MAPPING IMAGE RESTORATION TO A GRAPH PROBLEM

Hiroshi Ishikawa and Davi Geiger

Courant Institute of Mathematical Sciences
New York University
251 Mercer Street, New York, NY 10012, USA
ishikawa@cs.nyu.edu, geiger@cs.nyu.edu

ABSTRACT

We propose a graph optimization method for the restoration of gray-scale images. We consider an arbitrary noise model for each pixel location. We also consider a smooth constraint where the potentials between neighbor pixels are convex functionals. We show how to map this problem to a directed flow graph. Then, a global optimal solution is obtained via the use of the maximum-flow algorithm. The algorithm runs in a polynomial time with respect to the size of the image.

1. INTRODUCTION

In image restoration, a "true" image is corrupted by noise and the goal is to recover the "true" image from the noisy one. The modeling needs to remove the noise without removing the intensity discontinuities of an image, i.e., one can not simply remove the high frequency component of the image signal. In the segmentation problem, one seek a map from the set of pixels to a small set of levels such that each connected component of the set of pixels with the same level forms a relatively large and "meaningful" region.

There are various approaches and methods to these problems. We focus on a variational approach, where freedom is given to model geometrical properties through the potential between pixels. An usual difficulty of variational approaches are the lack of guaranteed and efficient methods to find the solution.

For a class of convex potentials between neighbor pixels and arbitrary noise model at each pixel location, we study a guaranteed and efficient algorithm. The idea is to map the problem to a minimum-cut problem on a directed graph, which can be solved globally in a polynomial time with a maximum-flow algorithm.

1.1 Background

A variational framework for image restoration and segmentation that first clearly address the problem of removing the noise preserving the discontinuities is given by Blake and Zisserman [1]; and Geman and Geman [8].

A limitation with both work is the lack of guaranteed and efficient optimization method. In [8] they used the guaranteed but very slow simulated annealing method while in [1] they used the very slow (and not guaranteed) GNC method. Various work to speed up these computations, e.g., [7] and related EM methods, lacked guaranteed methods.

For binary images, Greig, Porteous, and Seheult [9][10], studying the Geman and Geman's model, provided an efficient and optimal solution based on the maximum flow algorithm. They compared experimentally the two methods (simulated annealing versus the maximum-flow algorithm) to conclude that the guaranteed method were capable of achieving better image restoration. In particular they noticed that the simulated annealing method tended to over-smooth the noisy image.

In order to extend this work to gray-scale images (more than two levels), Ferrari, Frigessi, and de Sá [6] have shown that for arbitrary potential this problem is in general NP-hard.

More recently, Roy and Cox [14] have considered the use of network-flow algorithms (in an undirected graph) for computer stereo vision matching. Boykov, Veksle, and Zabih [2] used an approximate multiway-cut algorithm (in an undirected graph) to solve it approximately for a specific type of potential function that models discontinuities. The authors [11][12] have introduced the use of a directed graph maximum-flow for binocular symmetric stereo and studied image segmentation using network flow algorithms.

Here we focus on the image restoration problem, clearly stating the set of potentials where the maximum-flow algorithm in a directed graph can be applied. We show the experimental value of this approach.

2. IMAGE RESTORATION

Let the input g to be an image corrupted by noise. We can typically raster scan an image and so g is represented by a vector in an N^2-dimensional vector space (for a square image of size $N\times N$.) Then, g_k ($k=1,\cdots,N^2$) represents the gray-scale value at pixel k. The variational formulation of image restoration tries to find an output image f that minimizes the energy:

$$E(f)=\sum_{k=1}^{N^2}\left\{G(f_k-g_k)+\sum_{j\in N_k}F(f_k-f_j)\right\}, \qquad (1)$$

where G is an arbitrary noise model function and F is a potential. The role of F is to encourage the output f to be smooth, removing noise. However, image discontinuities need to be preserved.

