A Tool for N-way Analysis of Programming Exercises

C. Andujar   M. Comino   M. Fairen   A. Vinacua

ViRVIG, Universitat Politècnica de Catalunya, Barcelona

Overview

We present a tool to support the grading of programming exercises.

Key ideas:

- Compute syntactic, semantic and functional similarities.
- Embed submissions in 2D, mapping similar code to nearby locations.

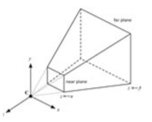
Users can:

- Identify clusters (similar submissions).
- Inspect individual submissions.
- Do pair-wise comparisons.
- Do abridged N-way comparisons.
- Sort and grade submissions by similarity.

Exercise example

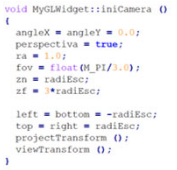
Exercise 1 [iniCamera] *Write a C++ method that initializes a camera whose frustum encloses the scene while maximizing the viewport occupancy.*

```
void MyGLWidget::iniCamera();
```



Submission examples

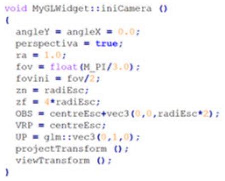
Student i



```
void MyGLWidget::iniCamera ()
{
  angleX = angleY = 0.0;
  perspectiva = true;
  ra = 1.0;
  fov = float(M_PI/3.0);
  zn = radiEsc;
  zf = 3*radiEsc;

  left = bottom = -radiEsc;
  top = right = radiEsc;
  projectTransform ();
  viewTransform ();
}
```

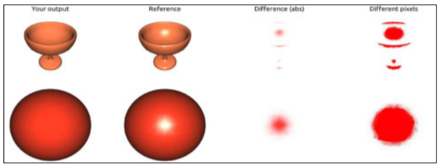
Student j



```
void MyGLWidget::iniCamera ()
{
  angleY = angleX = 0.0;
  perspectiva = true;
  ra = 1.0;
  fov = float(M_PI/3.0);
  fovini = fov/2;
  zn = radiEsc;
  zf = 4*radiEsc;
  OBS = centreEsc+vec3(0,0,radiEsc*2);
  VRP = centreEsc;
  UP = glm::vec3(0,1,0);
  projectTransform ();
  viewTransform ();
}
```

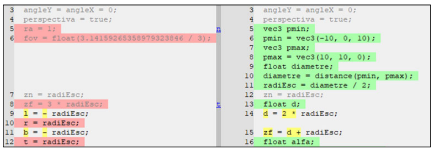
Operational similarity

- Based on pass/fail results when running a test set.
- Useful to group submissions by operational correctness.



Character-level similarity

- Similar to diff, i.e. length of matching blocks over total length.
- Useful to group nearly-identical code.



```
  angleY = angleX = 0;                          angleY = angleX = 0;
  perspectiva = true;                           perspectiva = true;
  ra = 1;                                   n   vec3 pmin;
  fov = float(3.14159265358979323846 / 3);      pmin = vec3(-10, 0, 10);
                                                vec3 pmax;
                                                pmax = vec3(10, 10, 0);
                                                float diametre;
                                                diametre = distance(pmin, pmax);
                                                radiEsc = diametre / 2;
  zn = radiEsc;                                 zn = radiEsc;
  zf = 3 * radiEsc;                         t   float d;
  l = - radiEsc;                                d = 2 * radiEsc;
  r = radiEsc;
  b = - radiEsc;                                zf = d + radiEsc;
  t = radiEsc;                                  float alfa;
```

