da vari componenti che vivono in diversi sistemi di riferimento.

Lista e spiegazione delle funzioni pre-esistenti

<function t(args...)>

Genera e ritorna una matrice "mat" di trasformazione affine(nello specifico di traslazione) in coordinate omogenee. Questa matrice ha "d + 1" righe e "d + 1" colonne, dove "d" è il numero di parametri nell'array "args" passato come argomento.

<function s(args...)>

Come la funzione precedente, genera e ritorna una matrice "mat" di trasformazione affine(questa volta di ridimensionamento) in coordinate omogenee. Questa matrice ha "d + 1" righe e "d + 1" colonne, dove "d" è il numero di parametri nell'array "args" passato come argomento.

<function r(args...)>

Come le funzioni precedenti, genera e ritorna una matrice "mat" di trasformazione affine(questa volta di rotazione) in coordinate omogenee. Questa matrice ha dimensione uguale a 3 per la rotazione 2D, o uguale a 4 per la rotazione 3D.

L'array "args", o contiene un singolo parametro ovvero l'angolo in radianti, o un vettore con tre elementi, la cui "norma" è l 'angolo di rotazione in 3D e il cui "valore normalizzato" fornisce la direzione dell'asse di rotazione in 3D.

<function removeDups(CW::Cells)::Cells>

Rimuove le celle duplicate dall'oggetto "Cells".

Mette poi "Cells" in forma canonica, ovvero vengono ordinati gli indici di vertici in ogni singolo elemento dell'array "Cells".

<function Struct(...)>

Funzione che applicata ad un array di oggetti crea un "contenitore" di oggetti geometrici con i seguenti attributi: <body, box, name, dim, category>. Ogni valore è definito in coordinate locali e può essere trasformato da tensori di trasformazione affini. Il valore restituito è di tipo "Struct" ed il suo sistema di coordinate è quello associato al primo oggetto degli argomenti della funzione. Inoltre, il valore geometrico risultante è spesso associato a un nome di variabile. La generazione dei contenitori può continuare gerarchicamente applicando opportunamente "Struct".

Avendo definito la struttura come "mutable" è possibile grazie all'utilizzo di apposite funzioni, discusse a seguire, modificare gli attributi anche dopo la creazione/costruzione della struttura.

<function name(self::Struct)>

Restituisce il nome della struttura di tipo "AbstractString".

<function category(self::Struct)>

Restituisce la categoria della struttura di tipo "AbstractString".

<function len(self::Struct)>

Restituisce la lunghezza del corpo(attributo "body" di tipo "Array") della struttura.

<function getItem(self::Struct,i::Int)>

Restituisce l'oggetto con indice "i" nel corpo della struttura.

<function setItem(self::Struct,i,value)>

Imposta il valore dell'oggetto con indice "i" nel corpo della struttura a "value".

<function pprint(self::Struct)>

Stampa il nome della struttura.

<function setName(self::Struct,name)>

Imposta il nome della struttura a "name".

<function clone(self::Struct,i=0)>

Restituisce l'oggetto "newObj" che contiene una copia della struttura.

<function set_category(self::Struct,category)>

Imposta la categoria della struttura a “category”.

<function struct2lar(structure)>

Restituisce una tupla(vertici,celle,spigoli) che consiste nella rappresentazione algebrica lineare della struttura in ingresso. Sostanzialmente trasforma la struttura fondamentalmente “appiattendola” in un’unica struttura dati di tipo LAR. //Trasforma la gerarchia in un solo modello LAR.

<function embedTraversal(cloned::Struct,obj::Struct,n::Int,suffix::String)>

Restituisce l’oggetto cloned di tipo “struct” che consiste in una struttura avente lo stesso body della struttura “obj” passata come parametro. Sono visitati uno per uno gli elementi del body di obj e siano essi matrici, tuple, array o a loro volta strutture, sono copiati all’interno di variabili locali e inserite nella struttura “cloned”.

<function embedStruct(n::Int)>

Definisce la funzione “embedStruct0”.

<function embedStruct0(self::Struct,suffix::String=“New”)>

Restituisce una struttura che è la copia della struttura “self” passata come parametro, dove però la dimensione è aumentata del fattore “n” e al nome è posto il suffisso “new”. Per copiare il body è chiamata “embedTraversal”.

<function box(model)>

I box sono associati ai nodi, ovvero alle strutture. Sia il modello in ingresso una struttura, una tupla o un’array, restituisce [theMin,theMax] che rappresenta il volume di contenimento, ovvero il minimo parallelepipedo parallelo agli assi che li contiene. Estrae quindi il solo volume di vista, caratterizzato dai parametri del box stesso(angolo di apertura, direzione asse, . . .).

<function apply(affineMatrix::Array{Float64,2}, larmodel::Union{LAR,LARmodel})>

Modifica l’oggetto “larmodel” passato come parametro e lo restituisce di tipo tupla. Nello specifico è una coppia (Geometria, Topologia) in cui la prima è memorizzata come “punti”, mentre la seconda come “array di celle”.

I suoi vertici vengono modificati secondo la matrice di affinità “affineMatrix” moltiplicata per il parametro “W” calcolato all’interno della funzione.

<function checkStruct(lst)>

Calcola e ritorna la dimensione “dim” dell’oggetto “lst”.

<function traversal(CTM::Matrix, stack, obj, scene=[])>

Trasformazione della gerarchia in un unico sistema di coordinate, ovvero quello della radice(chiamate coordinate mondo). Viene visitato in profondità il grafo(obj.body). Durante la visita viene conservata una matrice CTM(current transformation matrix) che viene inizializzata quando si parte dalla radice. Quando nella visita si incontra una matrice si aggiorna la CTM moltiplicandola per essa(vengono moltiplicati i vertici).

Man a mano che il grafo viene visitato si può pensare che la matrice resti associata all'ultimo arco e quando si entra in un nodo per visitarne il sottoalbero di cui è radice, bisogna salvare lo stato corrente di questa matrice facendo un "push" della CTM su uno stack. Quando invece si esce dal sottoalbero, si fa un "pop" dello stack.

<function evalStruct(self::Struct)>