GLL

Geomatics, Landmanagement and Landscape No. 4 • 2025, 93–109

ISSN 2300-1496 https://doi.org/10.15576/GLL/213750

Research paper

Received: 13.10.2025
Accepted: 29.10.2025
Published: 20.12.2025

Spatial indexing for access to cartographic archives in GIS/AI systems: A case study of the Warsaw Local Coordinate System (W-25/W-75)

Mariusz Zygmunt 0000-0003-0056-5215

Department of Geodesy, University of Agriculture in Krakow
✉ Corresponding author: mariusz.zygmunt@urk.edu.pl

Summary

Historical city plans and rasters of Warsaw are contained within local coordinate systems (Warsaw-25/75), which makes it difficult to address and integrate them quickly into contemporary data processing pipelines. This paper systematises the problem by defining rigorous procedures for section encoding and decoding: 1) transforming planar coordinates (X, Y) → a sheet code compliant with the N/S–O/W grammar (row, column), and 2) mapping a sheet code → the section envelope [X_{min}, Y_{min}, X_{max}, Y_{max}]. The proposed approach has constant time complexity O(1), employs unambiguous boundary rules, and is parameterisable (sheet dimensions, the position of the Warsaw origin), which simplifies adaptation to heterogeneous cartographic series. We demonstrate how these mechanisms plug into practical workflows: they allow one to identify the target sheet immediately from a given point, reconstruct its bounding box before loading data, and verify consistency between a code stored in metadata and a code computed from coordinates. These procedures interoperate with transformations to PUWG 2000 (PL-2000), enabling the seamless linkage of Warsaw materials with current reference databases. In AI/ML scenarios, the section code acts as a stable spatial identifier that facilitates sample selection and dataset organisation. We additionally provide numerical recommendations (ε and floor) that stabilise the classification of edge cases and remove ambiguities. The algorithms were implemented in Bentley MicroStation with Visual Basic for Applications, ensuring portability to other VBA-capable environments such as Microsoft Office. The solution requires no series lookup tables, operates directly on a metric grid, and is robust to coordinate precision variations, thereby supporting ETL automation and quality control across public administration and industry.

Keywords

map-sheet index • spatial indexing • GIS/ETL • archival rasters • AI/ML labelling

Abbreviations and symbols used in the paper

AI/ML	–	Artificial Intelligence / Machine Learning
AOI	–	Area of Interest
BNG	–	British National Grid – British grid reference system used by Ordnance Survey
DGN	–	MicroStation drawing file format (Bentley). Stores the geometry and metadata of a project
ε (epsilon)	–	A small positive numerical constant (epsilon) added to X,Y for stability at the boundaries
ETL	–	Extract–Transform–Load; data processing chain
GCS	–	Geographic Coordinate System – geographic coordinate system assigned to a file/project
GIS	–	Geographic Information System – geographic information system, software, and methods
IMW	–	International Map of the World – historical system of 1:1,000,000 map sheets (Penck)
MDL	–	MicroStation Development Library – native API/SDK for extending MicroStation
MGRS	–	Military Grid Reference System – military reference grid system based on UTM/WGS84
NTS	–	National Topographic System – Canadian system of dividing topographic map sheets
O(1)	–	Constant time computational complexity – the operating time does not depend on the size of the data
PL-2000	–	State rectangular coordinate system (Poland), zonal, based on G-K
Point3d	–	MicroStation data type (VBA/MDL) with X, Y, Z fields (Double). X and Y are used here
W-25, W-75	–	Warsaw local coordinate systems
Quadtree/R-tree	–	Spatial index structures that accelerate search and data operations
ULK	–	Układ Lokalny Krakowski – local coordinate system used in the Krakow resource
USNG	–	United States National Grid – civil equivalent of MGRS for the US (UTM reference grid)
UTM	–	Universal Transverse Mercator – Mercator transverse projection divided into zones
VBA	–	Visual Basic for Applications – macro language; in MicroStation for automation/processing
WMS/WMTS	–	Web Map Service / Web Map Tile Service – mapping service standards (e.g., rasters)
ZGW	–	‘geodetic Warsaw zero’ (point (0,0))

1. Introduction

Warsaw's resources include numerous plans and studies conducted in the local Warsaw-25 and Warsaw-75 coordinate systems. To use them effectively in digital repositories, historical materials need to be linked to current reference databases (PUWG 2000/1992) and mechanisms that allow for quick and unambiguous identification of the correct sheet. This requires procedures that assign a sheet code (emblem) based on the coordinate point (X, Y) and, conversely, reproduce the section frame [X_{min}, Y_{min}, X_{max}, Y_{max}] from the given code.

In procedural terms, we use the concept of a 2×2 hierarchical division and the established boundary rules presented in our earlier works for IMW/PL-1992 [Zygmunt et al. 2021, 2023]. However, in the Warsaw approach, the frame of reference, the grammar of the sheet code, and the neighbourhood relations change. We propose closed calculation formulas with O(1) time on a regular metric grid, along with unambiguous edge qualification rules and a validation procedure adjusted to the specifics of W-25/W-75 (including the location of the Warsaw 'zero' and the N/S–E/W direction convention). In other words, we extend the 2×2 frame to the conditions of the local Warsaw system, while maintaining its simplicity and determinism.

