predictkey (Calls: 1, Time: 5.942 s)

Generated 31-Mar-2019 14:30:17 using performance time.
function in file /Users/saniuri/Documents/MATLAB/dsp-final-project/predictkey.m
Copy to new window for comparing multiple runs

Refresh

☑ Show parent functions ☑ Show busy lines ☑ Show child functions

☑ Show Code Analyzer results ☑ Show file coverage ☑ Show function listing

Parents (calling functions)

Function Name	Function Type	Calls
keyclassification	function	1

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
7	`f0 = pitch(signal(:, 1), fs, '...`	1	5.781 s	97.3%	█████
18	`if closest == ndballmin(r,c)`	719400	0.061 s	1.0%	|
22	`end`	719400	0.054 s	0.9%	|
14	`ndballmin = abs(ndball - f0(i)...`	5995	0.008 s	0.1%	
17	`for c=1:col`	59950	0.008 s	0.1%	
All other lines			0.031 s	0.5%	|
Totals			5.942 s	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time	% Time	Time Plot
pitch	function	1	5.766 s	97.0%	█████
databasefrequencies	script	1	0.001 s	0.0%	
databasekeys	script	1	0.001 s	0.0%	
Self time (built-ins, overhead, etc.)			0.174 s	2.9%	▌
Totals			5.942 s	100%	

Code Analyzer results

Line number	Message

tes database

43	The value assigned here to 'noteval' appears to be unused. Consider replacing it by ~.
60	The value assigned here to 'bestmatchloc' appears to be unused. Consider replacing it by ~.

Coverage results

Show coverage for parent directory

Total lines in function	73
Non-code lines (comments, blank lines)	27
Code lines (lines that can run)	46
Code lines that did run	46
Code lines that did not run	0
Coverage (did run/can run)	100.00 %

Function listing

Color highlight code according to time

```
   time     Calls   line
                      1  function predictedkey = predictkey(signal, fs)
                      2
                      3  % setup
   0.002        1     4  databasefrequencies;
   0.001        1     5  databasekeys;
                      6
   5.781        1     7  f0 = pitch(signal(:, 1), fs, 'Method', 'CEP');
 < 0.001        1     8  detectednotes = zeros(length(f0), 1);
 < 0.001        1     9  [row, col] = size(ndball);
                     10
                     11  % finds the minimum difference between the frequency and the no
                     12  % and selects the note
 < 0.001        1    13  for i=1:length(f0)
   0.008     5995    14      ndballmin = abs(ndball - f0(i));
   0.004     5995    15      closest =  min(ndballmin, [], 'all');
   0.001     5995    16      for r=1:row
   0.008    59950    17          for c=1:col
   0.061   719400    18              if closest == ndballmin(r,c)
 < 0.001     5995    19                  detectednotes(i) = c;
                     20                  % detectednotes(i) is a list of detected notes
 < 0.001     5995    21              end
   0.054   719400    22          end
   0.004    59950    23      end
   0.002     5995    24  end
                     25
                     26  % counts the notes that occurred
 < 0.001        1    27  notecount = zeros(1, 12);
 1;

best match

only the candidates
red

%if the the top n note matches a note in the candidate key rows then
candidates(k))+1; %increment number of matches in the candidate

s -> that is
```

```
< 0.001      1   28 for i = 1:length(detectednotes)
< 0.001   5995   29     notecount(detectednotes(i)) = notecount(detectednotes(i)) +
  0.001   5995   30 end
                 31 
                 32 % notecount - counts per note
                 33 % notelist - counts per note but is tampered
                 34 
                 35 % keep getting the the max occuring note until there is only 1
< 0.001      1   36 notelist = notecount;
  0.001      1   37 notesmatcher = zeros(12, 1);
< 0.001      1   38 candidates = 1:12;
< 0.001      1   39 topn = 1;
< 0.001      1   40 matchedkeys = 10;
< 0.001      1   41 while matchedkeys > 1
                 42     % get the maximum note
  0.002      8   43     [noteval, noteloc] = max(notelist);
< 0.001      8   44     notesmatcher(topn) = noteloc;
< 0.001      8   45     notelist(noteloc) = 0;
                 46 
                 47     % get the counts of matched notes
< 0.001      8   48     notesmatched = zeros(1, 12);
< 0.001      8   49     for k=1:length(candidates) %for each candidate keys, match
< 0.001     37   50         for n=1:topn %for each (current) topnotes being conside
< 0.001    112   51             for m=1:7 %for each row/harmonic in the kdball
< 0.001    784   52                 if notesmatcher(n) == kdball(m, candidates(k))
< 0.001     95   53                     notesmatched(candidates(k)) = notesmatched(
< 0.001     95   54                 end
< 0.001    784   55             end
< 0.001    112   56         end
< 0.001     37   57     end
                 58 
                 59     % find the best matches
  0.001      8   60     [bestmatchval, bestmatchloc] = max(notesmatched);
                 61 
                 62     % only the best candidates will remain, until only 1 remain
                 63     % the predicted key
                 64 
                 65     % update candidates
< 0.001      8   66     candidates = 1:12;
  0.001      8   67     candidates = candidates(notesmatched==bestmatchval);
                 68 
< 0.001      8   69     matchedkeys = sum(notesmatched==bestmatchval);
< 0.001      8   70     topn = topn+1;
< 0.001      8   71 end
                 72 
< 0.001      1   73 predictedkey = candidates;
```