A Twist on Basis Pursuit

I was too tired to ask you about this problem last night on the phone. But here it is. First a summary of what I know:

Basis Pursuit:
$min|z|_1$ $s.t.$ $Wz = x$

The relaxed version, Basis Pursuit De-Noising:
$min_z \|x - Wz\|_2 + \lambda|z|_1$

Finally, Sparse Coding:
$min_{z,W} \|x - Wz\|_2 + \lambda|z|_1$

Ok, great. If I have lots of x_i's (data vectors to synthesize), then they are the columns of x and the codes z_i are the columns of z and so when I write $min\|z\|_1$ I really mean to minimize the sum of the 1-norms of the columns of z.

Now, assume that the input data to synthesize is *ordered*, that is the columns of x can't be permuted without changed the nature of the problem. An example are video frames arranged as columns of the matrix x. Let us define a new problem in such a scenario. First define a new matrix of coefficients z^* such that z^* is identical to z except that the columns have been shifted to the right (or left) by 1. That is if $z = \begin{bmatrix} | & | & & | \\ z_1 & z_2 & ... & z_n \\ | & | & & | \end{bmatrix}$ then $z^* = \begin{bmatrix} | & | & & | \\ 0 & z_1 & ... & z_{n-1} \\ | & | & & | \end{bmatrix}$.

We can define the family of problems, i.e. from Basis Pursuit culminating in the Sparse Coding-like Problem:

$min_{z,W} \|x - Wz\|_2 + \lambda|z - z^*|_1$

The $|z - z^*|_1 = |z_1|_1 + |z_2 - z_1|_1 + ... + |z_n - z_{n-1}|_1$. If you have a constant sequence then it reduces to original sparse coding. So I implemented an auto-encoder which implements a 'poor man's' solution so this problem, and the results are similar to sparse coding yet different. Convergence is faster and quality of solutions are better. One might think that in order to minimize L_1 difference between codes one might try to minimize the L_1 norm of the individual codes. Does this reduce to something simple?