Approximation of Large Scattered Irregular Spatio-temporal Data and Military Applications

Vaclav Skala[1,2] Václav Křivánek[1]
www.VaclavSkala.eu

[1]Dept. of Military Robotics, Faculty of Military Technology
University of Defence, Brno
[2]Dept. of Computer Science and Engineering, Faculty of Applied Sciences
University of West Bohemia, Pilsen
Czech Republic

October 20, 2026

Outline

1. Introduction
2. RBF Meshless Methods - Principle
3. RBF Interpolation
4. Spatio-Temporal RBF
5. Military Applications
6. Summary
7. Conclusion
8. Questions

Introduction

Data collected in modern military operations are increasingly distributed in space and time. Examples include sensor arrays deployed over a battlefield, radar cross-section measurements, atmospheric sampling for UAV navigation, and underwater acoustic sensing.

However, such data are rarely structured: sensors move, fail, or report asynchronously.

Classical grid-based interpolation methods (finite differences, splines, or kriging) require regular grids or specific correlation models. In contrast, *Radial Basis Function* (RBF) methods operate directly on scattered data, requiring only pairwise distances.

This makes the RBF ideal for **spatial–temporal fusion of irregular samples**.

Introduction

Typical examples:

- **UAV Sensor Fusion**: Consider a surveillance scenario with UAVs sampling temperature or radiation intensity at scattered positions. The goal is to reconstruct the field $f(\mathbf{x})$ over the area for situational awareness.
- **Gap Filling:** Estimating values in regions where no measurements were taken (e.g., due to sensor failure, terrain obstruction, or enemy jamming).
- **Temporal Forecasting with RBFs**: For spatio–temporal tracking (e.g., chemical plume propagation or moving targets), the time dimension can be used for short-term prediction by extrapolating coefficients or fitting dynamic models on the reconstructed field.

Introduction

Applied Military Examples:

- **Battlefield Sensor Fusion**: Deployed ground sensors measure chemical concentrations. Due to communication limits, data arrive sporadically. RBF interpolation reconstructs the continuous concentration field in real time, aiding decision making.
- **Radar Cross-Section Mapping**: RBFs can interpolate measured radar reflections over frequency and angle. This provides smooth RCS maps for target identification and stealth evaluation.
- **UAV Swarm Coordination**: Each drone estimates wind speed at its position. RBF interpolation of these data produces a continuous vector field, enabling distributed path planning and collision avoidance.
- **Ocean and Atmospheric Models**: Underwater sensors record salinity, pressure, or acoustic intensity at irregular depths. RBF-based spatio-temporal interpolation reconstructs continuous environmental maps for navigation and communication systems.

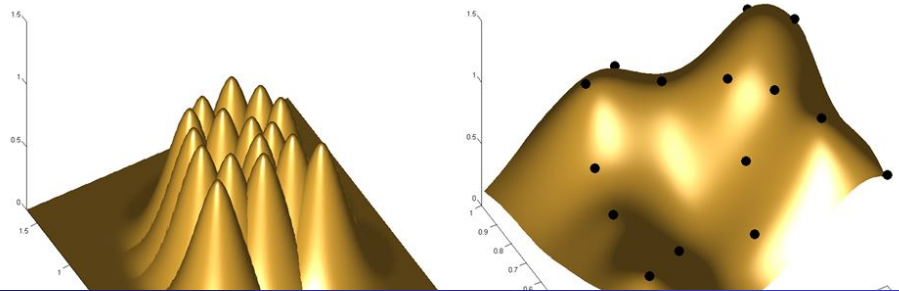
RBF Interpolation Principle

The key idea is to approximate a function $f(\mathbf{x})$ as a linear combination of basis kernel functions:

$$f(\mathbf{x}) = \sum_{j=1}^{N} c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) = \sum_{j=1}^{N} c_j \phi_j(\mathbf{x}) \tag{1}$$

where: $\phi_j(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{x}_j\|)$ is the kernel RBF centered at $\mathbf{x}_j$, the coefficients c_j are to be determined and $\| * \|$ is a "distance" of the points $\mathbf{x}$ and $\mathbf{x}_j$.

