

文章编号: 1672-8785(2012)05-0036-06

利用 HDL Coder 完成红外探测器 图像死点的替换

唐立梅 陈莲娜

(中国计量学院信息工程学院, 浙江杭州 310018)

摘 要: 利用 HDL Coder 工具从基于 simulink 模块的设计中生成可综合的硬件描述语言, 并用 FPGA 实现红外探测器图像数据流的死点替换。以非致冷红外探测器成像系统中的死点替换为例, 阐述了这一开发流程。将开发的关注点从 HDL 代码的底层编写转移到系统构架算法和仿真等更高级别的设计上来, 提高了开发效率, 并且取得了很好的图像处理效果。

关键词: HDL Coder; 红外图像; 死点替换; 非均匀性校正

中图分类号: TP751 **文献标识码:** A **DOI:** 10.3969/j.issn.1672-8785.2012.05.006

Dead Pixel Replacement in Infrared Detector Imaging System Using HDL Coder

TANG Li-mei, CHEN Lian-na

(College of Information Engineering, China Jiliang University, Hangzhou 310018, China)

Abstract: Firstly, the synthesizable hardware description language (HDL) codes are generated in the design based on a simulink module by using the HDL Coder. Then, the FPGA is used to implement the dead pixel replacement in the image data stream of an infrared detector. By taking the dead pixel replacement of an uncooled infrared imaging system for example, the development process is described. Because the concerns are transferred to the upper design levels such as system architects and simulation, the development efficiency is improved and good image processing effectiveness is obtained.

Key words: HDL Coder; infrared image; dead pixel replacement; non-uniformity correction

0 引言

非致冷红外探测器成像系统是目前国家重点建设的科技项目, 其目标是使核心成像部件红外探测器实现国内自主量产。从国内外的应用情况来看, 红外成像系统不仅在国防军事上具有非常重要的应用, 而且其民用市场也正在迅速增长。

红外探测器的成像技术和图像处理技术都是成像系统中的关键技术。为了满足红外图像实

时处理的要求, 通常需要采用硬件实现算法。其中, 基于 FPGA 是硬件实现的一种方式。HDL Coder 是 MATLAB 中附带的一种工具, 它可以将 simulink 模块或者 stateflow 有限状态机转换成 HDL 语言, 而且生成的 HDL 语言是可综合的, 即能够直接将这些生成的 HDL 代码放在 FPGA 芯片中实现^[1]。有了该工具, 设计人员便可以将精力集中于系统级和模块级的构建, 而不必再纠缠于 HDL 代码编写这一级。本文的立意是

收稿日期: 2012-03-04

作者简介: 唐立梅 (1970-), 女, 浙江杭州人, 讲师, 主要研究方向为图像和视频处理。E-mail: tanglm@cjluc.edu.cn

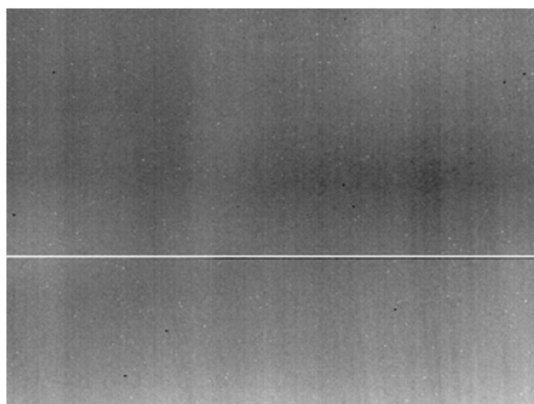
利用 HDL Coder 工具所提供的便利功能, 高效地进行应用系统开发, 以解决红外图像成像中的关键技术问题。

1 红外图像死点的替换算法

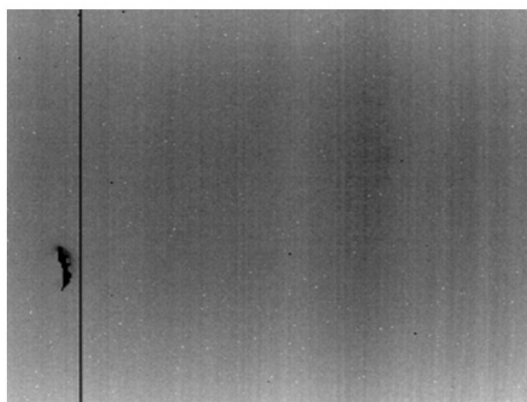
1.1 红外图像的死点情况

目前, 在制造红外探测器时, 在电极形成、材料生长、沉积、刻蚀、划片、释放和清洗等工艺过程中不可避免地会使一些像元失效 (失效的像元就是死点), 或者是形成一些不活跃的像元点。与正常的像元点相比, 不活跃的像元点的热敏感特性较差, 敏感范围窄。在探测一定温度