The diversity of formats and systems in the geodetic and cartographic resource complicates integration and automation of processing. Therefore, we focus on deterministic, lightweight, and dictionary-free algorithms that are resistant to precision fluctuations. These include an encoder (X, Y → sheet code in the north/south – east/west convention with row and column numbers) and a decoder (sheet code → XYmin‑max frame range) with clearly defined interval closure.

The applications are straightforward: when working in PUWG 2000, one can convert coordinates to the Warsaw system, assign a sheet and obtain its envelope before loading the raster. Conversely, one can calculate the frames from the code in the file name and load only the sections that overlap the working area. These mechanisms work in conjunction with W-25/W-75 ↔ PUWG 2000 conversions, allowing archival content to be seamlessly combined with reference data. In AI/ML tasks, the section code acts as a stable location identifier, facilitating sample selection, organisation of collections and verification of matches.

Main results

- Procedures for W-25/W-75: fast algorithms (X,Y → sheet code) and (sheet code → range) with clearly described edge closure; the approach is an extension of the earlier 2×2 framework.
- Code syntax: clarification of the N/S–E/W convention (row, column) and numbering rules.
- Quality and stability control: compliance tests in both mapping directions, recommendations regarding tolerances and rounding.
- Reference implementations: VB6 modules ready for integration into GIS and ETL flows.

- Interoperability: points of integration with PUWG 2000/1992, facilitating the linking of archives with current data.

Background and literature

Historical sheet identification schemas derive from logical grids and hierarchies – from Penck's International Map of the World (1:1,000,000) to contemporary national systems [Parry and Perkins 2002, Pearson et al. 2006]. Similar solutions are used in national grid systems (USGS/USNG, British National Grid, NTS) [Longley et al. 2005, Ordnance Survey 2016, Natural Resources Canada 2007]. Tiled web maps have become established in web services, and quadtree/R-tree databases have become standard in databases, speeding up spatial queries [Samet 2006, Worboys and Duckham 2004]. Polish literature describes formalizations of determining the sheet code and scope for national/international systems [Zygmunt et al. 2021, 2023], which confirms the practical importance of the problem. This paper transfers these assumptions to the Warsaw system, integrating them with the local grammar of sheet code and the *XY*↔ sheet code and sheet code → frame procedures designed to work with PUWG 2000.

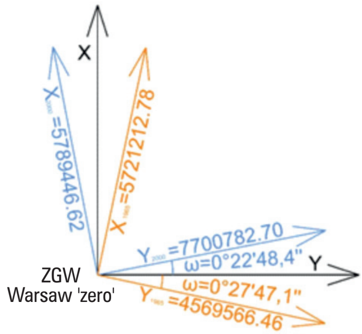
Due to the historical diversity of sheet divisions in Warsaw, we adopted a standardised sheet variant with dimensions: $dx = 2000$, $dy = 1500$, in accordance with the mathematical description of the axis, as the basis for the algorithm. We define the grid module Δx_{500}, Δy_{500} in metres as a derivative of 5×5 (400 m by 300 m).

2. Local coordinate system of the city of Warsaw (W-25/W-75)

The Warsaw local coordinate system was created and developed on the basis of the municipal triangulation and levelling network, connected to the national network and the practice of large-scale surveys. A local reference frame embedded in the so-called Warsaw zero (ZGW) was used, i.e. a point taken as the common origin of the coordinate system for urban cartographic works. In the following decades, two mature working versions of the system were in operation, commonly referred to as Warsaw-25 (W-25) and Warsaw-75 (W-75). They differed in terms of implementation details (including parametric and organisational aspects) and the scope of application in individual series of plans. In urban practice, the system was used to maintain and update large-scale maps, photoplans and industry documentation.

As part of this work, we do not reconstruct historical sheet formats in 1:1 scale, but rather standardise their handling on the algorithmic side. Given the numerous section divisions used in the past, we adopt a scale of 1:500 as the base variant (most common in urban practice) and define the Δx_{500}, Δy_{500} grid module in metres. This simplifies the integration of archives with reference data in PUWG 2000/1992, while using the Warsaw grammar of the N/S–O/W sheet code (row, column) and anchoring in ZGW.

The relationship between the local system and state systems, as well as the place of the proposed procedures in the broader context of the literature, are presented schematically in Figure 1 (axis orientations, conceptual shifts and rotations between systems).



Source: Author's own study

Fig. 1. Relations in the Warsaw local coordinate system (W-25/W-75) z PUWG 2000/1965

After the implementation of PUWG 2000 as a municipal standard (21st century), the Warsaw system was largely withdrawn from production, but it remains crucial in archival work and resource digitisation processes. This makes it all the more justified to provide unambiguous, constant time procedures for determining the sheet code and its frame, as well as consistent boundary rules, so that materials in W-25/W-75 can be automatically linked to reference layers and used in modern GIS/AI chains.