RBFs are generally independent of the data domain dimension, and lead to a system of linear equations $\mathbf{Ax} = \mathbf{b}$.



RBF Kernel Functions

Kernel interpolation functions $\phi(r)$ are generally of two types:

- "**global**" functions with unlimited influence over the data domain,
  e.g. $\phi(r) = \frac{1}{2} r^2 \log r^2$, $\phi(r) = e^{-\alpha r^2}$, $\phi(r) = \sqrt{1 + \alpha r^2}$, $\phi(r) = 1/\sqrt{1 + \alpha r^2}$,
  The matrix **A** is usually "full" and might be ill-conditioned,
- "**local**" functions with a limited influence only for $r \in [0, 1]$, mostly in the form
  $\phi(r) = (1 - r)^q P_k(r)$, e.g. $(1 - r)^5 (8r^2 + 5r + 1)_+$ "$_+$" means limitation to the interval $[0, 1]$. A shape parameter $\alpha > 0$ is a multiplicative factor to r used to extend/shrink the function influence to $[0, \alpha]$.
  The matrix **A** is usually "sparse" (depends on α).
  or
- "**combined**": e.g. Buhmann's $\phi(r) = (\frac{1}{3} + r^2 - \frac{4}{3} r^3 + 2r^2 \log r)_+$ etc. or
  Menandro's functions: Two new classes of compactly supported radial basis functions for approximation of discrete and continuous data,
  DOI: 10.1002/eng2.12028

RBF Kernel Functions

Table: Common Types of Radial Basis Functions

Category	**Name**	**Definition** $\phi(r)$	**Parameters**
Global	Gaussian (GA)	$\exp(-(\epsilon r)^2)$	$\epsilon > 0$ (shape)
	Multiquadric (MQ)	$\sqrt{1+(\epsilon r)^2}$	$\epsilon > 0$
	Inverse Multiquadric (IMQ)	$1/\sqrt{1+(\epsilon r)^2}$	$\epsilon > 0$
	Thin Plate Spline (TPS)	$r^2 \log(r)$	(none)
Compact	Wendland C^2	$(1-\epsilon r)_+^4(4\epsilon r+1)$	$\epsilon > 0$, $r \leq 1/\epsilon$

Note: The subscript '+' denotes the cutoff function: $(x)_+ = x$ if $x \geq 0$, and 0 otherwise.

Scalar value interpolation

If scalar values $h_i = h(\mathbf{x}_i)$ are given in mutually distinct scattered points $\mathbf{x}_i$, $i = 1, \ldots, N$ a system of linear equations is obtained:

$$h(\mathbf{x}_i) = \sum_{j=1}^{N} c_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) = \sum_{j=1}^{N} c_j \phi_j(\mathbf{x}_i) = \sum_{j=1}^{N} c_j \phi_{ij} \quad , \quad i = 1, \ldots, N \tag{2}$$

where: $\phi(\|\mathbf{x} - \mathbf{x}_j\|) = \phi_j(\mathbf{x})$ is the kernel RBF centered at $\mathbf{x}_j$, c_j are coefficients to be determined, $\mathbf{x}_i$ are the interpolation points, $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the Euclidean distance, $\phi_j(\mathbf{x}_i) = \phi_{ij}$ is the value of the function ϕ_j at the point $\mathbf{x}_i$.
This formulation leads to a system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, i.e.

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N1} & \phi_{N2} & \cdots & \phi_{NN} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix} \tag{3}$$

where the matrix $\mathbf{A}$ is symmetrical, i.e. $\mathbf{A} = \mathbf{A}^T$.

