

基于 Hausdorff 距离的图像配准研究

牛力丕 毛士艺 陈 炜

(北京航空航天大学电子信息工程学院 北京 100083)

摘 要: 针对图像配准中常出现的 RST(旋转-比例-平移)变换, 推导出了相应的盒距离变换公式。与传统的基于广义仿射变换的 Hausdorff 盒距离变换公式相比, 缩小了搜索距离空间。在计算 Voronoi 表面时, 根据 Hausdorff 距离的计算需要提出比较滑动窗口的区域 Voronoi 表面, 节省了计算 Voronoi 表面的时间。并且在利用边缘点计算 Hausdorff 距离时, 剔除琐碎的边缘, 仅使用较长的边缘计算。试验结果表明, 这些改进方法较大地提高了基于 Hausdorff 距离的图像配准的计算速度。

关键词: 图像配准; Voronoi 表面; Hausdorff 距离

中图分类号: TN911.73

文献标识码: A

文章编号: 1009-5896(2007)01-0035-04

Image Registration Based on Hausdorff Distance

Niu Li-pi Mao Shi-yi Chen Wei

(Department of Electronic Engineering, Beijing University of Aero. & Astro, Beijing 100083, China)

Abstract: As for the RST transform in image registration, corresponding formula of box distance transform is deduced. Compared with traditional formula of general affine Hausdorff box distance, search range of distance is reduced. The paper proposes regional Voronoi surface combining comparison of sliding windows when computing Hausdorff distance, characterized by reducing calculating-cost for Voronoi surface. It also has the advantages of eliminating trivial edges and preserving longer edges for calculating. Experimental results show that calculation speed of image registration based on Hausdorff distances is improved.

Key words: Image registration; Voronoi surface; Hausdorff distance

1 引言

多传感器图像配准是图像融合的必要前提。图像配准方法可以分为基于灰度信息和基于特征的配准。基于灰度的方法比较适用于同类传感器之间的配准。对于不同类的传感器, 基于灰度的配准方法往往失效。而基于特征的方法是利用图像间共有特征的对对应关系, 所以适合不同传感器间的图像配准^[1,2]。常见的特征匹配方法通常是寻找特征之间的一一对应关系, 计算对应特征之间的相似度量以得到图像之间的变换关系^[1-3]。这些方法中特征对的建立较为复杂, 当特征数目超过一定数量时, 计算时间将成倍地增加。并且当特征的抽取过程中产生虚假特征或者丢失特征时, 基于一一对应关系的配准算法往往很难得到正确的结果。

Hausdorff 距离是匹配点特征的一种方法, 它不需要建立点之间的一一对应关系, 只是计算两个点集之间的相似程度(最大距离), 所以可以有效地处理很多特征点的情况。1993 年, Huttenlocher^[4]提出的部分 Hausdorff 距离的概念, 有效地解决了图像之间发生遮挡和虚假点的问题。1997 年, Rucklidge^[5]提出的子空间分解-盒距离变换算法极大加速了 Hausdorff 距离的计算速度, 使 Hausdorff 距离有了一定的实用价值。但该算法是基于广义仿射变换模型的, 而在实际的

图像配准中, 特别是遥感图像配准中, 图像之间的变换更多的是 RST(旋转-比例-平移)变换, 直接使用广义仿射变换盒距离算法无疑增加了不必要的搜索空间和计算时间。本文针对图像中 RST 变换, 推导出了相应的盒距离变换公式, 缩小了搜索空间, 节省了计算时间。Voronoi 表面是点集图像的距离变换, 是计算 Hausdorff 距离的基础, 针对在 Hausdorff 距离的计算中, 只需计算变换点是否映射在 Voronoi 表面内小于距离 τ 的情况。所以并不需要计算点集图像的完全距离变换, 只需要计算小于距离 τ 的距离变换就可以满足 Hausdorff 距离的计算要求。本文提出了比较滑动窗口的区域 Voronoi 表面, 在满足 Hausdorff 距离计算要求下, 较大地节省了计算 Voronoi 表面的时间。另外, 在计算 Hausdorff 距离时, 点集的疏、密程度极大地影响着计算速度^[5], 应该尽量避免点密集的情况。而在边缘图像中, 琐碎的边缘点(或者短小的边缘)一般代表着较不重要的信息, 所以可以剔除掉, 只保留较长的边缘进行 Hausdorff 距离计算。试验结果表明, 这些改进方法较大地提高了基于 Hausdorff 距离的图像配准的计算速度。

