

Elastic Graph Data Structure

Xiaoyang Guo

January 31, 2020

```
>> EG{1}

ans =

struct with fields:

    Abeta: [2×101×5×5 double]
    beta: [2×101×4 double]
    nodeXY: [2×5 double]
    A: [5×5 double]
    G: [5×8 double]
    H: [5×8 double]
```

Fields in EG structure:

- **Abeta**: Generalized adjacency matrix whose entry is a curve. `size(Abeta,3)=size(Abeta,4)` = number of nodes. `Abeta(:, :, i, j)` is an edge curve between node `i` and `j`. All edges are original curves rather than SRVF
- **beta**: Flattened version of upper triangular of **Abeta**. `size(beta,3)` is the number of edges. Refer to `../src/adj2beta.m`
- **nodeXY**: Matrix of node coordinate. `nodeXY(:, i)` is the coordinate of node `i`
- **A**: Binary version of adjacency matrix
- **G**: Graph incidence matrix. `size(G,1)` is the number of nodes. `size(G,2)` is 2 times of number of edges. Row index is the nodes while column index is the edges. Refer to [1] for more details.
- **H**: Graph incidence matrix. `size(H,1)` is the number of nodes. `size(H,2)` is 2 times of number of edges. Row index is the nodes while column index is the edges. Refer to [1] for more details.

Note **G** and **H** are only used for graph matching algorithm. Specifically, we are using the algorithm for directed graph matching from [1]. The edge between `i` and `j` is treated as 2 virtual directed edges: one from `i` to `j` and one from `j` to `i`. By doing so, we are able to optimize over a flipped reparametrization. Accordingly, `Abeta(:, :, i, j) = flipplr(Abeta(:, :, j, i))`. **G** and **H**'s columns are doubled.

References

- [1] Feng Zhou and Fernando De la Torre. Factorized graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1774–1789, 2015.