Scalar value interpolation

To obtain better numerical robustness and matrix positivity, a polynomial $P_k(\mathbf{x})$ of the degree k is usually added together with additional orthogonal conditions, e.g., for the degree $k = 2$ the polynomial can have a form $P_2(\mathbf{x}) = a_0 + a_1 x + a_2 y + a_3 xy$. Then, the interpolation is given as:

$$h(\mathbf{x}_i) = \sum_{j=1}^{N} c_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) + P_k(\mathbf{x}_i) \ , \ i = 1, \ldots, N$$

$$P_k(\mathbf{x}) = a_0 + a_1 x + a_2 y + a_3 xy \tag{4}$$

$$\sum_{j=1}^{N} a_0 = 0 \ , \ \sum_{j=1}^{N} a_1 x_j = 0 \ , \ \sum_{j=1}^{N} a_2 y_j = 0 \ , \ \sum_{j=1}^{N} a_3 x_j y_j = 0$$

It should be noted, that the size of the matrix $\mathbf{A}$ is independent of the data domain dimension.

Scalar value interpolation

This leads to the linear system of equations:

$$\left[\begin{array}{cccc|cccc} \phi_{11} & \phi_{12} & \dots & \phi_{1N} & 1 & x_1 & y_1 & x_1y_1 \\ \phi_{21} & \phi_{22} & \dots & \phi_{2N} & 1 & x_2 & y_2 & x_2y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{N1} & \phi_{N2} & \dots & \phi_{NN} & 1 & x_N & y_N & x_Ny_N \\ \hline 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 \\ x_1 & x_2 & \dots & x_N & 0 & 0 & 0 & 0 \\ y_1 & y_2 & \dots & y_N & 0 & 0 & 0 & 0 \\ x_1y_1 & x_2y_2 & \dots & x_Ny_N & 0 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_N \\ \hline a_0 \\ a_1 \\ a_2 \\ a_3 \end{array}\right] = \left[\begin{array}{c} h_1 \\ h_2 \\ \vdots \\ h_N \\ \hline 0 \\ 0 \\ 0 \\ 0 \end{array}\right] \tag{5}$$

It can be rewritten as:

$$\begin{bmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{h} \\ 0 \end{bmatrix} \tag{6}$$

where $\mathbf{a} = [a_0, \dots, a_3]^T$, $\mathbf{c} = [c_1, \dots, c_N]^T$, $\mathbf{h} = [h_1, \dots, h_N]^T$ and $h_i = f(\mathbf{x}_i)$, $i = 1, \dots, N$. Generally, the polynomial $P_k(\mathbf{x})$ can be seen as a "global approximation" and RBF as a "local adjustment of values". It also leads to better numerical robustness and interpolation precision.

Scalar value interpolation CSRBF

It can be seen that the matrix in Eq.6 can be quite large and dense if global RBFs are used. However, if Compactly Supported RBFs (CS-RBF) are used, and the shape parameter α is adequately chosen the matrix of the linear system is sparse. It leads to faster computation of large data sets, and data domain subdivisions can be used.

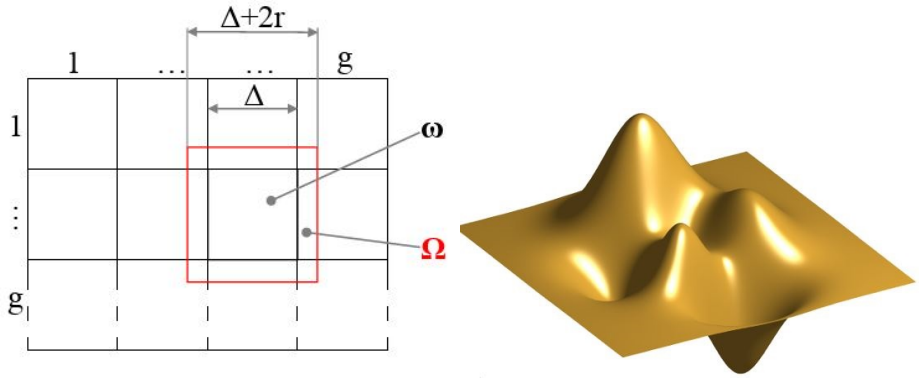


Figure: Data domain subdivision and blended interpolation [**Smolik201749**]