3. Sheet grid, indexing and section divisions (Warsaw)

In the case of the Warsaw local system (W-25/W-75), there have historically been various series and sectional divisions, which poses a challenge for the direct reproduction of uniform sheet specifications. For this reason, in this study, we standardise the notation and calculations, adopting a scale of 1:500 as the base variant (most commonly used in urban practice). The starting point of the system is 'Warsaw zero' (ZGW) – the cross on the dome of the Evangelical-Augsburg Church of the Holy Trinity at Małachowski Square, adopted at the end of the 19th century as the point (0,0) of the local system and the centre of first-order triangulation for Lindley's plans (Fundacja Warszawa 1939).

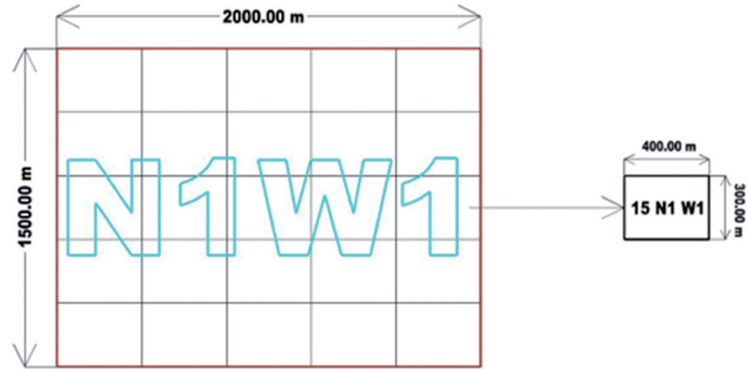
3.1. Geometrical model and sheet grid

We define the 1:500 standardisation variant using the sheet module:

$$\Delta x_{2000} = 2000 \text{ m}, \Delta y_{2000} = 1500 \text{ m},$$

which corresponds to a 5×5 grid of sub-sheets with dimensions:

$$\Delta x_{2000}/5 = 400 \text{ m}, \Delta y_{2000}/5 = 300 \text{ m}.$$



Source: Author's own study

Fig. 2. Division of the sheet module into 25 sections on a scale of 1:500

This solution is an engineering standardisation that facilitates the integration of archives with reference data and does not involve the 1:1 reconstruction of every historical sheet format. The practical need for such standardisation is demonstrated, among other things, by the discrepancies in the divisions of the Warsaw series (e.g. BOS 1:2500 – 50 sheets), as well as by the city's subsequent transition to the PUWG 2000 standard (Inventory Plan of Warsaw Destruction by the Capital Reconstruction Office from 1945–1946).

3.2. Sheet code grammar and spatial indexing

A deterministic grammar is adopted:

$$\langle \text{sheet code} \rangle = \langle \text{sub} \rangle \ \langle N \mid S \rangle \langle \text{row} \rangle \ \langle O \mid W \rangle \langle \text{column} \rangle$$

where:

$\langle \text{sub} \rangle \in \{1,\ldots,25\}$ – subindex of a sub-sheet in a grid of 5×5 (counted from top to bottom in lines of 5; 1..5 – upper row, 21..25 – bottom row),

$\langle \text{row} \rangle \geq 1$ – module line number (N/S) counted from ZGW,

$\langle \text{column} \rangle \geq 1$ – module column number (O/W) counted from ZGW.

Convention of directions:

N/S for axis Y (N: $Y \geq 0$, S: $Y < 0$); O/W for axis X (O: $X \geq 0$, W: $X < 0$).

Boundary rule:

left/bottom edge closed, right/top edge open – eliminates double counting of points located exactly on the boundaries.

Procedures in time O(1)

- $XY \rightarrow$ sheet code
  From point (X,Y), we determine ('row', 'column') by integer division:

$$\text{row} = \left\lfloor \frac{|Y|}{\Delta y_S} \right\rfloor + 1,\ \text{column} = \left\lfloor \frac{|X|}{\Delta x_S} \right\rfloor + 1,$$

and then subindex 1..25 by location in the 5×5 grid inside the module (from top/left). (*with tolerance ε added to X, Y only for deterministic resolution of points 'on the edge'*).

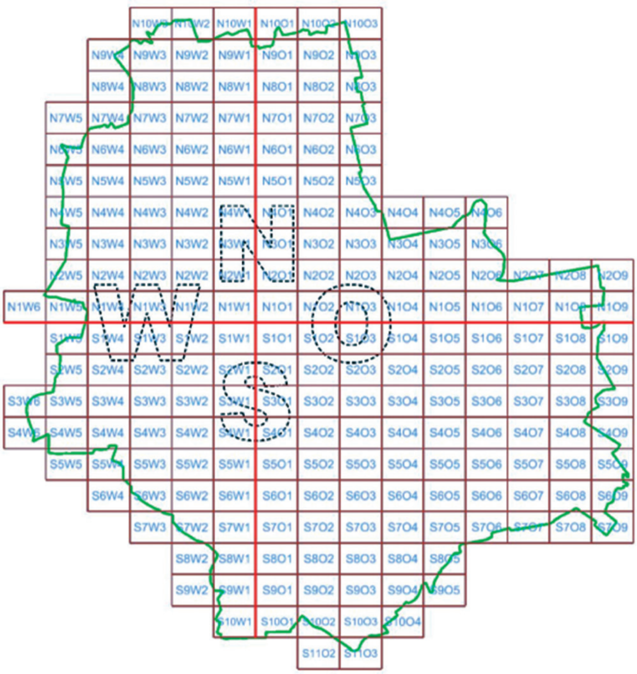
- sheet code → frame
  From (sub, *N*/*S*,row, *O*/*W*,column) bbox [X_{min}, Y_{min}, X_{max}, Y_{max}] is calculated as interval left-bottom–closed and right–top–oper, using the geometry of the Δx_S, Δy_S module and section location in the sheet (5×5).