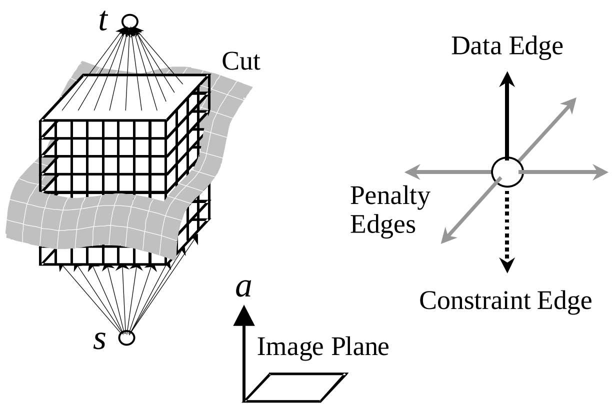

Figure 1. A directed graph. A cut of the graph can be thought of as a surface that separates the two parts. The optimal cut is the one that minimizes the sum of the capacities associated to the cut edges. Each node has six outgoing edges (except for boundary).

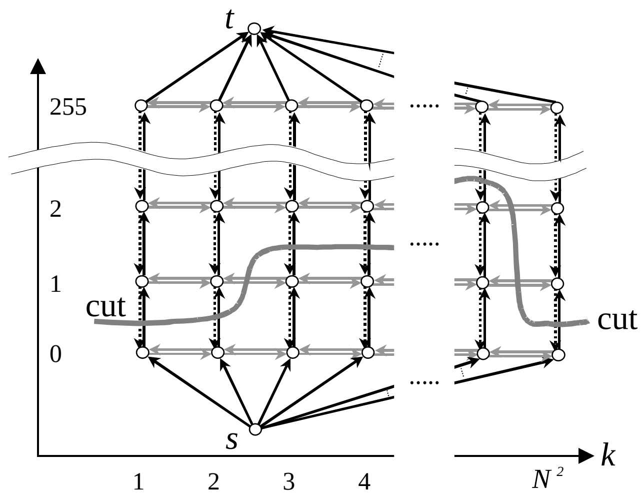

Figure 2. Data edges are depicted as black arrows. The cut here represents the output image value $f_1=0$, $f_2=0$, $f_3=1$, $f_4=1$, $f_{N^2-1}=2$, and $f_{N^2}=0$. Penalty edges are represented by gray arrows. By crossing consecutive penalty capacities, the cost is added linearly, accounting for the function $F(x)=|x|$. Constraint edges are depicted as dotted arrows. They ensure that the cut represents a function. These edges cannot be cut, preventing the cut from "going back."

Therefore, very large gradient of f should not be penalized. For our method to be applicable, F must be a convex function, as discussed in section 4. Among convex functions, the linear function on the magnitude of the change (i.e., $F(x)=\mu|x|$) is most preferable as it least penalizes large discontinuities.

3. GRAPH FORMULATION

In this section, we explain the segmentation assignment architecture that utilizes the maximum-flow algorithm to obtain the globally optimal assignment, with respect to the energy (1). We assume the gray-scale value ranges from 0 to 255.

3.1 The Directed Graph

We devise a directed graph and let a cut represent the output function $k \mapsto f_k$ so that the minimum cut corresponds to the optimal output. Let $\mathcal{M}$ be the set of all possible local value, or *hypothesis*, i.e.,

$$\mathcal{M} = \{ (m,k) \mid m \in \{0, \ldots 255\}, k \in \{1,\ldots, \mathrm{N}^2\} \}.$$

We define a directed graph $G=(V,E)$ as follows:

$$V = \{ u_{mk} \mid (m,k) \in \mathcal{M} \} \cup \{s,t\}$$
$$E = E_{\mathrm{D}} \cup E_{\mathrm{C}} \cup E_{\mathrm{P}}.$$

In addition to the source s and the sink t, the graph has a vertex u_{mk} for each hypothesis $(m,k) \in \mathcal{M}$. The set E of edges is divided into subsets E_{D}, E_{C}, and E_{P}, each one having a capacity with a precise meaning in terms of the model (1), which we will explain in the following subsections.