Scalar value interpolation

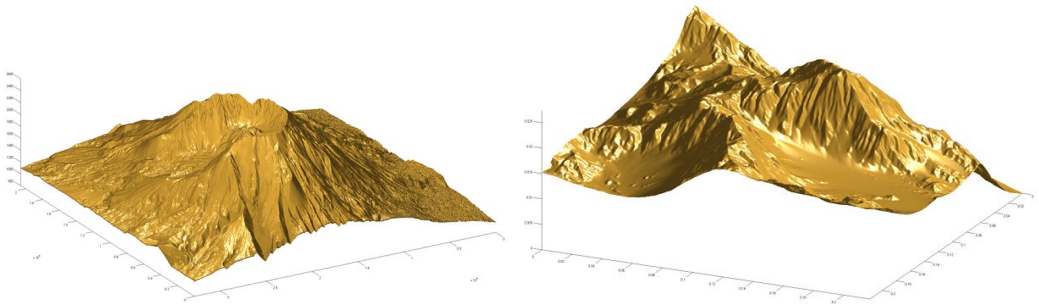


Figure: Mount St. Helens; 756 150 points Alps mountains; 131 044 points [**Smolik201749**]

RBF interpolation on larger datasets using Compactly Supported RBF(CS-RBF) and domain space subdivision was described in:

- **Smolik201749** [**Smolik201749**], IF=5.8
- **Skala20172433** [**Skala20172433**]

It leads to significant computational speedup, lower memory demands, and better numerical stability. In the case of the St.Helens, the RFB coefficients computation speedup was $\nu \approx 2.4\ 10^5$, and RBF evaluation speedup was $\nu \approx 172$. The existence of *individual* optimal shape parameters for each CS-RBF kernel function;

Spatio-Temporal RBF

The extension for Spatio-temporal (space-time) **irregularly scattered data in time** and in space is as simple as follows:

$$\left[\begin{array}{cccc|cccc} \phi_{11} & \phi_{12} & \dots & \phi_{1N} & 1 & x_1 & y_1 & t_1 \\ \phi_{21} & \phi_{22} & \dots & \phi_{2N} & 1 & x_2 & y_2 & t_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{N1} & \phi_{N2} & \dots & \phi_{NN} & 1 & x_N & y_N & t_N \\ \hline 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 \\ x_1 & x_2 & \dots & x_N & 0 & 0 & 0 & 0 \\ y_1 & y_2 & \dots & y_N & 0 & 0 & 0 & 0 \\ t_1 & t_2 & \dots & t_N & 0 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_N \\ \hline a_0 \\ a_1 \\ a_2 \\ a_3 \end{array}\right] = \left[\begin{array}{c} h_1 \\ h_2 \\ \vdots \\ h_N \\ \hline 0 \\ 0 \\ 0 \\ 0 \end{array}\right] \tag{7}$$

leading again to:

$$\begin{bmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{h} \\ 0 \end{bmatrix} \tag{8}$$

Spatio-Temporal RBF

A vital military consideration is dynamically processing asynchronous sensor data acquired at unpredictable intervals. For the Gaussian kernel, the RBF interpolation formula becomes:

$$f(\mathbf{z}) = f(\mathbf{x}, t) = \sum_{j=1}^{N} c_j e^{-\alpha^2 r^2}, \tag{9}$$

where $r^2 = \|\mathbf{x} - \mathbf{x}_j\|^2 + \gamma^2(t - t_j)^2 = \|\mathbf{x}(t) - \mathbf{x}_j(t_j)\|^2 + \gamma^2(t - t_j)^2$.
Note that the position $\mathbf{x} = \mathbf{x}(t)$ is not constant, generally.