3.3. Applications and visualisations

The defined grid and grammar of the sheet code allow for:

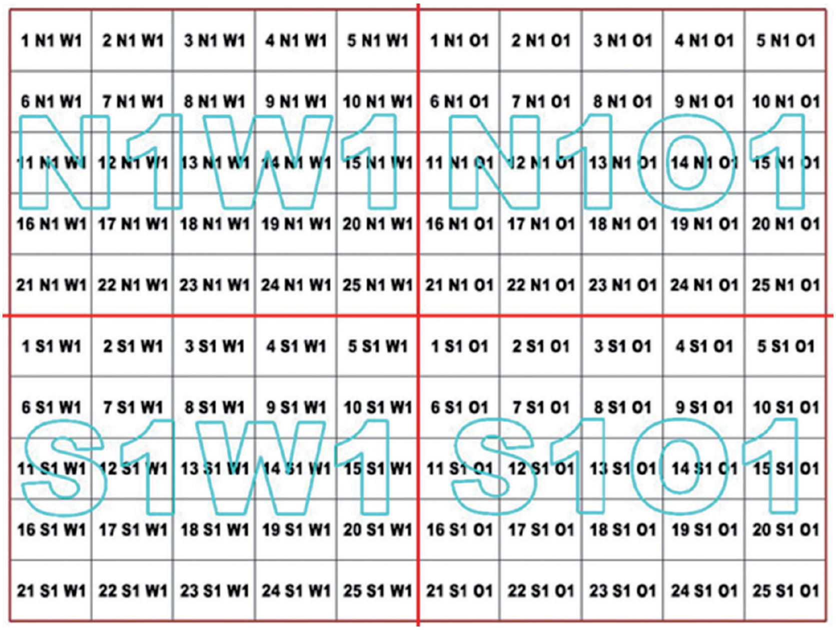
1. Planning the loading (prefetching) of sheets based on the file name.
2. Metadata quality control (compliance of the sheet code with geometry).
3. Labelling of raster fragments in AI/ML tasks.
4. Combining archives with reference layers in PUWG 2000.

   Figures 3 and 4 illustrate sample grid views.



Source: Author's own study based on the implementation presented in the paper

Fig. 3. Grid of sectional modules 2.0×1.5 km (standardised variant of the Warsaw W-25/W-75 system) against the backdrop of the city outline



Source: Author's own study

Fig. 4. Division of sample section modules measuring 2.0×1.5 km into 25 sub-sheets measuring 400×300 m (5×5 grid) for a scale of 1:500, with sub-index numbering 1..25

4. Practical significance and integration with resources

Under the conditions of parallel operation of the W-25/W-75 and PUWG 2000/1992 systems at PODGiK, stable and reproducible indexing procedures (encoder: *X*,*Y* → sheet code, and decoder: sheet code → bbox) become part of the processing infrastructure. They enable consistent 'bridging' of Warsaw archives with current reference databases, reducing ETL costs and the risk of errors at the contractor-user interface. The standardisation variant of the 2.0×1.5 km module (5×5 grid of 400×300 m sub-sheets) adopted in the work provides a single, simple logic for the entire resource: other scales are handled exclusively by rescaling the grid step, without changing the algorithm.

Key application scenarios:

1. Integration of archives with PUWG 2000. Transformation of the work point from PL-2000 to W-25/W-75, designation of the sheet code and frame, and then automatic pulling of the sheet scan to the current session (with closing left/bottom and opening right/top).
2. Metadata quality control. Comparison of the sheet code calculated from the geometry with the sheet code in the file name/metadata; inversibility tests (*XY* → sheet code → bbox contains the point; sheet code → bbox → centroid → sheet code returns the original label).

3. Mass ETL and file naming. Unambiguous naming rules (<sub> <N|S><row> <O|W><column>) facilitate directory organisation, selective loading (prefetch) and parallel batch processing.
4. Web services and applications. Query parameterisation based on sheet code (WMS/WMTS/WFS filters) and fast fetching of neighbouring sheets (5×5 grid heuristics) for smooth interactive operation.
5. AI/ML and semi-automatic digitisation. The sheet code, as a stable spatial identifier, is used to label samples, build training sets (raster segmentation, object classification) and validate model results.
6. Organisational resilience. Deterministic, constant time O(1) procedures minimise dependence on series-specific dictionaries and 'compliance maps,' facilitating maintenance and auditing.

Implementation (practical recommendations):

- Boundary rules: left/bottom edge closed, right/top edge open – unambiguous point classification.
- Numerical tolerance: single ε for edge cases (description in the appendix/tests).
- Parameterisation: base module 2.0×1.5 km for variant 1:500; other scales as 4 · S × 3 · S (m).
- Interfaces: make *XY* → sheet code and sheet code → bbox functions available in PODGiK tools (API/scripts), plus file naming rules and a minimum set of regression tests.

