

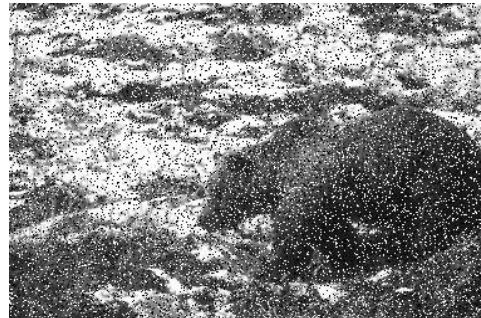
# P-HGRMS: A parallel hypergraph based root mean square algorithm for Image Denoising

Tejaswi Agarwal, Saurabh Jha and Rajesh Kanna  
School of Computing Science and Engineering,  
Vellore Institute of Technology, Chennai, India

**Motivation:** Designing an appropriate algorithm for image neighbourhood hypergraph model to remove noise and compare its effectiveness with the existing model which had limitations such as expensive computation, existence of residual generation of noisy pixels.



Original Image



20 % noise



Denoised Image  
using P-HGRMS

# The P-HGRMS model

80	84	85	95	110
V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	V <sub>4</sub>	V <sub>5</sub>
123	112	97	110	95
V <sub>6</sub>	V <sub>7</sub>	V <sub>8</sub>	V <sub>9</sub>	V <sub>10</sub>
139	167	150	145	142
V <sub>11</sub>	V <sub>12</sub>	V <sub>13</sub>	V <sub>14</sub>	V <sub>15</sub>
89	90	116	120	130
V <sub>16</sub>	V <sub>17</sub>	V <sub>18</sub>	V <sub>19</sub>	V <sub>20</sub>
210	220	213	244	10
V <sub>21</sub>	V <sub>22</sub>	V <sub>23</sub>	V <sub>24</sub>	V <sub>25</sub>

1. Input Image  
Set, Geometrical parameter =15  
Topological parameter =2



2. Construct Image  
neighbourhood hypergraph. Not  
all hyperedges marked here.

3	5	6	7	5
3	6	10	8	6
2	1	4	4	4
3	3	6	5	5
3	3	3	1	1

3. Find cardinality and mark  
isolated hyperedges  
with cardinality less than 3

3	5	6	7	5
5	6	10	8	6
3	5	7	7	6
3	3	6	5	6
3	3	3	7	5

5. Continue from step 2. At this  
stage no isolated hyperedges  
were found. So STOP.

80	84	85	95	110
V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	V <sub>4</sub>	V <sub>5</sub>
123	112	97	110	95
V <sub>6</sub>	V <sub>7</sub>	V <sub>8</sub>	V <sub>9</sub>	V <sub>10</sub>
140	142	150	145	142
V <sub>11</sub>	V <sub>12</sub>	V <sub>13</sub>	V <sub>14</sub>	V <sub>15</sub>
89	90	116	120	130
V <sub>16</sub>	V <sub>17</sub>	V <sub>18</sub>	V <sub>19</sub>	V <sub>20</sub>
210	220	213	145	148
V <sub>21</sub>	V <sub>22</sub>	V <sub>23</sub>	V <sub>24</sub>	V <sub>25</sub>

$$4. f(x,y) = \sqrt{\frac{1}{mn} \sum_{s,t} (g(s,t))^2} \quad (s,t) \in S_{x,y}$$

## Results:

- Maintains noise removal efficiency of the existing algorithm
- Outperforms the existing implementation by 6x-18x in terms computation efficiency.