2 Hausdorff 距离^[4]

给定两个有限集合 $A=\{a_1, a_2, \dots, a_p\}$ 和 $B=\{b_1, b_2, \dots, b_q\}$, 则 A, B 之间的 Hausdorff 距离定义为

$$H(A, B) = \max(h(A, B), h(B, A)) \quad (1)$$

其中

$$h(A, B) = \max_{a_i \in A} \min_{b_j \in B} \|a_i - b_j\| \quad (2)$$

$$h(B, A) = \max_{b_j \in B} \min_{a_i \in A} \|b_j - a_i\| \quad (3)$$

式中 $\| \cdot \|$ 表示点集 A, B 之间的距离范数。函数 $h(A, B)$ 和 $h(B, A)$ 分别称为前向和后向 Hausdorff 距离。如果 $h(A, B) = d$, 则表示 A 中所有点到 B 中点的距离不超过 d , 也就是说 A 中点都在 B 中点的距离为 d 的范围之内。

Hausdorff 距离表征了两个点集之间的不相似程度。但它对干扰很敏感。比如, 点集 A 和点集 B 非常相似, 而 B 中仅有一点与 A 相差较大时, $h(B, A)$ 的值就变得很大, 从而 $H(A, B)$ 很大。为了避免这一问题, 1993 年, Huttenlocher 提出了部分 Hausdorff 距离的概念:

$$H^{f_F f_R}(A, B) = \max(h^{f_F}(A, B), h^{f_R}(B, A)) \quad (4)$$

式中

$$h^{f_F}(A, B) = f_F \text{ th } \min_{a_i \in A, b_j \in B} \|a_i - b_j\| \quad (5)$$

$$h^{f_R}(B, A) = f_R \text{ th } \min_{b_j \in B, a_i \in A} \|b_j - a_i\| \quad (6)$$

这里 $f_F, f_R \in [0, 1]$ 分别称为前向分数和后向分数, 控制着前向距离和后向距离, th 表示排序。当 $f_F = f_R = 1$ 时, 该公式退化为原始的 Hausdorff 距离。

3 Rucklidge 快速计算 Hausdorff 距离的算法^[5]

假设有两幅图像边缘点集 A 和 B , 以 B 为参考位置, 将 A 进行变换, 此时的 Hausdorff 距离为

$$D[t] = \max(d[t], d'[t]) \quad (7)$$

式中 $d[t] = h^{f_F}(t[A], B)$, $d'[t] = h^{f_R}(B, t[A])$, t 为图像 A 的变换矢量。在此准则下, 将图像 A, B 配准就是寻找变换 t , 使得 $D[t]$ 达到最小。

为了建立快速的计算方法, Rucklidge 首先对广义仿射变换空间进行离散化处理。对于广义仿射变换:

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (8)$$

的 6 个变换参数 $(a_{00}, a_{01}, a_{10}, a_{11}, t_x, t_y)$, 可以组成一个六元数组。

假定图像 A 中的所有点 (x, y) 的取值范围是: $0 \leq x \leq x_{\max}$ 和 $0 \leq y \leq y_{\max}$, 则经过离散化处理后, 离散变换空间的单位矢量为: $e = (1/x_{\max}, 1/y_{\max}, 1/x_{\max}, 1/y_{\max}, 1, 1)$ 。这样处理的好处在于: 离散变换空间中的两个相邻变换 t_1 和 t_2 (相邻就是指变换的 6 个参数中只有一个不同, 且相差为 1 个单位) 作用于点 $(x, y) \in A$ 所得到的两个坐标点 (x'_1, y'_1) 和 (x'_2, y'_2) 在 x 或 y 方向上相差不超过一个像素。

Rucklidge 的子空间分解算法的步骤是:

(1) 将整个变换空间 t (人为指定的空间) 划分为一系列子空间 $R = (t^{\text{low}}, t^{\text{high}})$ 。其中, $t^{\text{low}} = [a_{00}^{\text{low}}, a_{01}^{\text{low}}, a_{10}^{\text{low}}, a_{11}^{\text{low}}, t_x^{\text{low}}, t_y^{\text{low}}]$,