Indexing arranged in this way acts as a 'connection key' between the worlds of archives and current data, ensuring repeatable results regardless of whether processing takes place in a desktop environment (CAD/GIS) or in server processes (ETL, web services, AI pipelines).

5. Algorithm for determining the sheet code and the scope (Warsaw system)

5.1. Premises and notation

Module (1:500): $\Delta E = 2000$ m, $\Delta N = 1500$ m; 5×5 cells: sub$W = 400$ m, sub$H = 300$ m. Reference point: ZGW = (0, 0). Directions: *N*/*S* for the *N* axis (*Y*) – (*N*: $Y \geq 0$, *S*: $Y < 0$); *E*/*W* for the *E* axis (*X*) – (*E*: $X \geq 0$, *W*: $X < 0$). Boundary rule: left/bottom closed, right/top open. Epsilon: in the encoder, we add $\varepsilon = 1e{-}8$ to (E, N), which only influences edge cases. Axis notation: in descriptions, we use $\Delta E/\Delta N$; in the code, they correspond to the variables $dx \equiv \Delta E$, $dy \equiv \Delta N$ (CAD style: $X = E$, $Y = N$).

- $\Delta x = 2000\ \Delta y = 1500 \text{ m } w = \dfrac{\Delta x}{5} = 400 \text{ m } h = \dfrac{\Delta y}{5} = 300 \text{ m.}$

Standardisation variant 1:500. The sheet module has dimensions and

Inside the module, we use a fixed 5×5 grid, i.e. cells.

and

- (0,0)Reference point: ZGW.
- Directions convention: *N/S* for axis *Y*(N: $Y \geq 0$, S: $Y < 0$); *O/W* for axis *X* (O: $X \geq 0$, W: $X < 0$).
- Boundary rule: left/bottom edge closed, right/top edge open (eliminates double assignment of points at the boundary).
- Numerical tolerance: $\in = 10^{-8}$ added only in the encoder to *X,Y* before calculations, solely for the purpose of deterministically resolving ‘borderline’ cases.

Sheet code grammar (deterministic):

⟨sheet code⟩::=⟨sub⟩ ⟨N|S⟩⟨row⟩ ⟨O|W⟩⟨column⟩,

given ⟨sub⟩ ∈ {1,…,25} is subindex of a cel in a 5×5 grid (counted **from top to bottom** in rows of 5: 1..5 – top row, 21.. 25 – bottom row), and ⟨row⟩, ⟨column⟩ ≥ 1 are the module numbers relative to ZGW in directions N/S and O/W.

5.2. Encoder (XY → sheet code)

Aim. For the point (*X*, *Y*) in the Warsaw system, determine sheet code in the form of `“<sub> <N|S><row> <O|W><column>”.`

Steps of the calculation (represented by SheetWarszawaCodeFromXY):

1. Boundary stabilisation:

$$X \leftarrow X + \varepsilon,\ Y \leftarrow Y + \varepsilon.$$

2. Module size: $\Delta x = 2000$, $\Delta y = 1500$ (m).
3. Module numbers (1..∞):

$$\text{col} = \left\lfloor \frac{|X|}{\Delta x} \right\rfloor + 1,\ \text{row} = \left\lfloor \frac{|Y|}{\Delta y} \right\rfloor + 1.$$

Direction: *O* when $X \geq 0$, otherwise *W*; *N* when $Y \geq 0$, otherwise *S*.

4. Position in the module (grid 5×5):

$$x_0 = \left\lfloor \frac{X}{\Delta x} \right\rfloor \Delta x,\ y_{\text{top}} = \left\lfloor \frac{Y}{\Delta y} \right\rfloor \Delta y + \Delta y,$$

$$\text{slup} = \left\lfloor \frac{X - x_0}{\Delta x/5} \right\rfloor,\ \text{pas} = \left\lfloor \frac{y_{\text{top}} - Y}{\Delta y/5} \right\rfloor,$$

$$\text{sub} = 5 \cdot \text{pas} + \text{slup} + 1 \in \{1,\ \ldots,\ 25\}.$$

5. Label premis:

result = `sub + " " + (N|S)+row + " " + (O|W)+column.`

Complexity: time O(1), memory O(1).

Example: $(X, Y) = (0,0) \Rightarrow$ „21 N1 O1".

5.3. Decoder (sheet code → range)

Aim. For sheet code "<sub> <N|S><row> <O|W><column>" determine the sheet frame $[X_{min}, Y_{min}, X_{max}, Y_{max}]$.