范围内的目标时, 其表现与正常像元的表现相差不大; 但在温度范围以外, 其表现可呈现出死点的特征。死点数量的多少和工艺的控制情况有关。死点的分布没有固定的规律, 它们大体上有均匀分布的, 有单个或者几个并在一起的, 有成线形的, 也有成簇的, 如图 1 所示。图 1(a) 中, 单个黑点和白点均有可能为失效点, 而白色横线即为整行失效点, 即死点; 其他的点均呈现出非均匀性。图 1(b) 同样具有明显的非均匀性, 其中单个黑点和白点均有可能为失效点, 黑色竖线为死点, 同时也有黑色成簇的死点。



(a)



(b)

图 1 红外图像的死点分布情况

在提到死点的时候, 同时也需要提到像元的非均匀性^[2]。同样是因为工艺的问题, 人们不可能将红外探测器整个像元面阵上所有像元点的特性做到一致。此时, 即使没有产生死点, 也依然会存在这样一个现象: 所有像元点在接收同一照度的红外能量的情况下, 每个像元点的输出特性是不一样的; 在红外图像的处理上, 往往需要对这种情况进行校正。假设目标温度处在一定范围内且每个像元点的特性为线性的前提下, 可对每个像元点的输出进行补偿, 人为地使所有像元点的最后输出在相同输入下达到一致 (称为非均匀性校正)。最常用的非均匀性校正方法是一点校正法和两点校正法。其原理如下: 在给定温度分别为 T_1 和 T_2 的黑体辐射充满探测器视场的情况下, 分别测得 $M \times N$ 像元面阵

中所有像元的输出电平, 然后通过计算得到两个参数矩阵 G_i 和 O_i (此参数即为像元矩阵中任意一点 P_i 的修正参数)。令 P_i 校正前的输出电平为 V_i , 非均匀性校正后的输出为 $V_{icorrect}$:

$$V_{icorrect} = V_i * G_i + O_i \quad (1)$$

其中, 一点校正法可用于修正 G_i 值, 两点校正法可用于得到 G_i 和 O_i 。在实际应用中, G_i 一般无需频繁修正, 而由于像元本身的不稳定性, 却往往需要用一点校正法来修正 O_i 值。

1.2 死点的相邻点替换算法

常用的相邻点替换方法包括前点替换法、 3×3 窗口法和 5×5 窗口法等^[3]。假设当前点为死点, 则取该点相邻点的输出作为当前点的输出, 以达到欺骗人眼的目的, 从而造成该死点被

消除的假象。前点替换只是一种简单的处理方法,其效果只是在死点呈单个死点分布的情况下才会生效。若横向有 2 个或者 2 个以上的死点,则不能完全消除死点,因此在实际应用中并不会采用该方法。

3×3 窗口法和 5×5 窗口法的原理是一样的。在死点簇超过 3×3 或 5×5 的情况下,这两种方法也不能完全奏效(可扩大窗口,如 7×7 等)。不过,在商用探测器上很难看到有大于 5×5 的死点簇,因此讨论 5×5 窗口是具有实用价值的。图 2 所示为 5×5 窗口。

5×5 窗口校正的基本方法是取死点相邻的 5×5 窗口中的 24 个像元点的输出,首先对其进

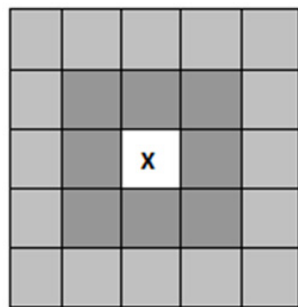


图 2 5×5 窗口

31				27	26											16	15													0
----	--	--	--	----	----	--	--	--	--	--	--	--	--	--	--	----	----	--	--	--	--	--	--	--	--	--	--	--	--	---