表示 R 中所有变换参数的最小值。 $t^{\text{high}} = [a_{00}^{\text{high}}, a_{01}^{\text{high}}, a_{10}^{\text{high}}, a_{11}^{\text{high}}, t_x^{\text{high}}, t_y^{\text{high}}]$, 表示 R 中变化参数的最大值。

(2) 考察当前所有的子空间, 对每个子空间 R_i , 用 t_i^{low} 对每个点 $a \in A$ 进行变换。计算点 a 对应的盒距离变换 $\Delta'_{w,h}([t_i^{\text{low}}(a)])$, 如果 $\Delta'_{w,h}([t_i^{\text{low}}(a)]) \leq \tau_F$ 的点的个数不小于 $f_F \cdot n'_A$, 则标记 R_i 为感兴趣的变换空间, 否则删除该空间。

(3) 将当前所有子空间都考察后, 可以生成一个感兴趣子空间列表。对每个感兴趣子空间分解成 2^6 个大小相等的下一级子空间, 重复(2)、(3)步骤, 直到当前的子空间只包含一个单位变换 e 为止。

(4) 对于当前的每个子空间(单位变换 e_i), 如果同时满足 $h^{f_F}(t[A], B) \leq \tau_F$ 和 $h^{f_R}(B, t[A]) \leq \tau_R$, 则接受 t , 否则删除。

此算法的核心部分是第(2)步, $\Delta'_{w,h}[x, y]$ 为图像 B 的盒距离变换, 定义为

$$\Delta'_{w,h}[x, y] = \min_{\substack{0 \leq x' \leq w \\ 0 \leq y' \leq h}} \Delta[x + x', y + y'] \quad (9)$$

其中 $\Delta[x, y] = \min_{b_i \in B} \|(x, y) - b_i\|$ 为二值图像 B 的 Voronoi 表面,

也就是图像 B 的距离变换。对图像二值矩阵 B , 其 Voronoi 变换 Δ 的大小和 B 相同, $\Delta(x, y)$ 表示点 (x, y) 到 B 中的非零元素最近的距离, 特别地, 当 $B(x, y) = 1$ 时, $\Delta(x, y) = 0$ 。

下标 w, h 的定义为

$$w = (a_{00}^h - a_{00}^l) + (a_{01}^h - a_{01}^l) + (t_x^h - t_x^l) \quad (10)$$

$$h = (a_{10}^h - a_{10}^l) + (a_{11}^h - a_{11}^l) + (t_y^h - t_y^l) \quad (11)$$

含义是, 对于任意变换 $t \in R_i = (t_i^l, t_i^h)$, 点 $a \in A$ 的变换 $[a]$ 在图像 B 中的变化范围被限定在一个以 $[t^l(a)]$ 为左上角, 大小为 $(w+1) \times (h+1)$ 的矩形中。由式(8)可知, 如果 $\Delta'_{w,h}([t_i^{\text{low}}(a)]) > \tau_F$, 则不可能存在 $t \in R_i$, 满足 $\Delta([a]) \leq \tau_F$, 所以可以将此子空间 R_i 排除。

4 改进算法

从 Rucklidge 的子空间分解算法步骤(2)可看出, Voronoi 表面只需要判断点 (x, y) 是否大于 τ_F 。这样我们并不需要计算整个二值图像 B 的 Voronoi 表面, 而仅仅计算出以非零点为中心, 半径为 τ_F 的距离变换就可以满足要求了。即

$$\Delta_{\text{area}}[x, y] = \begin{cases} \min_{b_i \in B} \|(x, y) - b_i\|, & \Delta[x, y] \leq \tau_F \\ \tau_F, & \Delta[x, y] > \tau_F \end{cases} \quad (12)$$

具体实现方法为

(1) 设置大小为 $(2 \cdot N + 1) \times (2 \cdot N + 1)$ 的查表矩阵 $T(i, j)$, 其中: N 为不比 τ_F 小的整数, 令

$$T(i, j) = \sqrt{(i - N - 1)^2 + (j - N - 1)^2}$$

(2) 令 $\Delta_{\text{area}}[x, y]$ 为大小与 B 相等, 值为 τ_F 的常数矩阵;