Steps of the calculation (represented by `GetSheetWarszawaBoundsFromCode`):

1. Parsing and validation:

   split the code into three tokens; read:

$$\text{sub} \in [1,25],\ NS \in \{N, S\},\ \text{row} \geq 1,\ OW \in \{O, W\},\ \text{column} \geq 1.$$

2. Module and cell size:

$$\Delta x = 2000,\ \Delta y = 1500,\ w = 400,\ h = 300\ (\text{m}).$$

3. Position of the module relative to ZGW:

$$x_{left} = \begin{cases} (\text{column} - 1)\Delta x, & O \\ \text{column}\ \Delta x, & W \end{cases} \quad y_{top} = \begin{cases} \text{row}\ \Delta y, & N \\ -(\text{row} - 1)\ \Delta y, & S \end{cases}$$

4. Sub-sheet position in the grid 5×5:

$$\text{rowTop}=(\text{sub}-1)/5\ (\text{integer division}),\ \text{colLeft}=(\text{sub}-1) \bmod 5.$$

5. Left-top corner of the sub-sheet:

$$x_0 = x_{left} + \text{colLeft} \cdot w,\ y_0 = y_{top} - \text{rowTop} \cdot \text{h}.$$

6. Frame (boundary rule: left/bottom closed; right/top opened):

$$X_{min} = x_0,\ X_{max} = x_0 + w,\ Y_{max} = y_0,\ Y_{min} = y_0 - h.$$

Complexity: time O(1), memory O(1).

Example: „21 N1 O1" $\Rightarrow$ [0,0,400,300).

5.4. Edge stability and reversibility

- Stability. Adding ε to the encoder results in deterministic resolution of points on the grid (without changing the logic thresholds). The decoder operates without ε.
- Reversibility (round-trip test). For any sheet code, the centre of the designated frame encoded by the XY algorithm → sheet code gives precisely the same sheet code. For a point not lying on the right/top edge (open), inclusion occurs $(X, Y) \in [X_{min}, Y_{min}, X_{max}, Y_{max})$.

5.5. Implementation notes (VB6/VBA)

- Referential versions adopt the type **`Point3d`** (MicroStation). The encoder used `Int()` (floor also for negative numbers), while the decoder used a simple parser based on `Split`.
- Sheet code format is uniform: "`<sub> <N|S><row> <O|W><column>`" (multiple spaces/tabs permitted).
- No dictionary structures; algorithms are constant time and easily transferable to GIS/ETL/AI.

5.6. Referential listings (VB6/VBA)

```
Option Explicit

‘ WARSAW (W-25/W-75) — encoder/decoder
‘ Module: dx=2000 m, dy=1500 m; 5×5 sub-grid → subW=400 m, subH=300 m
‘ Edges: left/bottom CLOSED, right/top OPEN; Directions: N/S (Y), O/W (X)

Public Function SheetWarszawaCodeFromXY(Point As Point3d) As String
    Const Skala As Double = 500#, Epsilon As Double = 0.00000001#
    Dim point2 As Point3d, dx As Double, dy As Double
    Dim vert As String, horiz As String
    Dim Pas As Integer, Slup As Integer, Sklad As Integer
    Dim Slupk As Integer, Pask As Integer
    Dim Sklad1 As String, Sklad2 As String

    ‘ epsilon: deterministic on-edge classification
    Point.x = Point.x + Epsilon: Point.y = Point.y + Epsilon

    ‘ module size
    dx = Skala * 4#: dy = Skala * 3#

    ‘ module row/column (1..), relative to origin (ZGW)
    Slupk = Int(Abs(Point.x) / dx) + 1: Pask = Int(Abs(Point.y) / dy) + 1
    Sklad2 = CStr(Slupk): Sklad1 = CStr(Pask)

    ‘ directions EW/NS
    vert = IIf(Point.x < 0#, “W”, “O”): horiz = IIf(Point.y < 0#, “S”, “N”)

    ‘ module reference edges (left/top); Int() is floor for negatives
    point2.y = Int(Point.y / dy) * dy + dy
    point2.x = Int(Point.x / dx) * dx

    ‘ position in 5×5 sub-grid (0..4 from top/left)
    Pas = Int((point2.y - Point.y) / (dy / 5#))
    Slup = Int((Point.x - point2.x) / (dx / 5#))
```

```
    ‘ subindex 1..25 (row-major, 5 per row)
    Sklad = Pas * 5 + Slup + 1
    ‘ “<sub> <N|S><row> <O|W><col>”
    SheetWarszawaCodeFromXY = CStr(Sklad) & “ “ & horiz & Sklad1 & “ “ & vert
    & Sklad2
    End Function

Public Function GetSheetWarszawaBoundsFromCode( _
    ByVal SheetCode As String, ByRef Pmin As Point3d, ByRef Pmax As Point3d _
) As Boolean
    On Error GoTo Fail
    Const Skala As Double = 500#, dx As Double = 4# * Skala, dy As Double = 3# * Skala
    Const subW As Double = dx / 5#, subH As Double = dy / 5#
    Dim s As String, t() As String, NrSheet As Long, NrNS As Long, NrOW As Long
    Dim NorS As String, OorW As String, xLeft As Double, yTop As Double
    Dim rowTop As Long, colLeft As Long, x0 As Double, y0 As Double

    ‘ normalize whitespace and split into 3 tokens
    s = Trim$(Replace(SheetCode, vbTab, “ “)): Do While InStr(s, “ “) > 0: s =
    Replace(s, “ “, “ “): Loop
    t = Split(s, “ “): If UBound(t) <> 2 Then GoTo Fail

    ‘ parse “<sub> <N|S><row> <O|W><col>”
    NrSheet = CLng(t(0)): NorS = UCase$(Left$(t(1), 1)): NrNS = CLng(Mid$(t(1), 2))
    OorW = UCase$(Left$(t(2), 1)): NrOW = CLng(Mid$(t(2), 2))

    ‘ basic validation
    If NrSheet < 1 Or NrSheet > 25 Or NrNS < 1 Or NrOW < 1 Then GoTo Fail
    If (NorS <> “N” And NorS <> “S”) Or (OorW <> “O” And OorW <> “W”) Then
    GoTo Fail

    ‘ module left/top edges relative to origin
    xLeft = IIf(OorW = “O”, (NrOW - 1) * dx, -NrOW * dx)
    yTop = IIf(NorS = “N”, NrNS * dy, -(NrNS - 1) * dy)

    ‘ position in 5×5 sub-grid
    rowTop = (NrSheet - 1) \ 5: colLeft = (NrSheet - 1) - rowTop * 5

    ‘ upper-left corner and bbox (left/bottom closed; right/top open)
    x0 = xLeft + colLeft * subW: y0 = yTop - rowTop * subH
    Pmin.x = x0: Pmin.y = y0 - subH: Pmax.x = x0 + subW: Pmax.y = y0

    GetSheetWarszawaBoundsFromCode = True: Exit Function
Fail:
    GetSheetWarszawaBoundsFromCode = False
End Function
```