We denote a directed edge from vertex u to vertex v as (u,v). Each edge (u,v) has a nonnegative capacity $c(u,v)$. A *cut* of G is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$ (see Figure 1). We mean by a cut of an edge (u,v) that $u \in S$ and $v \in T$. This is the only case that the cost $c(u,v)$ of the edge contributes to the total cost $\sum_{u \in S, v \in T} c(u,v)$. We note that if the cut is through the edge (u,v) with $u \in T$ and $v \in S$, the cost is $c(v,u)$, which in general is different from $c(u,v)$. It is well known that by solving a maximum-flow problem one can obtain a *minimum cut*, a cut that minimizes the total cost over all cuts. (See [3].) Because of the way we define the capacity of each edge, the resulting minimum cut exactly represents the output image that minimizes the energy (1). To see this, let us now analyze the different set of edges E_{D}, E_{C}, and E_{P}.

3.2 Data Edges

From each vertex u_{mk}, there is one outgoing *data edge*: $(u_{mk}, u_{m(k+1)})$ if $k < 255$, or (u_{mk}, t) otherwise. It has a capacity $G(m - g_k)$. Thus, the capacities associated to these edges contribute to account for the first term of (1). We denote the set of data edges by E_{D}. If a data edge originating from u_{mk} is cut, we interpret that the output function f has gray-scale value m at pixel k. Figure 2 shows the nodes and data edges. The cut shown represents the output image value $f_1=0, f_2=0, f_3=1, f_4=1, f_{N^2-1}=2$, and $f_{N^2}=0$. Also, edges (s, u_{1k}) are added for all k with an infinite capacity. Note these edges are actually unnecessary and s and first layer vertices $\{u_{1k} \mid k = 1 \ldots N^2\}$ can be merged to one vertex, but for clarity are shown thus.

3.3 Penalty Edges

Penalty edges are defined as

$$E_P = \{(u_{mk}, u_{mj}) \mid (m, k) \in \mathcal{M}; j \in N_k\}$$

These edges are for paying for discontinuities (region boundaries). Edges in E_P are cut whenever a change in the gray-scale value occurs. For instance, if the output image has a gray value a at pixel k and $a+2$ at pixel $k+1$, two edges will be cut, namely $(u_{(a+1)(k+1)}, v_{(a+1)k})$ and $(u_{(a+2)(k+1)}, v_{(a+2)k})$. We set the capacity to be some constant value. By crossing consecutive penalty capacities the cost is added linearly, yielding a cost function $F(x) = |x|$. While we have used a simple connectivity and capacity setting here, we could seek more general connectivity of the form

$$E_P = \{(u_{mk}, u_{lj}) \mid (m, k), (l, j) \in \mathcal{M}; j \in N_k\},$$

with arbitrary capacity.

By setting the capacity for these edges, we control the potential function $F(x)$ between levels. We prove in the next section that for any general form of connectivity graph, where maximum-flow can be applied, the edge penalty must yield convex potential function $F(x)$. This follows from the requirement that the capacities must be nonnegative. Conversely, it can be shown that any convex function $F(x)$ can be used.

3.4 Constraint Edges

Constraint edges ensure that the cut expresses a function, i.e., that each pixel is assigned only one gray-scale value.

$$E_C = \{(u_{mk}, u_{(m-1)k}) \mid (m, k) \in \mathcal{M}, m > 1\}$$

The capacity of each constraint edge is set to infinity. Therefore, any cut with a finite total flow cannot cut these edges. Note that, because the edges have directions, a constraint edge prevents only one of two ways to cut them. In Figure 2, constraint edges are depicted as dashed arrows, and none is cut.

4. ADMISSIBLE POTENTIAL

Since a maximum-flow algorithm runs in a polynomial time, the formulation gives a method to find a global minimum of energy (1). Yet the energy (1) is in general NP-hard. As one would expect from this fact, the formulation has a limitation on the potential $F(x)$:

The potential function $F(x)$ must be convex.