图 3 数据索引值和校正参数排列

2 替换算法的 FPGA 实现

下面将从两个数据到 5×5 窗口数据,讨论通过 HDL Coder 生成可综合 HDL 语言来完成死点替换的 FPGA 实现。

通常情况下,设计者可以在 simulink 里编写待验证模块的测试模块,也可以生成 testbench 而从 modelsim 得到仿真。当仿真通过后,可以利用 HDL Coder 工具检查 HDL 的兼容性,并最终得到可综合的 HDL 语言。但是,并不是所有的 simulink 模块都可以被 HDL Coder 利用的。如果在设计完成后再来检查 HDL 的兼容性,那么一旦有使用的模块不能用于 HDL 语言的生成,这时就需要重新去设计原系统,所以,通常在开始

行排序,然后或判断中值,或取排序后指定的序列值,或加权取值等。

当所选的这 24 个值的某个点不是死点的时候,可将其作为处于窗口中心位置的死点的输出值。排序索引值的选择可根据情况而定。有取中值的索引的,也有随机定一个索引值的,只要拿来用于替换的那个点不是死点即可。

一般情况下,可将欲替换的点的排序索引值和非均匀性校正的参数都保存在一起。红外热像仪采用 12 位、14 位或 14 位以上的 A/D 转换芯片对探测器输出信号进行采样。一般每个像元点的值用 16 位表示。在非均匀性参数计算中往往也可以用 16 位来表示 G_i 和 O_i 。不过, G_i 用不了 16 位。经综合考虑后,用 5 位表示替换排序索引值,用 11 位表示 G_i 值,用 16 位表示 O_i 值(组合成 32 位,可暂存在 16 位外部 SRAM 或 32 位外部 SRAM 中,以提高处理速度)。这在实际的热像仪处理系统中是合理的。图 3 所示为数据索引值和校正参数排列。其中, 31 ~ 27 为替换的排序索引值, 26 ~ 16 为 G_i 值, 15 ~ 0 为 O_i 值。

设计前,需要事先利用 hdlLib 命令生成 HDL 兼容的一个库(库里包含了 simulink 中所有 HDL 兼容的模块),然后调用这个库里的单元模块所构建的系统模块。这样做是完全可以实现 HDL 兼容的。

HDL Coder 可以采用 GUI 方式,也可以采用命令行方式,其应用对象是 simulink 模块。在 5×5 窗口模式下,因为探测器的输出是逐点输出,所以需要缓存 5 行数据。以 320×240 面阵大小的探测器为例,它需要缓存 5×320 个点的数据。若一个点以 16 位表示(即 2 个字节),则一共需要缓存 3200 个字节。

根据排序索引值的选择, 5×5 窗口法的最基本要点是排序,而排序的最小单元为 2 个值

比较大小。因此, 以 2 个值比较大小为例 (见图 4), 演示利用 HDL Coder 来生成 HDL 语言。

在 simulink 中搭建一个名为 sub_sort_2 的模块。在该模块输入端任意输入 2 个值, 即可按大小排序。HDL Coder 可以生成 VHDL 语言, 也可以生成 Verilog HDL 语言。现设置 HDL Coder 生成 Verilog HDL 语言代码。利用 2 个数据的交换模块, 即可构建出 5 个输入数据的排序模块, 进而可构建出 25 个输入数据的排序模块。图 5 所示为 5 个输入数据的排序模块。