6. Verification and test examples

6.1. Criteria for correctness

- Consistency *XY* → sheet code → range.
  For any point $P = (E, N)$ in the Warsaw system the encoder (*XY* → sheet code) determines the label $\mathcal{G}$. Decoder (sheet code → range) has to return the frame $[E_{min}, N_{min}, E_{max}, N_{max})$ that contains the point *P*. The following rule applies to the edges: left/bottom closed, right/top open. The encoder assumes $\varepsilon = 10^{-8}$ added to *E*, *N* for the only reason to qualify deterministically the points 'on the line'.
- Consistency range → XY → sheet code.
  For any point *Q* selected from within the frame (e.g. its centroid), the decoder → encoder must return the identical sheet code $\mathcal{G}$.
- Sheet code format compatibility.
  The sheet code has the form ⟨sub⟩ ⟨*N*|*S*⟩⟨row⟩ ⟨*O*|*W*⟩⟨row⟩, where ⟨sub⟩∈{1..25}, ⟨row⟩≥1, ⟨column⟩≥1. Multiple spaces/tabs are permitted; the letters *N/S/O/W* are case-insensitive.

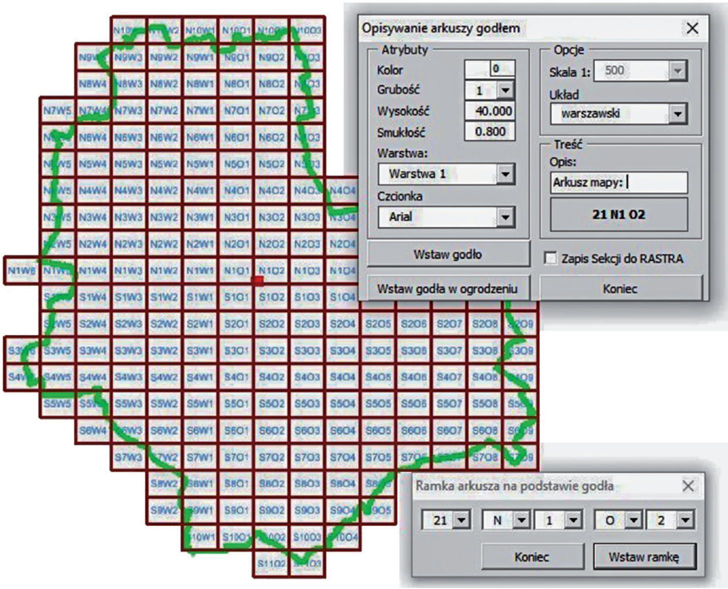
7. Practical application

The algorithm was implemented and tested in a CAD environment (MicroStation; VBA/MDL – Fig. 5). Rasters/references are attached as georeferenced; the file name = sheet code simplifies automatic searching and attaching sheets. Simple neighbourhood heuristics (on a 5×5 grid inside the module and by rows/columns of modules) allow adjacent sheets to be reloaded.

Workflow with PL-2000 (operationally):

the user indicates a point in PL-2000 → coordinates are (on the fly) transformed to W-25/W-75 → XY → the sheet code designates the label → sheet code → the range returns the frame → the application attaches the appropriate raster/reference and, after converting the frame from W-25/W-75 to the current GCS, sets the range and orientation. The entire process can be automated (macros, 'Point and Load' context menu, name-sheet code compliance verifier). Result: shorter operation time, fewer errors and greater consistency of the GIS/ETL process.

Note regarding axes: we use the CAD convention in the implementation: $E \equiv X$, $N \equiv Y$. In the mathematical description, we use ΔE, ΔN to avoid confusion with the PL-2000 geodetic notation ($X = N$, $Y = E$).