(3) 对 B 中每个非零位置 (x_i, y_i) , 找到 $\Delta_{\text{area}}[x, y]$ 中对应的位置, 与矩阵 $T(i, j)$ 的中心对准, 然后比较区域 $\Delta_{\text{area}}[x, y]$ 与 $T(i, j)$ 的对应值的大小, 只保留小的值;

(4)循环执行(3)直到比较完 B 中所有非零位置。

该方法在计算 Voronoi 表面时, 不使用平方、开根号等计算, 只使用查表和比较运算。所以很大程度上减小了计算 Voronoi 的计算量。

Rucklidge 快速算法中的盒距离变换是基于广义仿射变换模型, 而在图像配准中, 尤其是在遥感图像的配准中, 更多的图像变形是基于 RST 变换模型, 直接套用该盒距离变换, 无疑是加大了不必要的搜索空间, 浪费了搜索时间。

RST 变换公式为

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = s \cdot \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (13)$$

$$\text{即} \quad \begin{cases} x_2 = s \cdot \cos \theta \cdot x_1 + s \cdot \sin \theta \cdot y_1 + t_x \\ y_2 = -s \cdot \sin \theta \cdot x_1 + s \cdot \cos \theta \cdot y_1 + t_y \end{cases} \quad (14)$$

令 $u = s \cdot \cos \theta$, $v = s \cdot \sin \theta$ 。此时, 变换空间有 4 个参数 (u, v, t_x, t_y) , 即

$$(x, y) \rightarrow (u \cdot x + v \cdot y + t_x, -v \cdot x + u \cdot y + t_y) \quad (15)$$

此时, 为了满足离散变换空间中的两个相邻变换 t_1 和 t_2 作用于点 $(x, y) \in A$ 所得到的两个坐标点 (x'_1, y'_1) 和 (x'_2, y'_2) 在 x 或 y 方向上相差不超过一个像素的条件, 对于图像 A 中的点 (x, y) 取值范围: $0 \leq x \leq x_{\max}$ 和 $0 \leq y \leq y_{\max}$, RST 离散变换空间的单位矢量为

$$\left. \begin{aligned} e_1 &= \left(\frac{1}{\max(x_{\max}, y_{\max})}, 0, 0, 0 \right) \\ e_2 &= \left(0, \frac{1}{\max(x_{\max}, y_{\max})}, 0, 0 \right) \\ e_3 &= (0, 0, 1, 0) \\ e_4 &= (0, 0, 0, 1) \end{aligned} \right\} \quad (16)$$

对于子空间 $R_i = (t^{\text{low}}, t^{\text{high}})$, 其中: $t^{\text{low}} = (u^l, v^l, t_x^l, t_y^l)$, $t^{\text{high}} = (u^h, v^h, t_x^h, t_y^h)$ 。

用 t^{low} 和 t^{high} 分别对点 $a = (x, y) \in A$ 进行变换, 可以得到两个点 (x^l, y^l) 和 (x^h, y^h) :

$$\begin{aligned} (x^l, y^l) &= (u^l \cdot x / w_{\max} + v^l \cdot y / w_{\max} + t_x^l, \\ &\quad -v^l \cdot x / w_{\max} + u^l \cdot y / w_{\max} + t_y^l) \end{aligned} \quad (17)$$

$$\begin{aligned} (x^h, y^h) &= (u^h \cdot x / w_{\max} + v^h \cdot y / w_{\max} + t_x^h, \\ &\quad -v^h \cdot x / w_{\max} + u^h \cdot y / w_{\max} + t_y^h) \end{aligned} \quad (18)$$

其中 $w_{\max} = \max(x_{\max}, y_{\max})$ 。

当子空间各个边长彼此相等时, 最有利于快速分解。利用这一特点, 有 $u^h - u^l = v^h - v^l = t_x^h - t_x^l = t_y^h - t_y^l$ 。所以, 理论上盒距离变换的边长为

$$\begin{aligned} w &= x^{\text{high}} - x^{\text{low}} \\ &= (u^h - u^l) \cdot x / w_{\max} + (v^h - v^l) \cdot y / w_{\max} + (t_x^h - t_x^l) \end{aligned} \quad (19)$$

$$\begin{aligned} h &= y^{\text{high}} - y^{\text{low}} \\ &= -(v^h - v^l) \cdot x / w_{\max} + (u^h - u^l) \cdot y / w_{\max} + (t_y^h - t_y^l) \end{aligned} \quad (20)$$