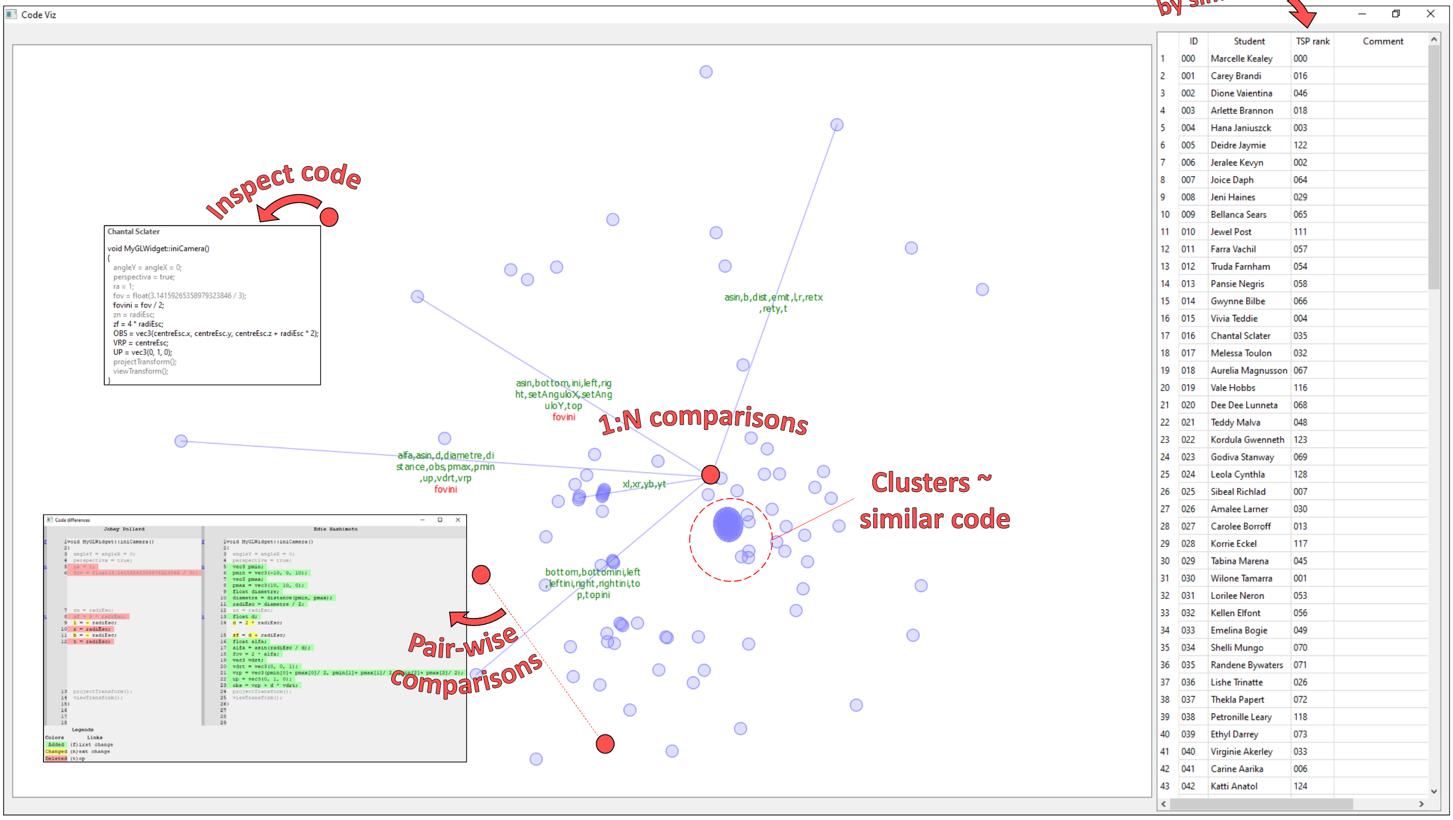
Semantic similarity

- Based on features extracted through a high-level Python API [AVV20].
- Relevant features are found with a χ^2 test based on test pass/fail ratio.
- Useful to group by efficiency (e.g. nested loops), quality (e.g. wrong coordinate space) or robustness (e.g. float equality comparisons).

```
# Manual rubric
R("gl_Position in wrong space",
 "clip" not in vs.space("gl_Position"))

# Automatic rubric
R("Calls to cross", vs.numCalls("cross"))
```

User Interface


Code Viz
Sort submissions by similarity
Inspect code
1:N comparisons
Clusters ~ similar code
Pair-wise comparisons
asin,b,dist,emit,l,r,retx,rety,t
asin,bottom,ini,left,right,setAnguloX,setAnguloY,top
fovini
alfa,asin,d,diametre,distance,obs,pmax,pmin,up,vdrt,vrp
fovini
xl,xr,yb,yt
bottom,bottomini,left,leftini,right,rightini,top,topini

Chantal Sclater

```
void MyGLWidget::iniCamera()
{
  angleY = angleX = 0;
  perspectiva = true;
  ra = 1;
  fov = float(3.14159265358979323846 / 3);
  fovini = fov / 2;
  zn = radiEsc;
  zf = 4 * radiEsc;
  OBS = vec3(centreEsc.x, centreEsc.y, centreEsc.z + radiEsc * 2);
  VRP = centreEsc;
  UP = vec3(0, 1, 0);
  projectTransform();
  viewTransform();
}
```