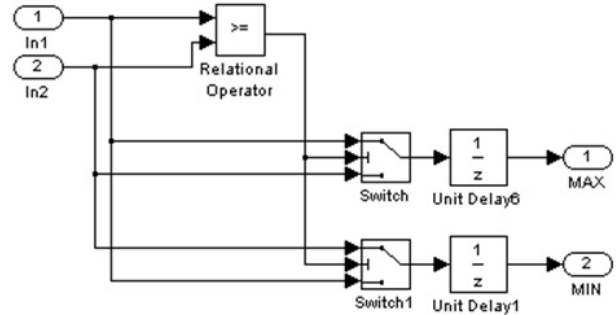


图 4 两个值比较大小模块

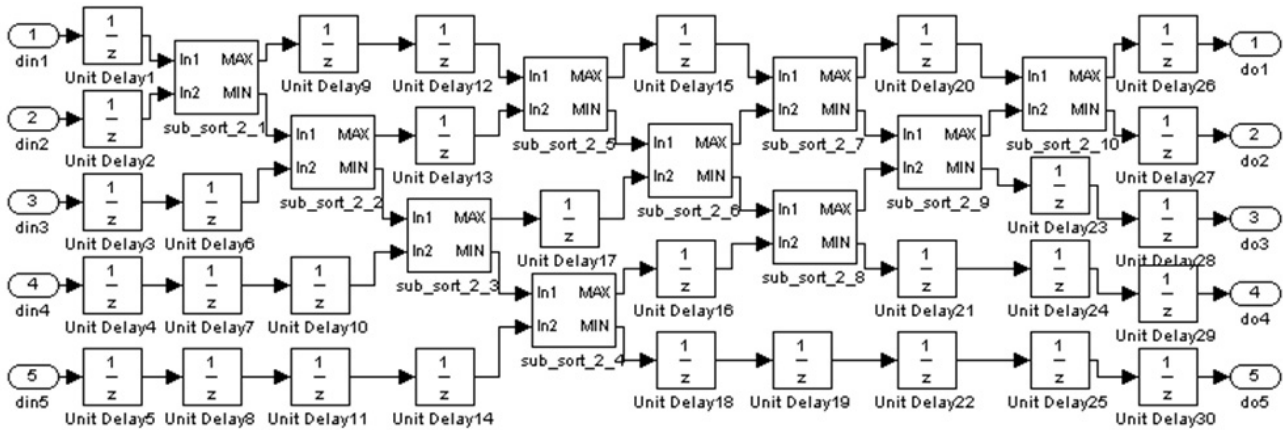


图 5 5 个输入数据的排序模块

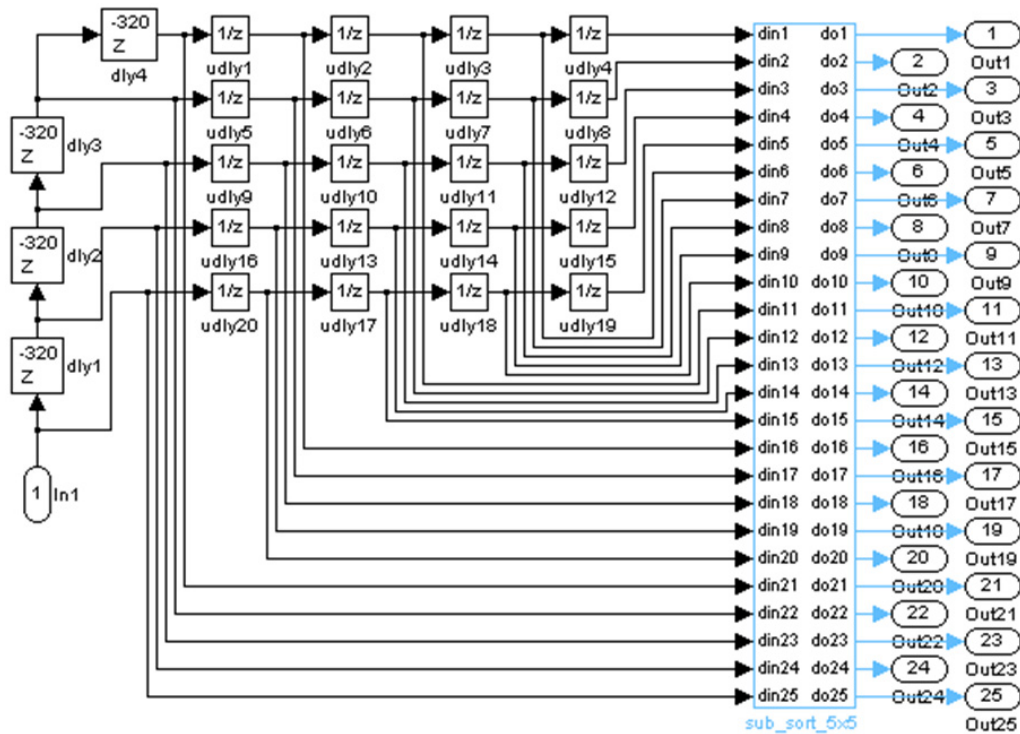


图 6 5×5 窗口排序的模块

在 simulink 中搭建 5×5 窗口排序模块 (主要是利用 swap_value 和缓存模块)。假设红外探测器为 320×240 面阵, 因此需要缓存 5 行数据。用 5×5 窗口滑动, 对该窗口的 25 个数据进行排序。如果窗口的中心点为死点, 那么就可以用这 25 个数据中的一个代替死点来作为输出值。至于用哪个值, 可以通过再次优化来定。将排序的