Spatio-Temporal RBF

In the Gauss kernel function[1], case the space and time parts can be separated and the formula Eq.9 can be rewritten as:

$$f(\mathbf{x},t) = \sum_{j=1}^{N} c_j e^{-\alpha^2\left(\|\mathbf{x}-\mathbf{x}_j\|^2+\gamma^2(t-t_j)^2\right)} = \sum_{j=1}^{N} c_j e^{-\alpha^2\|\mathbf{x}-\mathbf{x}_j\|^2} * e^{-\alpha^2\gamma^2(t-t_j)^2} = \sum_{j=1}^{N} c_j e^{-\alpha^2\|\mathbf{x}-\mathbf{x}_j\|^2} * e^{-\beta^2(t-t_j)^2} \tag{10}$$

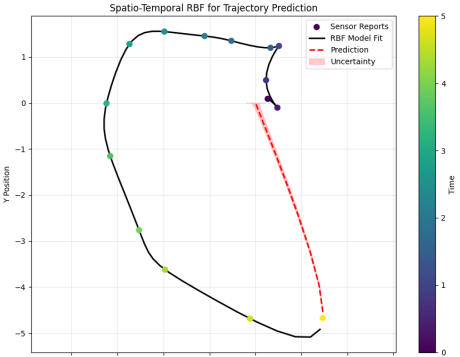
where $\beta^2 = \alpha^2\gamma^2$.
This means that there are actually two shape parameters, α for spatial coordinates and β for the time coordinate[2]

Military Applications

Terrain Reconstruction and Elevation Modeling

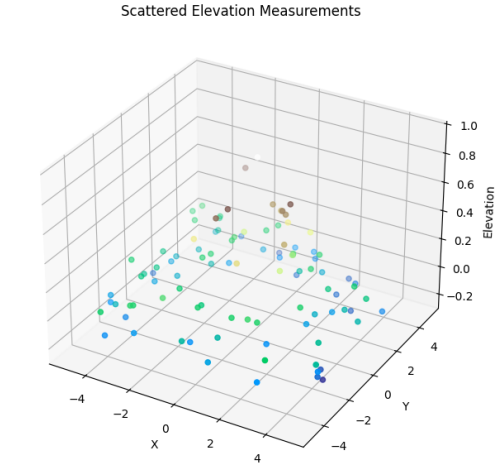
UAVs and reconnaissance patrols often collect scattered elevation points. RBFs can create a smooth, continuous Digital Elevation Model (DEM):

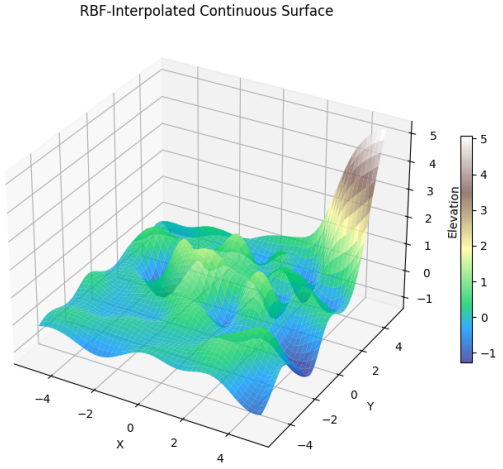
- Line-of-sight analysis for sniper placement or communication relay siting.
- Path planning for autonomous ground vehicles.
- Mission planning and simulation.



Military Applications

The TPS RBF is particularly well-suited for this task as it minimizes the bending energy of the surface, creating a physically plausible "smoothest" surface.





Military Applications

Sensor Fusion for Comprehensive Air Picture

Multiple radars (e.g., long-range early warning and short-range point defence) provide overlapping but incomplete tracks on multiple targets. The data is noisy, asynchronous, and may contain false tracks. RBFs can be used to:

- **Fuse Tracks:** Create a single, coherent, and accurate estimate of each target's position and velocity.
- **De-ghosting:** Resolve ambiguities that arise when multiple radars see the same target from different angles, potentially creating "ghost" targets.

A spatio-temporal RBF model can assimilate all position reports in space and time, creating a continuous track for each target that is more accurate and reliable than any single sensor's output.

Military Applications

Prediction of Threat Trajectories As shown conceptually, RBFs are excellent for predicting the future path of dynamic threats.

By modeling the target's history as a spatio-temporal curve, the RBF interpolant can be extrapolated to provide a probabilistic forecast. This is vital for:

- **Missile Defence:** Predicting the impact point of an incoming ballistic missile.
- **Counter-UAS:** Forecasting the flight path of a hostile drone to enable effective interdiction.