Source: Author's own study

Fig. 5. Practical use of the algorithm in the application, for point *XY* (top), or generating a section frame based on the sheet code (bottom)

8. Discusson

Module constants: $\Delta E = 2000$ m, $\Delta N = 1500$ m; constant sub-grid: sub$W = 400$ m, sub$H = 300$ m (grid 5×5).

Set of procedures

Two coupled algorithms were developed for the Warsaw system: 1) *XY* → sheet code (index directly from rectangular coordinates) and 2) sheet code → range (frame from hierarchical code). The normalisation module constants are key:

$$\Delta E = 2000 \text{ m}, \ \Delta N = 1500 \text{ m},$$

and constant sub-grid: $w = 400$ m, $w = 300$ (m) grid 5×5). The uniform use of these parameters guarantees the reproducibility and consistency of both calculation directions.

Hierarchy and unambiguity

The grammar of the sheet code ⟨sub⟩ ⟨*N*|*S*⟩⟨row⟩ ⟨*O*|*W*⟩⟨column⟩ combines the hierarchy of modules (row/column relative to ZGW) with a subindex 1..25 inside the module (5×5 grid). This notation is concise, deterministic and easy to validate.

Consistency and reversibility

Common fixed and identical boundary rules (left/bottom closed; right/top open) guarantee that *XY* → sheet code and sheet code → range are mutually consistent. Round-trip tests (point → sheet code → frame → centroid → sheet code) provide simple and strong proof of correct implementation.

Numerical aspects

Applying $\varepsilon = 10^{-8}$ in the encoder stabilises the qualification of points on grid lines; in other places is not necessary. In VB6, the Int function also acts as floor for negative numbers – with ports (C/C++, Python), the floor function must be explicitly used to maintain edge consistency.

Application of GIS/ETL

The Warsaw index functions as a spatial key:

- pre-filtering of archives (sheet code from name → frame → quick intersection tests with AOI),
- automatic attachment of rasters (name ≡ sheet code),
- stable joins between layers and raster directories,
- loading of adjacent sheets according to the current view/position on the grid.

Contribution to AI/ML

The sheet code acts as a spatial label: it supports the construction of training sets (raster segmentation, object classification), verification of results and organisation of historical data, also in multimodal pipelines (text-image-space).

Limits of applicability

1. The algorithms operate within the Warsaw system; points outside the system must be transformed to W-25/W-75.
2. The compatibility of the sheet code format (letters, number ranges) is required; anomalies in the archives (typos) should be detected by the validator.
3. The method does not replace geodetic transformations between CRS — it is deterministic indexing on the W-25/W-75 plane.

9. Conclusions

We presented concise, constant-time O(1) indexing procedures for the Warsaw system (W-25/W-75): *XY* → sheet code and sheet code → range, based on a normalisation module of 2.0×1.5 km and a 5×5 sub-grid. Common boundary rules and simple sheet code grammar provide reversibility and easy validation (round-trip tests), which translates into operational benefits in GIS/ETL (faster loading, stable connections, automation of work with archives). Reference implementations in VB6 are ready to use; porting to C++/Python only requires maintaining floor and ε in the coder for edge cases.

Directions for further work:

1. Multilingual library (VB/C++/Python) with unit tests and round-trip examples.
2. Anomaly detectors in ETL processes (detection of sheet code–name inconsistencies).
3. Modules for preloading resources – prefetching.
4. Extension with neighbourhood rules enabling quick navigation between sheets.

References

Fundacja Warszawa. 1939. https://www.warszawa1939.pl/obiekt/ewangelicki-malachowskiego [accessed:13 October 2025].

Goodchild M.F. 2007. Citizens as sensors: The world of volunteered geography. GeoJournal, 69(4), 211–221.

Longley P.A., Goodchild M.F., Maguire D.J., Rhind D.W. 2005. Geographic Information Systems and Science (2nd ed.). Wiley.

Natural Resources Canada, 2007. National Topographic System (NTS) of Canada.

Ordnance Survey. 2016. A Guide to the National Grid (Great Britain). Ordnance Survey.

Parry R.B., Perkins C.R. 2002. World Mapping Today (2nd ed.). K.G. Saur.

Pearson A.W., Collier P., Forrest D. 2006. Cartographic ideals and geopolitical realities: International maps of the world from the 1890s to the present. The Canadian Geographer, 50(2), 149–171.

Plan inwentaryzacji zniszczeń Warszawy Biura Odbudowy Stolicy z lat 1945‑1946. https://warszawa.ap.gov.pl/wp-content/uploads/2021/09/Zaproszenie.pdf [accessed: 13 October 2025].

Samet H. 2006. Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann.

Smith de M.J., Goodchild M.F., Longley P.A. 2018. Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools (6th ed.). The Winchelsea Press.

Worboys M.F., Duckham M. 2004. GIS: A Computing Perspective (2nd ed.). CRC Press.

Zygmunt M., Gniadek M., Szewczyk R. 2021. The method of determining the International Map of the World sheet number based on geographic coordinates on the example of Poland. Geomatics, Landmanagement and Landscape, 1(1), 69–77.

Zygmunt M., Michałowska K., Ślusarski M. 2023. The possibilities of the use of GUGIK's resources and functionalities in the spatial data processing and geocoding process. Geomatics, Landmanagement and Landscape, 1(3), 31–46.