序列号与前面提到的非均匀性校正参数组合在一起, 然后将其保存起来并应用到成像系统中。图 6 所示为 5×5 窗口排序的模块。

3 算法结果

3.1 模块仿真结果

图 7 所示为 5 个输入数据的模块的仿真结

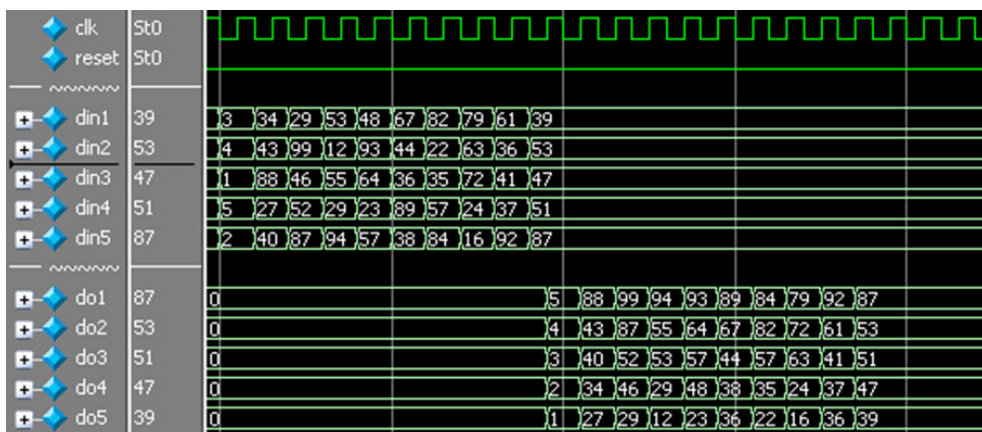


图 7 5 个输入数据排序的仿真结果

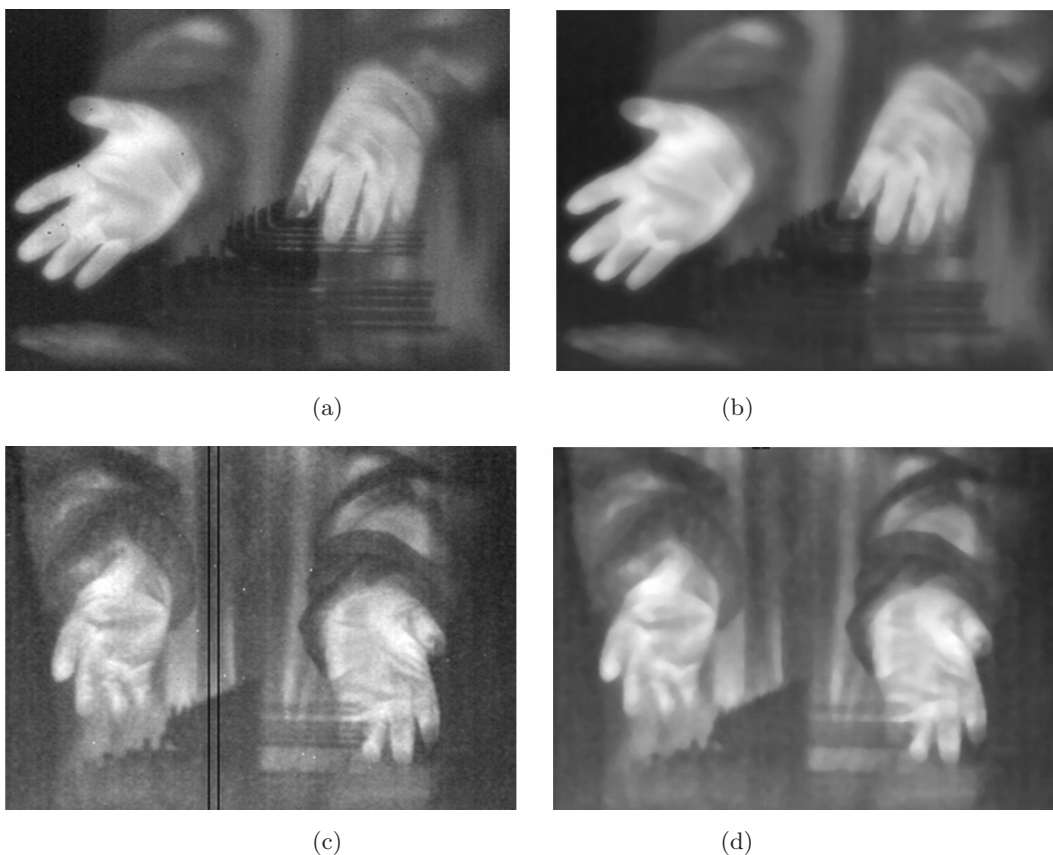


图 8 图像处理结果

果。从图 7 中可以看出, 该模块的输出比输入延迟了 10 个 clk。也就是说, 对于前端输入, 经流水处理和整体延迟 10 个 clk 后, 便可即时输出排序结果。假设系统时钟为 100 MHz, 那么 10 个 clk 的延迟时间为 100 ns。