Summary - Main advantages

- **Simplicity**: The formulation leads to a simple formula relying on a solution of linear systems of equations, which is heavily supported by numerical libraries with efficient solution methods.
- **No meshing (tessellation) required**: The method relies only on the mutual positions of scattered points in the domain, making it particularly useful for irregular domains or problems with moving boundaries.
- **Flexibility**: RBFs can be chosen based on the problem, with typical choices including Gaussian, multi-quadric, thin-plate splines (TPS), or Wenland's functions.
- **Smoothness**: RBF interpolation typically results in a smooth and accurate approximation; however, the interpolation error might be higher on the data domain border.
- **Independence on data domain dimensionality**: RBF use leads to a system of linear equations $\mathbf{Ax} = \mathbf{b}$, where the matrix $\mathbf{A}$ is of size $N \times N$, but is independent of the dimensionality of the data domain.
- **Speed-up**: If Compactly Supported RBFs (CS-RBFs) are used, the matrix $\mathbf{A}$ can be sparse and specific methods can be used.
- **Space subdivision**: If the CS-RBFs are used, space subdivision of the data domain can be used to significantly speed up the solution and the function $f(\mathbf{x})$ evaluation.

Summary - Main disadvantages

- **Numerical problems**: The system of linear equations can be very large for large data sets.
  In [Majdisova-2016-SIGRAD] over 6.7 mil. of points have been interpolated, i.e., the RBF matrix was over the size $6.7\ 10^6 \times 6.7\ 10^6$ was processed.
  Majdisova-2016-SIGRAD [Majdisova-2016-SIGRAD]
- **Numerical robustness**: The matrix **A** tends to be very ill-conditioned, primarily when "global" kernel functions are used.
- **Time consuming**: Evaluation of the interpolated value $h(\mathbf{x})$ can be unacceptable for high N and if the matrix **A** is dense.
- **Parameter value**: Some kernel RBFs are sensitive to choosing a "shape" parameter α value with influence on RBF matrix conditionality.

It should be noted that the RBF, which is used with a polynomial, cannot be directly used for approximation using the least square errors method.

- Majdisova201751 [Majdisova201751]

Applications

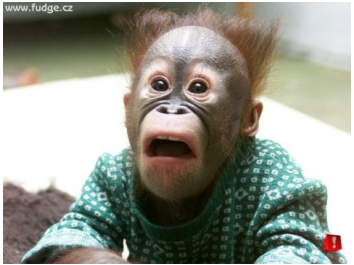
RBF-based meshless methods are widely used in fields such as interpolation, approximation, computational fluid dynamics, structural mechanics, image processing, GIS systems, and data fitting, especially when mesh generation is challenging or computationally expensive and/or higher dimension of the data domain.

Conclusions

- RBF interpolation and approximation provide an elegant and flexible method for reconstructing smooth fields from irregularly distributed spatial–temporal data.
- Mesh-free nature, local adaptability, and robustness to missing data make them highly suitable for defence and security systems.
- Applications range from UAV mapping and radar analysis, radiation level estimation, submarine position detection to environmental reconstruction and real-time sensor fusion, etc.
- Sensors report radiation and chemical agent concentration at various (x(t), y(t)) locations and times. Using this approach, we can reconstruct the continuous 4D concentration field $C(x, y, z, t)$ and predict its future dispersion.
- Trajectory Prediction: Forecasting the future position of aircraft, missiles, or ground vehicles.
- Threat Anticipation: Predicting the likely path of a spreading forest fire or flood towards a base of operations.

Thanks for your attention!

Questions?



Dedication This work has been supported by the Czech Republic Ministry of Defence and the University of Defence, Development program of the organization "VAROPS – DZRO Military autonomous and robotic systems" and "AIROPS – DZRO Conduct of airspace operation" and
by the Ministry of Education, Youth and Sports of the Czech Republic, Institutional Support for Long-term Development of Research Organizations, University of West Bohemia in Pilsen.