由于, 子空间 R 内任意 v^i 满足: $v^h \geq v^i \geq v^l$, 故 h 应取值为最恶劣的情况, 即在 RST 变换条件下, 盒距离变换边长应该为

$$\left. \begin{aligned} w &= \Delta u \cdot x / w_{\max} + \Delta v \cdot y / w_{\max} + \Delta t_x \\ h &= \Delta u \cdot y / w_{\max} + \Delta t_y \end{aligned} \right\} \quad (21)$$

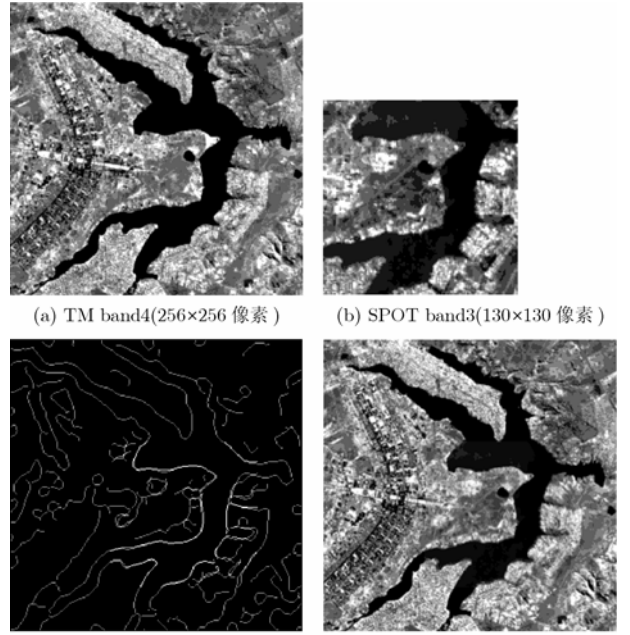
可以看出, 此时的盒距离变换并不像仿射变换那样是正方形, 而是 y 方向变短了的矩形, 包含了更少的距离变换矩阵的面积。更有利于提高算法的搜索速度。

在子空间分解算法中, 点集的疏、密程度极大地影响了计算速度^[5], 应该尽量避免点密集的情况。而在边缘图像中, 琐碎的边缘点(或者短小的边缘)一般代表着较不重要的信息, 所以可以剔除掉。本文在得到边缘图像后, 对图像进行边缘链码提取, 删除掉长度小于 N 的边缘(N 一般选择 5~30), 只对保留的较长的边缘进行计算, 从而进一步加快了图像配准的速度。

5 试验结果

为了验证改进算法的性能, 利用两个例子进行了本文算法与 Rucklidge 算法的对比试验(实验中 Rucklidge 算法也采用 4 参数空间搜索)。在同一实验中, 所有算法都采用相同的参数, 所有算法的程序代码都是用 Matlab 编写, 计算机的配置为: CPU 是 AMD2500+, 操作系统为 Windows 2000 Professional。

实验 1 图 1(a)和图 1(b)分别是同一地区不同波段的图像。图 1(a)为 256×256 像素的 TM band4 图像, 图 1(b)为 130×130 像素的 SPOT band3 图像。在参数范围: $(0.95 \leq s \leq 1.05, -5^\circ \leq \theta \leq 5^\circ, 0 \leq t_x, t_y \leq 100)$ 内搜索。图 1(c)是图 1(a)和图 1(b)边缘图像的配准结果。图 1(d)是图 1(a)与



(c) (a) 与 (b) 的边缘配准图 (d) (a) 与 (b) 的配准图

图 1 实验 1 图像配准结果

1(b)的配准图。仔细对比图 1(a)和配准结果图 1(d), 较难看出拼接的痕迹。

实验 2 图 2(a)和图 2(b)分别是 2000 年上海地区长江口某岛的绿光和蓝光图像。二者存在比例、旋转和平移变化。在参数范围: $(0.8 \leq s \leq 0.9, -6^\circ \leq \theta \leq 6^\circ, 0 \leq t_x, t_y \leq 100)$ 内搜索。边缘配准图见图 2(c), 配准结果见图 2(d)。从配准结果图 2(d)中可以看到, 配准结果很好。