Code differences

```
                Jobey Pollerd                                        Edie Hashimoto

f void MyGLWidget::iniCamera()                       f void MyGLWidget::iniCamera()
  {                                                    {
    angleY = angleX = 0;                                 angleY = angleX = 0;
    perspectiva = true;                                  perspectiva = true;
n   ra = 1;                                          n   vec3 pmin;
    fov = float(3.14159265358979323846 / 3);             pmin = vec3(-10, 0, 10);
                                                         vec3 pmax;
                                                         pmax = vec3(10, 10, 0);
                                                         float diametre;
                                                         diametre = distance(pmin, pmax);
                                                         radiEsc = diametre / 2;
    zn = radiEsc;                                        zn = radiEsc;
t   zf = 3 * radiEsc;                                t   float d;
    l = - radiEsc;                                       d = 2 * radiEsc;
    r = radiEsc;
    b = - radiEsc;                                       zf = d + radiEsc;
    t = radiEsc;                                         float alfa;
                                                         alfa = asin(radiEsc / d);
                                                         fov = 2 * alfa;
                                                         vec3 vdrt;
                                                         vdrt = vec3(0, 0, 1);
                                                         vrp = vec3(pmin[0]+ pmax[0]/ 2, pmin[1]+ pmax[1]/ 2, pmin[2]+ pmax[2]/ 2);
                                                         up = vec3(0, 1, 0);
                                                         obs = vrp + d * vdrt;
    projectTransform();                                  projectTransform();
    viewTransform();                                     viewTransform();
  }                                                    }

        Legends
Colors        Links
Added         (f)irst change
Changed       (n)ext change
Deleted       (t)op
```

	ID	Student	TSP rank	Comment
1	000	Marcelle Kealey	000	
2	001	Carey Brandi	016	
3	002	Dione Vaientina	046	
4	003	Arlette Brannon	018	
5	004	Hana Janiuszck	003	
6	005	Deidre Jaymie	122	
7	006	Jeralee Kevyn	002	
8	007	Joice Daph	064	
9	008	Jeni Haines	029	
10	009	Bellanca Sears	065	
11	010	Jewel Post	111	
12	011	Farra Vachil	057	
13	012	Truda Farnham	054	
14	013	Pansie Negris	058	
15	014	Gwynne Bilbe	066	
16	015	Vivia Teddie	004	
17	016	Chantal Sclater	035	
18	017	Melessa Toulon	032	
19	018	Aurelia Magnusson	067	
20	019	Vale Hobbs	116	
21	020	Dee Dee Lunneta	068	
22	021	Teddy Malva	048	
23	022	Kordula Gwenneth	123	
24	023	Godiva Stanway	069	
25	024	Leola Cynthla	128	
26	025	Sibeal Richlad	007	
27	026	Amalee Larner	030	
28	027	Carolee Borroff	013	
29	028	Korrie Eckel	117	
30	029	Tabina Marena	045	
31	030	Wilone Tamarra	001	
32	031	Lorilee Neron	053	
33	032	Kellen Elfont	056	
34	033	Emelina Bogie	049	
35	034	Shelli Mungo	070	
36	035	Randene Bywaters	071	
37	036	Lishe Trinatte	026	
38	037	Thekla Papert	072	
39	038	Petronille Leary	118	
40	039	Ethyl Darrey	073	
41	040	Virginie Akerley	033	
42	041	Carine Aarika	006	
43	042	Katti Anatol	124	

MDS embedding of all submissions

Submission list

Benefits

Before grading:

- Instructors were able to spot clusters immediately.
- The tool helped checking whether clusters corresponded to uncompleted exercises, similar approaches, or just copies.
- This analysis provided insights to define grading criteria.

During grading:

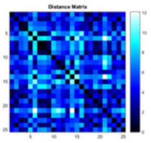
- Submissions could be graded in the TSP rank order, with similar submissions being graded together. Some submissions could be graded in seconds.
- Instructors reported more consistent scores.

After grading:

- The tool facilitated collecting evidences for plagiarism suspicions.

Limitations

- Our current prototype only supports C++ / GLSL code.
- Useful for exercises requiring small pieces of code (up to 100 lines).
- Dissimilarity matrices have quadratic cost. For massive groups, the approach should operate hierarchically, or on a representative subset.



Future work

- User study to evaluate and quantify these advantages.
- Add further software metrics.
- Add output scores from plagiarism detection software.

Acknowledgments

This work has been funded by the Spanish Ministry of Economy and Competitiveness and FEDER Grant TIN2017-88515-C2-1-R.