Proof. Each node u_{mk} can in general have as a neighbor any node in column over a neighboring pixel, i.e.,

$$c(u_{mk}, u_{lj}) \neq 0 \Rightarrow j \in N_k$$

The potential function $F(x)$, at pixel k, is the result of a cut through various edges bridging nodes u_{mk} and u_{lj}, where $j \in N_k$. Let us focus on the bridge between pixel k and j. In the directed graph, this cut goes through a column of hypotheses separating these two pixels k and j. Given the gray-scale value change from m at pixel k to l at pixel j, we have the cost as follows:

$$\tilde{F}(m,l) = \sum_{m'=0}^{m} \sum_{l'=l+1}^{M} c(u_{m'k}, u_{l'j}) + \sum_{m'=m+1}^{M} \sum_{l'=0}^{l} c(u_{l'j}, u_{m'k}) \qquad (2)$$

where M is the maximum gray-scale value. The potential depends only on the difference of m and l. Hence we have $F(x) = \tilde{F}(m, x-m)$, setting $x = m - l$.

To show the second derivative of $F(x)$ is non-negative, we take

$$\frac{\delta^2 F(x)}{\delta x^2} = \{F(x+1) - F(x)\} - \{F(x) - F(x-1)\}$$
$$= \tilde{F}(m, l-1) - \tilde{F}(m, l) - \tilde{F}(m-1, l-1) + \tilde{F}(m-1, l),$$

where we used the invariance of F under a translation of m and l with $x = m - l$ fixed. After using (2) and simplifying the summation, we get

$$\frac{\delta^2 F(x)}{\delta x^2} = c(u_{mk}, u_{lj}) + c(u_{lj}, u_{mk}) .$$

Since all edge capacities are non-negative, the discontinuity penalty cost $F(x)$ must be a convex function, i.e., second derivative is always nonnegative.

5. RESULT

We implemented the architecture explained above. For maximum-flow algorithm we used standard push-relabel method with global relabeling [5].

Figure 3 shows three restoration examples. Figure 4 shows a comparison of three restorations of images with three noise levels.

6. REFERENCES

[1] A. Blake and A. Zisserman. *Visual reconstruction.* MIT Press, Cambridge, Mass., 1987.

[2] Y. Boykov, O. Veksle, and R. Zabih. "Markov random fields with efficient approximations." In *Proc. IEEE Conf. CVPR*, Santa Barbara, Calif., June 1998, pp. 648–655.

[3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms.* McGrow-Hill, New York, 1990.

[4] J. F. Canny. "A computational approach to edge detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[5] B. V. Cherkassky and A. V. Goldberg. "On implementing push-relabel method for the maximum flow problem." In *Proc. 4th Int. Programming and Combinatorial Optimization Conf.*, pp. 157–171, 1995.

[6] P. A. Ferrari, A. Frigessi, and P. de Sá. "Fast approximate maximum a posteriori restoration of multicolour images." *J. R. Statist. Soc.* B, 57, 485-500, 1995.

[7] D. Geiger and F. Girosi. "Parallel and deterministic algorithms for MRFs: surface reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13:401–412, 1991.

[8] S. Geman and D. Geman. "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:721–741, 1984.

[9] D. M. Greig, B. T. Porteous, and A. H. Seheult. Discussion of: "On the statistical analysis of dirty pictures" (by J. E. Besag.) *J. R. Statist. Soc.* B, 48, 282-284, 1986.

[10] D. M. Greig, B. T. Porteous, and A. H. Seheult. "Exact maximum a posteriori estimation for binary images." *J. R. Statist. Soc.* B, 51, 271-279, 1989.

[11] H. Ishikawa and D. Geiger. "Occlusions, discontinuities, and epipolar lines in stereo." In *Fifth European Conference on Computer Vision*, Freiburg, Germany, Springer-Verlag. LNCS 1406, pp. 232–248, June 1998.

[12] H. Ishikawa and D. Geiger. "Segmentation by Grouping Junctions." In *Proc. IEEE Conf. CVPR*, Santa Barbara, Calif., June 1998, pp. 125–131.

[13] D. Mumford and J. Shah. "Boundary detection by minimizing functionals, I." In *Proc. IEEE Conf. CVPR*, San Francisco, Calif., 1985.

[14] S. Roy and I. Cox. "A maximum-flow formulation of the N-camera stereo correspondence problem." *ICCV98*, Bombai, India, pp. 492–499, 1998.