对比 Rucklidge 的快速搜索算法, 两个配准实验的配准时间如表 1 所示:

从表 1 可以看出, 改进方法加快了配准速度, 缩短了配准时间。

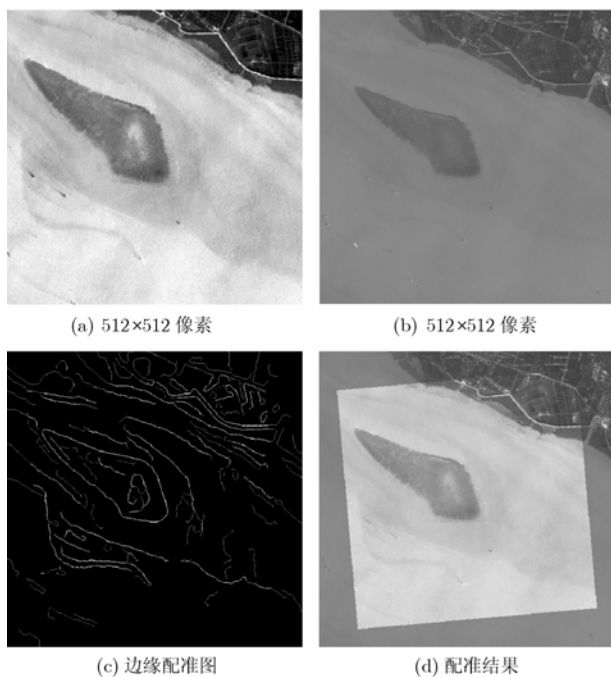


图 2 实验 2 图像配准结果

表 1 两种方法计算时间比较

	实验 1		实验 2	
	Rucklidge 算法	本文算法	Rucklidge 算法	本文算法
配准时间 (s)	9.7324e+001	5.041e+001	1.934e+003	1.438e+003

6 结束语

在 Rucklidge 算法的基础上, 推导出了图像配准中常用的 RST 变换的盒距离变换公式, 并根据算法的需要, 简化了计算 Voronoi 表面的方法。以及仅使用较长的边缘进行配准。从而较大地提高了图像配准的计算速度。如选用 VC++ 语言并引入并行计算来实现本算法, 计算速度会有进一步的提高。再者, 如果使用特征点代替边缘点来计算, 则可以使配准速度更大幅度地提高。所以, 如何提取到可靠、正确率高的特征点是今后需要研究的方向。虽然如此, 如果初始参数的搜索空间设置的太大, 配准的过程还是极为耗时的。所以搜索策略还需要进一步的改进。

参考文献

- [1] Brown L G. A survey of image registration techniques[J]. *ACM Computing Surveys*, 1992, 24(4): 325-376.
- [2] Fonseca L M G. Registration techniques for multisensor remotely sensed imagery[J]. *Photogrammetric Engineering & Remote Sensing*, 1996, 62(9): 1049-1056.
- [3] Xia Minghui and Be de. Image registration by "super-curves"[J]. *IEEE Trans. on Image Processing*, 2004, 13(5): 720-732.
- [4] Huttenlocher D P, Klanderman G A, and Rucklidge W J. Comparing images using the Hausdorff distance[J]. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1993, 15(9): 850-863.
- [5] Rucklidge W J. Efficiently locating objects using the Hausdorff distance[J]. *International Journal of Computer Vision*, 1997, 24(3): 251-270.

牛力丕: 男, 1973 年生, 博士生, 研究方向为多传感器图像处理、实时信号处理、雷达信号处理、DSP 系统设计和实现。

毛士艺: 男, 1935 年生, 教授, 博士生导师, 中国电子学会信号处理专业副主任委员, 中国电子学会会士。主要研究领域为高分辨雷达成像、信号建模与谱分析、信号检测与分类、多目标跟踪和数据融合等。

陈 炜: 男, 1963 年生, 副教授, 硕士生导师, 主要研究领域为图像融合、数字图像处理、数字信号处理等。