同理, 可构建 25 个输入的排序模块。不过, 其延时 clk 需要更多, 按同样的方法设计, 可达 60 个 clk 或者更多 (视添加的单元 clk 延时个数而定)。同样在 100 MHz 的主时钟下, 60 个 clk 的延迟时间为 600 ns。FPGA 流水节拍处理时为实时处理。

3.2 图像处理效果

利用 HDL Coder 将该 5×5 窗口排序模块转换成可综合的 Verilog 语言, 并将其应用在一台实际的非致冷红外探测器成像系统的死点替换中。在本文采用的系统里, 采用 Out13 来替换当前点的死点 (其效果见图 8)。在一个成熟的成像系统中, 可以灵活采用 25 个序列中的一个, 或者采用更为优化的方法, 如对于不同的死点簇, 视簇的大小而采用不同的序列值 (并非单一指定固定的某个序列号), 同样可以达到良好的视觉效果。图 8(a) 为仅采用单点校正后得到的图像 (可以看到死点位置和分布)。图 8(b) 为采用死点替换和非均匀性校正后得到的图像。图 8(c) 中的黑色竖线为死点, 其校正结果见图 8(d)。从图 8(b) 和图 8(d) 中已经看不出明显的死点。

4 结论

本文例中, 在 100 MHz 主时钟下, 原始输入信号经模块处理后, 其输出信号比输入信号延

迟 60 个主时钟 (600 ns) 即可获得实时输出的图像。所以完成系统实时性的要求是有很大的富余量的, 这为完成较复杂的算法提供了很大的空间。本文利用 HDL Coder 将模块转换到 HDL 语言, 实现了红外探测器死点替换的 FPGA 算法, 并取得了很好的效果。通过用 HDL Coder 将在 simulink 里创建的 mdl 模块转换成 FPGA 芯片可综合的 HDL 语言, 设计者的工作可以从 HDL 语言的具体代码编写转到系统实现和仿真的层次上来, 从而可以将更多的精力放在系统级别的关注上。此外, 设计者还可以利用 simulink 现有的许多高效而方便的模块, 使 FPGA 在实现复杂的算法方面变得更加容易和高效。充分利用工具的资源进行简便高效的优化开发, 在实际应用中是非常必要的。

参考文献

- [1] Gavin L. Bates & Saeid Nooshabadi, FPGA Implementation of a Median Filter [C]. London: IEEE TENCON-Speech and Image Technologies for Computing and Telecommunications, 1997.
- [2] Sergio N, Majeed M H. Kalman Filtering for Adaptive Nonuniformity Correction in Infrared Focal-plane-arrays [J]. Journal of Optical Society of America, 2003, 20(3): 126-131.
- [3] Helmut Budzier, Gerald Gerlach. Thermal Infrared Sensors: Theory, Optimisation, and Practice [M]. London: Chichester West Sussex, 2011.

新闻动态 News

美国 Northrop Grumman 公司演示 水面舰载威胁跟踪系统

据 www.militaryaerospace.com 网站报道, 美国 Northrop Grumman 公司的一个研究小组最近演示了一种使水面舰船能够探测和跟踪外部威胁的静态注视电光红外 (EO/IR) 分布式孔径系统 (DAS)。

Northrop Grumman 公司最初设计和研制这种静态注视电光红外分布式孔径系统的目的是为了给 F-35 战斗机的飞行员提供 360° 态势感知能力。这种威胁告警系统不久将会为有人 / 无人水面舰船或潜艇平台提供态势感知功能。

该系统能够自动跟踪水面和空中目标, 并采用多个传感器来提供 360° 全天候图像。它能够自动探测和跟踪水面舰船、飞机以及诸如弹道导弹和巡航导弹之类的潜在威胁。

□ 岳桢千