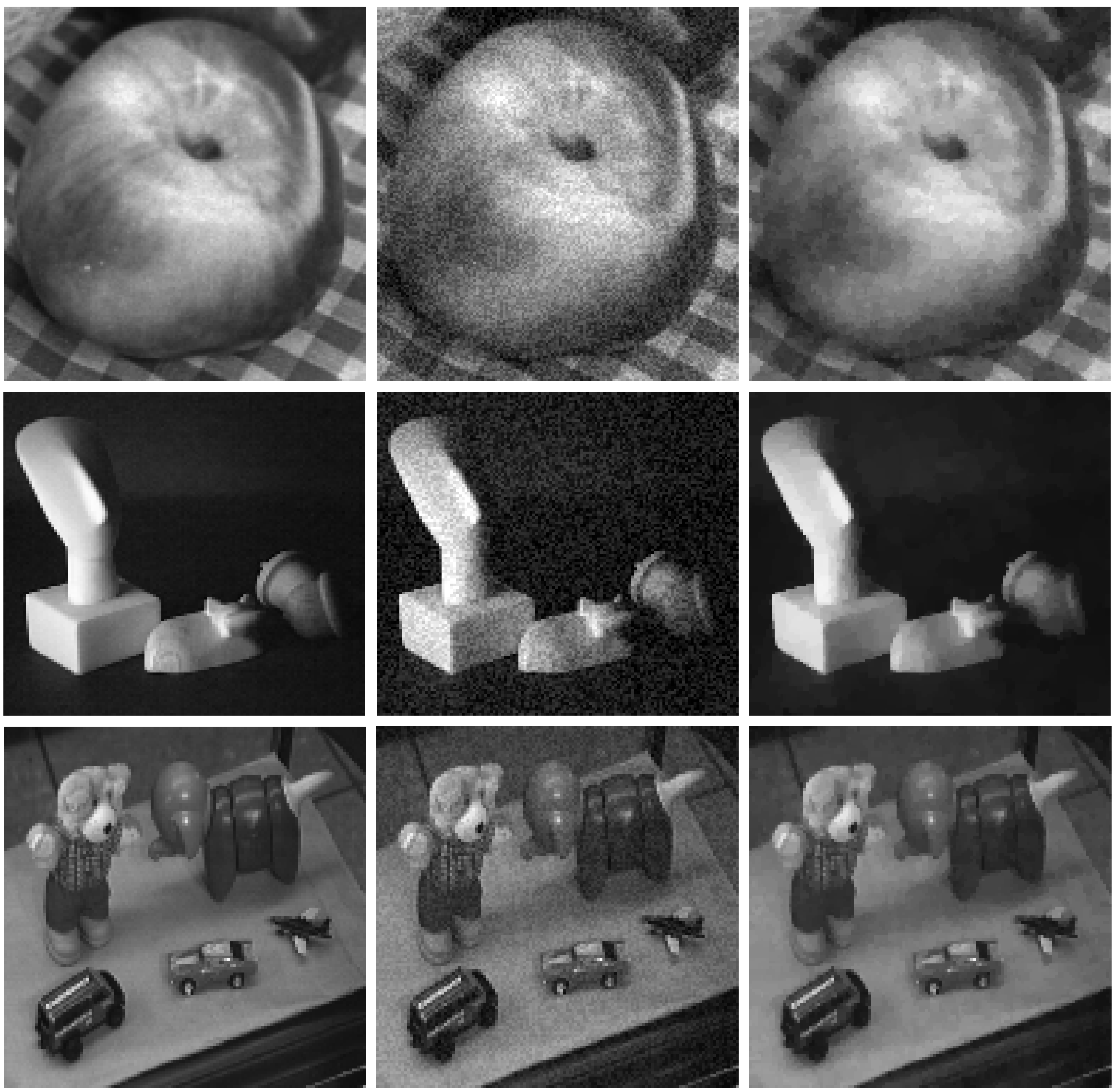

Figure 3. Each row shows an example of restoration; original image (left column), noisy image (middle), and restored image (right).

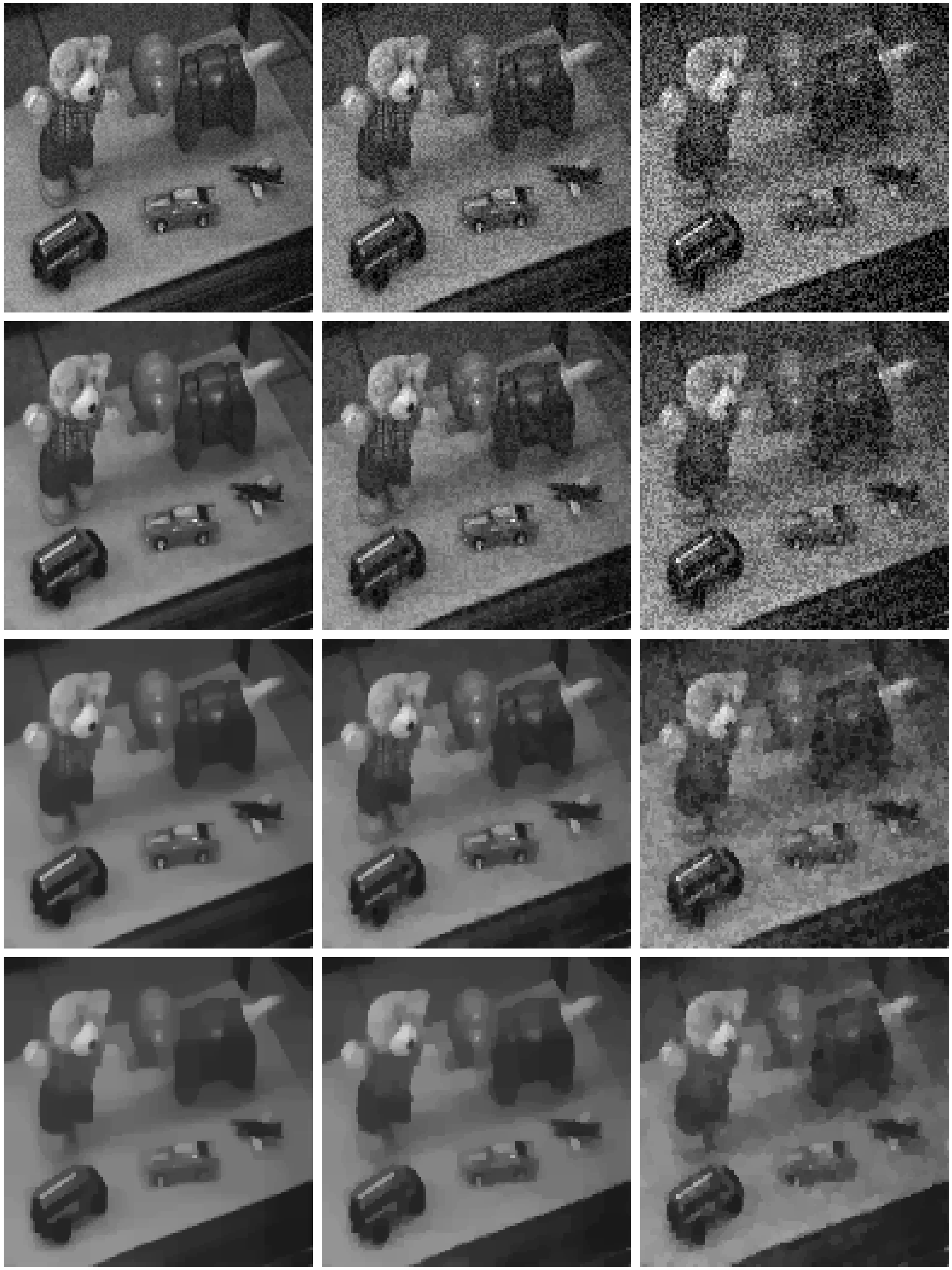

Figure 4. Comparison of three restoration of images with three noise levels (original image is shown in the bottom left of Figure 3.) Each column shows input (top row) and restoration using three smoothing constant μ's; μ=6 (second row), μ=20 (third row), and μ=40 (bottom), where $F(x)=\mu|x|$.